ISSN 1213-6972 *Volume 23, Number 1, 2015*

Journal of WSCG

An international journal of algorithms, data structures and techniques for computer graphics and visualization, surface meshing and modeling, global illumination, computer vision, image processing and pattern recognition, computational geometry, visual human interaction and virtual reality, animation, multimedia systems and applications in parallel, distributed and mobile environment.

Vaclav Skala – Union Agency

Journal of WSCG

Editor-in-Chief: Vaclav Skala
c/o University of West Bohemia
Faculty of Applied Sciences
Univerzitni 8
CZ 306 14 Plzen
Czech Republic
http://www.VaclavSkala.eu

Managing Editor: Vaclav Skala

Printed and Published by:
Vaclav Skala - Union Agency
Na Mazinach 9
CZ 322 00 Plzen
Czech Republic

Hardcopy: ***ISSN 1213 – 6972***
CD ROM: ***ISSN 1213 – 6980***
On-line: ***ISSN 1213 – 6964***

Journal of WSCG

Editor-in-Chief

Vaclav Skala

c/o University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering
Univerzitni 8
CZ 306 14 Plzen
Czech Republic

http://www.VaclavSkala.eu
Journal of WSCG URLs: http://www.wscg.eu or http://wscg.zcu.cz/jwscg

Editorial Advisory Board
MEMBERS

Board of Reviewers

Agathos, Alexander (Greece)
Aires, Kelson (Brazil)
Aliaga-Badal, Carlos (Spain)
Apolinario Junior, Antonio Lopes (Brazil)
Assarsson, Ulf (Sweden)
Ayala, Dolors (Spain)
Bae, Juhee (United States)
Birra, Fernando (Portugal)
Bourke, Paul (Australia)
Brandao, Andre (Brazil)
Bucak, Serhat (United States)
Cakmak, Hueseyin Kemal (Germany)
Carozza, Ludovico (United Kingdom)
Cline, David (United States)
Didandeh, Arman (Canada)
Djado, Khalid (Canada)
dos Santos, Jefersson Alex (Brazil)
Drechsler, Klaus (Germany)
Durikovic, Roman (Slovakia)
Eisemann, Martin (Germany)
El Shafey, Laurent (Switzerland)
Emile, Bruno (France)
Fabio, Pellacini (Italy)
Facon, Jacques (Brazil)
Frejlichowski, Dariusz (Poland)
Fuenfzig, Christoph (Germany)
Galo, Mauricio (Brazil)
Gao, Zhi (Singapore)
Garcia Hernandez, Ruben Jesus (Germany)
Garcia-Alonso, Alejandro (Spain)
Gobron, Stephane (Switzerland)
Gois, Joao Paulo (Brazil)
Gomez-Nieto, Erick (Brazil)
Griffin , Amy (Australia)
Grottel, Sebastian (Germany)
Hast, Anders (Sweden)
Hernandez, Benjamin (United States)
Hinkenjann, Andre (Germany)
Hitomi, Yasunobu (Japan)
Hlawatsch, Marcel (Germany)
Horain, Patrick (France)
Hu, Xianlin (United States)
Hua, Binh-Son (Singapore)
Chajdas, Matthaeus (Germany)
Chen, Ding (Japan)
Chen, Weiya (France)
Iwasaki, Kei (Japan)
Jarabo, Adrian (Spain)
Jeschke, Stefan (Austria)
Jones, Mark (United Kingdom)
Jones, Ben (United States)
Jung, Soon Ki (Korea)
Kahler, Olaf (United Kingdom)
Kasprzak, Wlodzimierz (Poland)
Kerkeni, asma (Tunisia)
Klosowski, James (United States)
Kolcun, Alexej (Czech Republic)
Kraus, Martin (Denmark)
Kriglstein, Simone (Austria)
Kumar, Subodh (India)
Kurillo, Gregorij (United States)
Kurt, Murat (Turkey)
Lange, Benoit (France)
Last, Mubbasir (United States)
Lee, Jong Kwan (United States)
Lee, YT (Singapore)
Leite, Neucimar (Brazil)
Leon, Jean-Claude (France)
Lessig, Christian (Germany)
Li, Bo (United States)
Lin, Yuewei (United States)
Linsen, Lars (Germany)
Little, James (Canada)
Livesu, Marco (Italy)
Loscos, Celine (France)
Lu, Aidong (United States)
Maciel, Anderson (Brazil)
Mantiuk, Radoslaw (Poland)
Marques, Ricardo (France)
Masia, Belen (Spain)
Meiguins, Bianchi (Brazil)

Meng, Weiliang (China)
Menotti, David (Brazil)
Mestre, Daniel,R. (France)
Meyer, Alexandre (France)
Michael, Despina (Cyprus)
Michels, Dominik (United States)
Monti, Marina (Italy)
Montrucchio, Bartolomeo (Italy)
Movania, Muhammad Mobeen (Pakistan)
Mukai, Tomohiko (Japan)
Mura, Claudio (Switzerland)
Nagai, Yukie (Japan)
Nah, Jae-Ho (Korea)
Nanni, Loris (Italy)
Nogueira, Keiller (Brazil)
Nurzynska, Karolina (Poland)
Nyul, Laszlo (Hungary)
Oliveira, Joao Fradinho (Portugal)
Oztimur Karadag, Ozge (Turkey)
Paiva, Jose Gustavo (Brazil)
Parsons, Paul (Canada)
Patane, Giuseppe (Italy)
Paul, Padma Polash (Canada)
Peethambaran, Jiju (India)
Penedo, Manuel (Spain)
Pina, Jose Luis (Spain)
Pobegailo, Alexander (Belarus)
Puig, Anna (Spain)
Ramos, Sebastian (Germany)
Rasool, Shahzad (Singapore)
Reddy, Pradyumna (India)
Rehfeld, Stephan (Germany)
Rind, Alexander (Austria)
Rupprecht, Christian (Germany)
Sadlo, Filip (Germany)
Saito, Shunsuke (United States)
Santagati, Cettina (Italy)
Saraiji, MHD Yamen (Japan)
Saru, Dhir (India)
Seipel, Stefan (Sweden)
Shesh, Amit (United States)
Shi, Xin (China)
Shimshoni, Ilan (Israel)
Schaefer, Gerald (United Kingdom)
Schmidt, Johanna (Austria)
Schultz, Thomas (Germany)
Schwarz, Michael (Switzerland)
Silva, Romuere (Brazil)
Silva, Samuel (Portugal)
Singh, Rajiv (India)
Solis, Ana Luisa (Mexico)
Soriano, Aurea (Brazil)
Souza e Silva, Lucas (Brazil)
Spiclin, Ziga (Slovenia)
Svoboda, Tomas (Czech Republic)
Tavares, Joao Manuel (Portugal)
Teixeira, Raoni (Brazil)
Theussl, Thomas (Saudi Arabia)
Tomas Sanahuja, Josep Maria (Mexico)
Torrens, Francisco (Spain)
Tytkowski, Krzysztof (Poland)
Umlauf, Georg (Germany)
Vasseur, Pascal (France)
Vazquez, David (Spain)
Veras, Rodrigo (Brazil)
Walczak, Krzysztof (Poland)
Wanat, Robert (United Kingdom)
Wang, Lili (China)
Wang, Ruizhe (United States)
Wang, Lisheng (China)
Wenger, Rephael (United States)
Wijewickrema, Sudanthi (Australia)
Wu, YuTing (Taiwan)
Wu, Jieting (United States)
Wuensche, Burkhard,C. (New Zealand)
Xiong, Ying (United States)
Xu, Tianchen (Hong Kong SAR)
Xu, Chang (China)
Yang, Shuang (China)
Yasmin, Shamima (United States)
Yoshizawa, Shin (Japan)
Yu, Hongfeng (United States)
Zheng, Jianping (United States)
Zhong, Li (China)

Journal of WSCG

Vol.23, No.1, 2015

Contents

A New Extension to Kernel Entropy Component Analysis for Image-based Authentication Systems

Sepehr Damavandinejadmonfared
Advanced Cyber Security Research Centre Dept. of Computing, Macquarie University Sydney, Australia
sepehr.damavandinejadmonfared@mq.edu.au

Vijay Varadharajan
vijay.varadharajan@mq.edu.au

Abstract

We introduce Feature Dependent Kernel Entropy Component Analysis (FDKECA) as a new extension to Kernel Entropy Component Analysis (KECA) for data transformation and dimensionality reduction in Image-based recognition systems such as face and finger vein recognition. FDKECA reveals structure related to a new mapping space, where the most optimized feature vectors are obtained and used for feature extraction and dimensionality reduction. Indeed, the proposed method uses a new space, which is feature wisely dependent and related to the input data space, to obtain significant PCA axes. We show that FDKECA produces strikingly different transformed data sets compared to KECA and PCA. Furthermore a new spectral clustering algorithm utilizing FDKECA is developed which has positive results compared to the previously used ones. More precisely, FDKECA clustering algorithm has both more time efficiency and higher accuracy rate than previously used methods. Finally, we compared our method with three well-known data transformation methods, namely Principal Component Analysis (PCA), Kernel Principal Component Analysis (KPCA), and Kernel Entropy Component Analysis (KECA) confirming that it outperforms all these direct competitors and as a result, it is revealed that FDKECA can be considered a useful alternative for PCA-based recognition algorithms

1. Introduction

Fundamentally data transformation is of importance in machine learning and pattern analysis. The goal is to, alternatively, represent the high-dimensional data into a typically lower dimensional form revealing the underlying format and structure of the data. There is a large amount of literature on data transformation algorithms and methods [1], [2]. A dominant research area in data transformation is known as the so-called spectral methods. In spectral methods, the bottom or top eigenvalues (spectrum) and their corresponding eigenvectors play the main role in feature extraction and dimensionality reduction especially in constructed data matrixes. Some recent spectral methods include locally linear embedding [3], isometric mapping [4], and maximum variance unfolding [5], to name a few. See the recent review papers [6], [7] for thorough reviews of several spectral methods for dimensionality reduction. One of the most powerful and well known methods in the mentioned area is Principal Component Analysis (PCA) [8] which has been used in numerous applications and algorithms in data classification and machine learning[9],[10]. However, PCA [11] is a linear method which may not be beneficial when there might exist non-linear patterns hidden in the data. Over the last few decades, there have been a number of advanced improvements on PCA trying to overcome the drawback of linearly transformation and make PCA influential when dealing with nonlinear data. A very well-known and influential method is Kernel Principal Component Analysis (KPCA) [12]. In Kernel PCA [13], PCA is performed in a kernel feature space which is non-linearly related to the input data. It is enabled using a positive semi-definite (psd) kernel function computing the inner products within the new space (kernel feature space). Therefore, constructing the so-called kernel matrix or the inner product matrix is vital. Then, using the top eigenvalues and their corresponding eigenvectors to perform metric MDS [14] will lead to kernel PCA data transformation method. Kernel PCA has extensive use in many different contexts. For instance, kernel PCA has been used in machine learning algorithms from data classification [15] to data denoising [16][17][18]. In [19], kernel PCA is introduced for face recognition systems. Kernel PCA also has been used in finger vein recognition algorithms [20]. In 2010 [21], R. Jenssen proposed Kernel Entropy Component Analysis KECA as a new extension to kernel PCA. Kernel

ECA is fundamentally different from other spectral methods in two ways explained as follows; (1): The data transformation reveals structure related to the Renyi entropy of the input space data set and (2): The method does not necessarily use the top eigenvalues and eigenvectors of the kernel matrix. Shekar in 2012 [22], implemented KECA on face data base claiming KECA outperforms KPCA for face recognition purpose. In this paper, we develop a new spectral data transformation method, which is fundamentally different from Kernel ECA in the following important way:

- In FDKECA the dimension of the feature space is dependent on the dimension of the input data, not the number of input data. It means no matter how many data to analyze, the dimension of kernel matrix (kernel feature space) is fixed.

The mentioned difference will make the following advantages FDKECA has over KECA:

- FDKECA is much less computationally expensive than KECA as the dimension of the feature space, where the optimal PCA axes are calculated, is just as high as the dimension of the input data. This leads to a much faster method than traditionally used KECA.
- FDKECA has lower error rate than KECA as the axes obtained from our proposed feature space will contribute to more efficiency and less dimension compared to KECA.

The reminder of this paper is organized as follows: Section 2 illustrates some examples of spectral data transformation methods of importance. Feature Dependent Kernel Entropy Component Analysis (FDKECA) is developed in Section 3. The image reconstruction method and eigenface analysis using FDKECA are developed in Section 4. A spectral clustering algorithm using FDKECA is developed in section 5. Experimental results are presented in section 6. Finally, section 7 concludes the paper.

2. Spectral Data Transformation

In this section, we explain the fundamentals of PCA, KPCA, and KECA with examples to comprehend spectral basic data transformation methods.

2.1. Principal Component Analysis (PCA)

A well-known spectral data transformation method is PCA. Let $X = [x_1, ..., x_n]$, where $x_t \in R^d$ and $t = [1, ..., N]$. As PCA is a linear method, the following transformation is sought assuming A is $[d \times d]$ such that $y_t \in R^d$ and $t = [1, ..., N] : Y_{pca} = AX$ where $Y_{pca} = [y_1, ..., y_n]$. Therefore, the sample correlation matrix of Y_{pca} equals to:

$$\frac{1}{N} Y_{pca} Y_{pca}^T = \frac{1}{N} AX(AX)^T = A \frac{1}{N} XX^T A^T \qquad (1)$$

The sample correlation matrix of X is $\frac{1}{N} XX^T$. Determining A such that $\frac{1}{N} Y_{pca} Y_{pca}^T = I$ is the goal. Considering eigen-decomposition, we will have $\frac{1}{N} XX^T = V \triangle V^T$,where $\triangle$ is a diagonal matrix of the eigenvalues $\delta_1, ..., \delta_n$ in descending order having the corresponding eigenvectors $v_1, ..., v_n$ as the columns of V. Substituting into (1), it can be clearly observed that $A = \triangle^{-1/2} V^T$ leads to the goal such that $Y_{pca} = \triangle^{-1/2} V^T X$.

Performing a dimensionality reduction from d to $l \leq d$ is often achieved by the projection of data onto a subspace spanned by the eigenvectors (principal axes) corresponding to the largest top l eigenvalues.

2.2. Kernel Principal Component Analysis (KPCA)

Scholkoft in 1998 proposed Kernel PCA which is a nonlinear version of PCA operating in a new feature space called kernel feature space. This space is non-linearly related to the input space. The nonlinear mapping function (kernel function) is given $\Phi : R^d \rightarrow F$ such that $x_t = \Phi(x_t), t = 1, ..., N$ and $\Phi = [\Phi(x_1), ..., \Phi(x_N)]$. After performing such mapping in input data, PCA if implemented in F, we need an expression for the projection of P_{U_i} of Φ onto a subspace of feature space principal axes, for example, top l principals. It can be given by a positive semi-definite kernel function or Mercer kernel [23] [24], $k_\sigma = R^d \times R^d \rightarrow R$ computes an inner product in the Hilbert space F:

$$k_\sigma(x_t, x_t') = \langle \phi(x_t) \phi(x_t') \rangle \qquad (2)$$

The $(N \times N)$ kernel matrix K is defined such that element (t, t') of the kernel matrix equals to $k_\sigma(x_t, x_t')$. Therefore, $K = \Phi^T \Phi$ is the inner product matrix (Gram matrix) in F. Then, Eigen-decomposing the kernel matrix we have $K = EDE^T$ where E is the eigenvectors $e_1', ..., e_n'$ column wise and their corresponding eigenvalues are in D - $\lambda_1, ..., \lambda_n$- . Williams in [25] discussed that the equivalence between PCA and KPCA holds in KPCA as well (kernel feature space). Hence, we have:

$$\Phi_{pca} = P_{U_i} \Phi = D_l^{1/2} E_l^T \qquad (3)$$

Where D_l is the top large l eigenvalues of K andE_l is their corresponding eigenvectors stored in columns. It means that projecting Φ onto spanned feature space (principal axes) is given by $P_{U_i} \Phi = \sqrt{\lambda_i} e_i^T$.

Considering the analogy in (3), $\Phi_{pca} = D_l^{1/2} E_l^T$ is the solution to the following optimization problem:

$$\Phi_{pca} = D_l'^{1/2} E_l'^T : \min_{\lambda_1', e_1', ..., \lambda_N', e_N'} 1^T (K - K_{pca})^2 .1. \quad (4)$$

Where $K_{pca} = \Phi_{pca}^T \Phi_{pca}$. Therefore, this procedure minimizes the norm of $K - K_{pca}$.

2.3. Kernel Entropy Component Analysis

Selection of the subspace where the data is projected onto is of importance in spectral methods, which is achieved based on the top or bottom eigenvectors in PCA and KPCA. In KECA, however, this stage is based on entropy estimate. Using entropy estimate, the data transformation from higher dimension to lower dimension is obtained by projecting the input data onto the axes, which contribute to the entropy estimate of input space. The procedure of entropy estimate in KECA is given as follows: The Renyi entropy function is defined by

$$H(P) = -\lg \int p^2(x) d(x) \quad (5)$$

Where p is probability density of the input data. Considering the monotonic nature of logarithmic function, (12) can be replaced by the following equation:

$$V(P) = \int p^2(x) d(x) \quad (6)$$

Estimating $V(p)$, (14) is given:

$$\hat{p}(x) = 1/N \sum_{x_t \epsilon S} k_\sigma(x, x_t) \quad (7)$$

$k(x, x_t)$ is the kernel centred matrix, then:
$\hat{V}(p) = 1/N \sum_{x_t \epsilon S}^{p} (x_t)$

$$1/N \sum_{x_t \epsilon S} 1/N \sum_{x_t \epsilon S} k_\sigma(x, x_t) = 1/N^2 1^T K 1 \quad (8)$$

where K is $k_\sigma(x, x_t)$ and 1 is an $(N \times 1)$ vector which contains all ones. The Renyi entropy estimating can be calculated for eigenvalues and eigenvectors of the Kernel matrix. It is defined as $K = EDE^T$, where D includes the eigenvectors, $\lambda_1, \lambda_2, ..., \lambda_N$, and E consists of eigenvalues, $\alpha_1, \alpha_2, ..., \alpha_N$. Finally, rewriting (15), we have:

$$(p) = 1/N^2 \sum_{1}^{N} (\sqrt{\lambda_i} \alpha_i^T 1)^2 \quad (9)$$

3. Feature Dependent Kernel Entropy Component Analysis (FDKECA)

In this section, we will go through PCA and KECA feature space in details and clarify our motivation to propose the new transformation method, and then FDKECA is introduced.

3.1. Defining the Feature Dependent Kernel Entropy transformation

Generally, in spectral data transformation methods, finding the most valuable principal axes (appropriate directions in the feature space) is of greatest importance. In PCA, for example, it is extracted linearly from the principal feature space. In KECA, however, these axes are extracted from kernel Entropy feature space as discussed in previous subsection. We define Feature Dependent Kernel Entropy Component Analysis as a k-dimensional data transformation method obtained by projecting input data onto a subspace spanned by principal kernel axis contributing to the feature dependent kernel Entropy space. Feature dependent kernel Entropy space is defined as follows:

Let $X = [x_1, ..., x_N]$, where $x_t \in R^d$ and $t = [1, ..., N]$. The nonlinear mapping function is given $\Phi : R^d \rightarrow F^d$ such that $x_t' = \Phi(x_t'), t = 1, ..., d$ where x_t' is an N dimensional vector including all of the t_{th} features from N input data. Explaining this, we have $\Phi = [\phi(x_1'), ..., \phi(x_d')]$. The use of a positive semi-definite kernel function or Mercer kernel computes an inner product in the new space F^d:

$$k_\sigma(x_t', x_{t'}') = \langle \phi(x_t') \phi(x_{t'}') \rangle \quad (10)$$

The $(N \times N)$ kernel matrix-we define that as K_{FDKECA} -is now defined such that element (t, t') of the kernel matrix is $k_\sigma(x_t', x_{t'}')$. Therefore, K_{FDKECA} is the Gram matrix or the inner product matrix in F^d. The next stage in FDKECA is to perform PCA on K_{FDKECA}. Note that the kernel matrix taken in FDKECA feature space (K_{FDKECA}) is totally different from that of KPCA.

Fig. 1. illustrates a brief flow diagram of reaching kernel Entropy feature space from scratch. As it is shown in Fig. 1, N input data are first mapped into kernel space by ϕ and then the Gram matrix (kernel matrix) is calculated using inner product. Note that the dimension of kernel matrix is equal to the number of input data- N. Eigen-decomposition is the next step where all eigenvalues and their corresponding eigenvectors are extracted and reordered in a descending manner from the greatest to the smallest value. After finding the kernel axes in this space, the kernel matrix, which represents the input data, is projected onto the kernel feature vectors (eigenvectors). The drawback to KECA is that the dimension of feature space and kernel matrix could become too high and as a result data transformation could be computationally expensive. In addition, finding the most

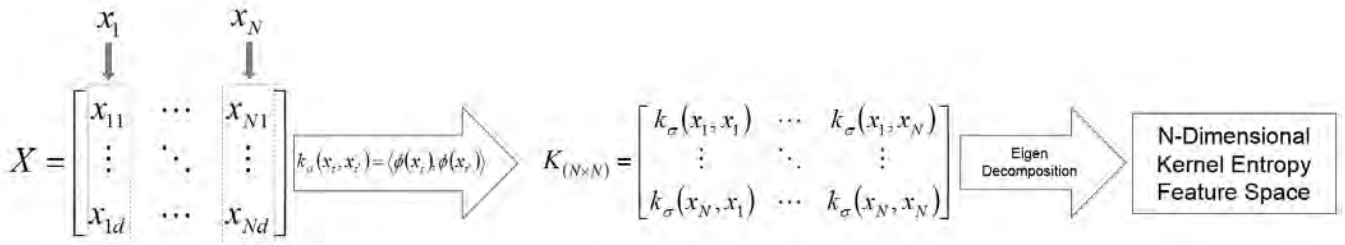


Figure 1. Flow diagram of reaching Kernel Entropy Feature Space

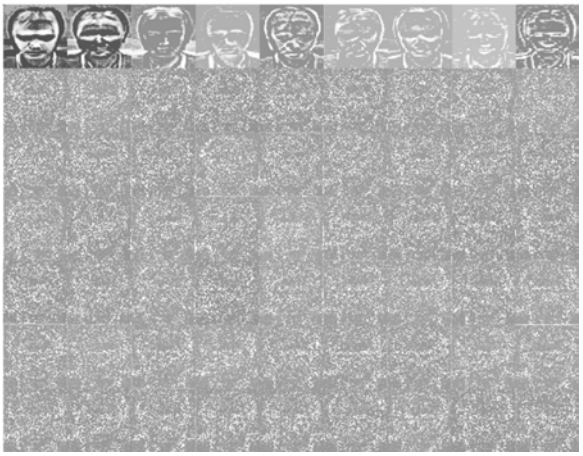


Figure 3. Top 63 eigenvectors obtained by PCA

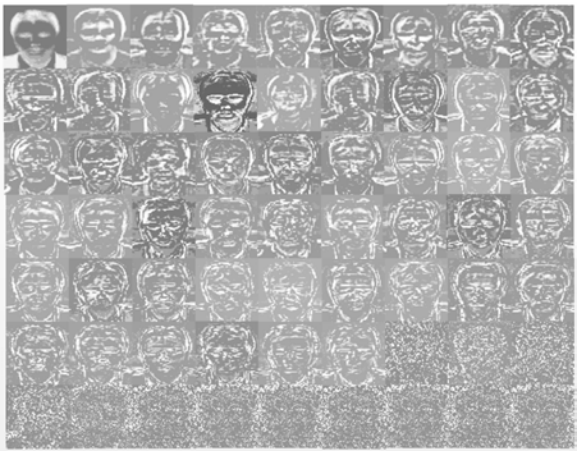


Figure 4. Top 63 eigenvectors obtained by FDKECA

optimized sub-space in kernel feature space could be challenging and sometimes inefficient.

Fig. 2. demonstrates FDKECA feature space where the input data is projected onto a subspace spanned by principal kernel entropy axes contributing to the feature dependent kernel entropy space. As it is illustrated in Fig.2, FDKECA considers all features having the same dimension from all input data in separate vectors first and then maps them into kernel space which is called FDKECA feature space. Then it computes the kernel matrix (Gram matrix) using inner products which is a d-dimensional space. Note that the input data has the dimension of d which means there is no growth of dimension while computing FDKECA feature space. Having d-dimensional FDKECA feature space, the eigenvectors and their corresponding eigenvalues are decomposed in this step using the estimation of entropy. The original input data is projected onto a sub-space of FDKECA feature vectors for the purpose of transformation and dimensionality reduction.

4. Eigenface Analysis on PCA and FDKECA

For more detailed comparison, we have performed PCA and FDKECA on the first individuals samples and visualized the first 63 feature vectors (eigenfaces) which are shown in Fig. 4 and 5.

In this analysis, we used 10 samples of the first subject of SCface database in PCA and FDKECA. In PCA, all samples were first converted into 1-D vectors. After calculating the mean vector (the mean image), the co-variance matrix is obtained and then, the Eigen-decomposition is performed on the co-variance matrix. The eigenvectors (PCA eigenfaces) were then reordered according to the greatness of their corresponding eigenvalues (in descending order). Fig. 4 shows the top 63 eigenvectors obtained by PCA. As it was expected, the top eigenvector carries the most information and the amount of information being carried by the feature vectors reduces as the eigenvector gets farther from the top one and closer to the bottom one. Another expectation is that only the first 9 or 10 top eigenvectors have some valuable information and the rest of the axes (eigenvectors) seem not to be useful as almost no related information can be seen in them. In terms of FDKECA, however, it is different.

In FDKECA, we used the polynomial kernel function with the degree of two. Firstly, all samples were converted into 1-D vectors. After calculating the mean vector (the mean image), all samples were mapped by the polynomial kernel function (as described in section III). Then, the Eigen-decomposition was performed on K_{FDKECA} to achieve the feature vectors and finally the axes were reordered based on entropy estimate. Fig. 5 illustrates the

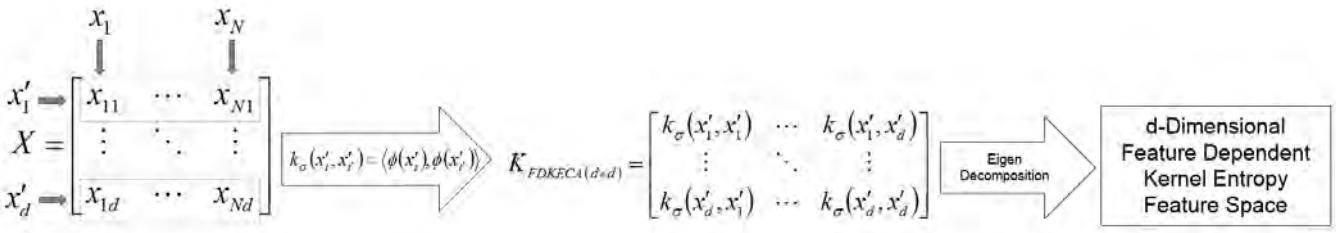


Figure 2. Flow diagram of reaching Feature Dependent Kernel Feature Space

top 63 eigenvectors obtained by FDKECA. Same as PCA, it was expected that the top eigenvector carries the most information and the amount of information drops as the number of the eigenvector gets closer to the bottom one. However, there is a considerable discrepancy between the shown eigenfaces obtained by PCA and FDKECA. In FDKECA, all eigenfaces carry relevant information except for the last 12 while in PCA only the first 9 or 10 ones have information related to the original face images. This analysis shows that FDKECA finds more informative and valuable feature vectors compared to PCA (as shown in Fig. 3 and4).

5. Spectral Clustring Algorithm Using FDKECA

In this section, a spectral clustering algorithm is developed using FDKECA transformation. The proposed algorithm, actually, is suitable for image classification which works in a supervised system as there are some samples to train the system and then using different samples, the system is tested. We first introduce the FDKECA clustering algorithm and then compare it with other algorithms such as PCA, KPCA and KECA in next section. As FDKECA can be considered as an extension to 1-D PCA, in our clustering algorithm all samples are converted into vectors. The goal is to propose a clustering system which not only is fast enough (not as computationally expensive as KECA), but also outperforms PCA, KPCA and KECA in terms of clustering image samples. Such an algorithm can be used in recognition systems like face, finger print, finger vein, palm vein etc.

Fig.5 indicates the flow diagram of the proposed clustering algorithm for image classification. We believe this algorithm can be applied in image-based recognition systems such as face and finger vein recognition. Moreover, this algorithm is much faster than normal KECA as its dimension of feature vector is fixed and it does not become too computationally expensive when analyzing a huge number of data. In addition to having a high speed, this algorithm is believed to be more appropriate than PCA, KPCA and KECA as it was shown in previous section. We have conducted different experiments on two different databases to have a complete analysis on the proposed algorithm. Next section gives experimental results on face and finger vein database.

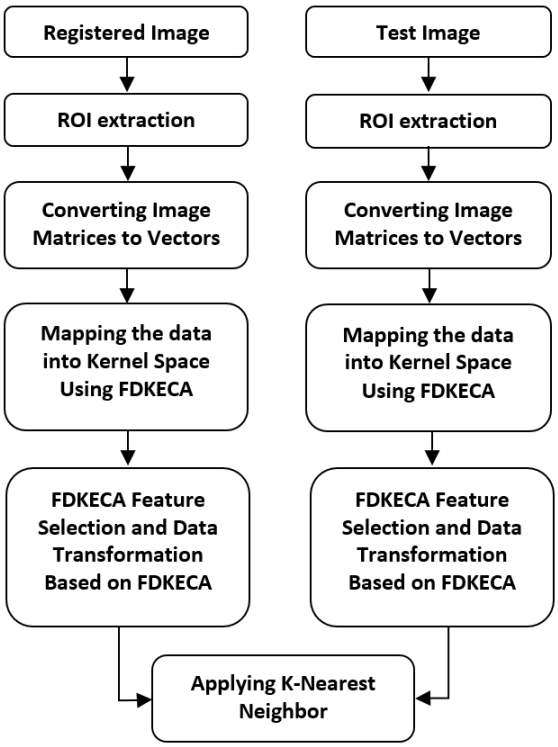


Figure 5. Flow diagram of the proposed clustring algorithm using FDKECA

6. Experimental Results

In this section, the performance of FDKECA is evaluated and compared with PCA, KPCA, and Kernel Entropy Component Analysis (KECA) on two different databases- finger vein and face. The experiments are conducted on Surveillance Camera Face Database (SCface database) and Finger vein database which are explained in two experimental setups in the following part of this section.

6.1. Experimental setup-1

The first part of the experiments is on finger vein database. The finger vein samples are collected using our own designed scanner. We will not go through the detailed discussion on how the data is collected and prepared as it might not be totally relevant to this work, See [27] for more information on the database.

10 samples were used from each of 200 individuals which results in a finger vein database consisting of 2000 samples. Region of Interest is detected and extracted from each sample automatically. Fig. 6 shows an original and cropped sample from the database. Two independent experiments have been conducted on this database. Firstly, the performance of FDKECA is compared with PCA, KPCA, and KECA where 5 randomly selected samples were used to train the algorithm and the remaining 5 to test. Then we used leave-one-out strategy to have a better comparison. Gaussian kernel is used in FDKECA, KPCA, and KECA algorithms in this stage. As in PCA-based image analysis the size of the samples is of importance, all finger vein samples have been normalized to the size of (10×20) to have a balance between speed and efficiency. In one-dimensional PCA-based algorithms, the first step is to convert the data from matrices into vectors which leads into vectors with the dimension of (1×200). It means there could be 200 different implementations of FDKECA on the data using 200 different feature vectors to project the data onto. However, it is totally different in KPCA and KECA as it is dependent on the number of input data being transferred into kernel space. For the sake of comparison, the first 200 kernel feature vectors were used in our implementations. In each single experiment, the implementation is repeated 200 times and the maximum accuracies and their corresponding dimension of feature vector are gathered and shown in Table 1. As it is observed from this table, KPCA and KECA achieve their maximum accuracy in a much higher dimension of feature vector in comparison with PCA. It is because feature space in KPCA and KECA is very high dimensional. more precisely, if 9 image from each category is used to train, it leads to a total number of 1800 train samples as there are 200 individuals. Having 1800 input samples in KPCA and/or KECA will result in a feature space with the dimension of (1800×1800) , while in PCA the dimension is fixed and equal to 200 in this experiment. The FDKECA, however, results in having the highest accuracy rate while its dimension of feature vector is almost as high as PCA, which means this method is not computationally as expensive as KPCA and KECA. Moreover, there is a dramatic gap between FDKECA and KECA which is more than 10 percent in the first experiment.

Table 1. Comparison of FDKECA with Other Methods Using the finger vein Database

Strategy	Method	Max Acc %	Dimension
5 for training	$KPCA$	85.9	200
	$KECA$	86	175
	PCA	95.3	68
	$FDKECA$	97.2	46
Leave-one-out	$KPCA$	92.5	173
	$KECA$	93.5	86
	PCA	98.5	35
	$FDKECA$	99.4	85

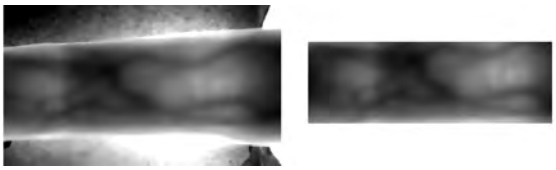


Figure 6. Original and ROI extracted finger vein sample

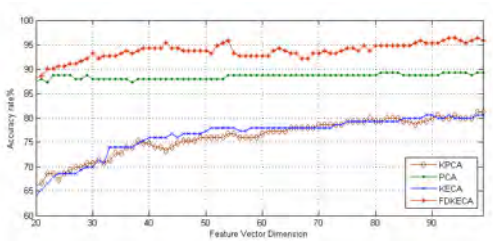


Figure 7. SCface classification using images of 4 cameras for training and 1 to test

6.2. Experimental setup-2

In the second experimental setup, we chose SCface database which is already explained in section 4. There are five different cameras located in three different distances from the individuals to collect the face data. In this part, we conducted the experiment using the images of 4 randomly selected cameras for training and the remaining 1 camera for testing. For each algorithm, the experience was repeated 100 times using the first 100 different eigenvectors to project the data onto and the results were gathered and visualized in Fig. 7. It is observed that Like the previous setup, FDKECA outperforms PCA, KPCA, and KECA in all experiments. As Fig.7 indicates, FDKECA reaches the highest accuracy of almost %98 while PCA, KPCA, and KECA get the accuracy of %89, %79 and %81 respectively.

7. Conclusion

We introduced a new data transformation method in this research work for dimensionality reduction in image-based recognition systems. Feature Dependent Kernel Entropy

Component Analysis (FDKECA) is an extension to both 1D-PCA and 1D-KECA. In FDKECA, all data is mapped into kernel space feature-wisely which results in having a constant dimension of data as well as being able to extract more valuable feature vectors in FDKECA feature space. Eigenface analysis showed that the feature vectors in FDKECA feature space are more informative than PCA. To examine FDKECA in practical clustering and classification methods and to be able to have a complete comparison with PCA, KPCA, and KECA, we proposed a clustering algorithm using FDKECA which was examined in two different areas- face recognition and finger vein recognition. Experimental results showed that FDKECA outperforms PCA, KPCA, and KECA which shows the reliability of FDKECA to be applied in image classification and recognition systems.

References

[1] R.O. Duda, P. E. Hart, and D.G. Stork *Pattern Classification.*, John Wiley and Sons, 2001.

[2] S. Theodoridis and K. Koutroumbas, *Pattern Recognition.* Academic Press, 1999.

[3] S. Roweis and L. Saul, *Nonlinear Dimensionality Reduction by Locally Linear Embedding,* Science, vol. 290, pp. 2323-2326, 2000.

[4] J. Tenenbaum, V. de Silva, and J.C. Langford, *A Global Geometric Framework for Nonlinear Dimensionality Reduction,* Science, vol. 290, pp. 2319-2323, 2000.

[5] K.Q. Weinberger and L.K. Saul, *Unsupervised Learning of Image Manifolds by Semidefinite Programming,* Intl J. Computer Vision, vol. 70, no. 1, pp. 77-90, 2006.

[6] L.K. Saul, K.Q. Weinberger, J.H. Ham, F. Sha, and D.D. Lee, *Spectral Methods for Dimensionality Reduction,* Semisupervised Learning, O. Chapelle, B. Scholkopf, and A. Zien, eds., chapter 1, MIT Press, 2005.

[7] C.J.C. Burges, *Geometric Methods for Feature Extraction and Dimensional Reduction,* Data Mining and Knowledge Discovery Handbook: A Complete Guide for Researchers and Practitioners, O. Maimon and L. Rokach, eds., chapter 4, Kluwer Academic Publishers, 2005.

[8] Jolliffe, I. 2005. *Principal Component Analysis. Encyclopedia of Statistics in Behavioral Science.*

[9] L. Sirovich and M. Kirby, *Low-Dimensional Procedure for Characterization of Human Faces,* J. Optical Soc. Am., vol. 4, pp. 519-524, 1987.

[10] M. Kirby and L. Sirovich, *Application of the KL Procedure for the Characterization of Human Faces,* IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 1, pp. 103-108, Jan. 1990.

[11] I.T. Jolliffe, *Principal Component Analysis. Springer Verlag*, 1986.

[12] B. Scholkopf, A.J. Smola, and K.-R. Mu ller, *Nonlinear Component Analysis as a Kernel Eigenvalue Problem,* Neural Computation, vol. 10, pp. 1299-1319, 1998.

[13] Schlkopf, Bernhard, Alexander Smola, and Klaus-Robert Mller. *"Kernel principal component analysis."* Artificial Neural Networks-sICANN'97. Springer Berlin Heidelberg, 1997. 583-588.

[14] H. Hotelling, *Analysis of a Complex of Statistical Variables into Principal Components,* J. Educational Psychology, vol. 24, pp. 417- 441, 1933.

[15] M.L. Braun, J.M. Buhmann, and K.-R. Mu ller, *On Relevant Dimensions in Kernel Feature Spaces,* J. Machine Learning Research, vol. 9, pp. 1875-1908, 2008.

[16] J.T. Kwok and I.W. Tsang, *The Pre Image Problem in Kernel Methods,* IEEE Trans. Neural Networks, vol. 15, no. 6, pp. 1517- 1525, 2004.

[17] S. Mika, B. Scholkopf, A. Smola, K.R. Mu ller, M. Scholz, and G. Ratsch, *Kernel PCA and Denoising in Feature Space,* Advances in Neural Information Processing Systems, 11, pp. 536-542, MIT Press, 1999.

[18] B. Scholkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Mu ller, G. Ratsch, and A.J. Smola, *Input Space versus Feature Space in Kernel-Based Methods,* IEEE Trans. Neural Networks, vol. 10, no. 5, pp. 1299-1319, 1999.

[19] Kim, Kwang In, Keechul Jung, and Hang Joon Kim. *"Face recognition using kernel principal component analysis."* Signal Processing Letters, IEEE 9.2 (2002): 40-42.

[20] Damavandinejadmonfared, Sepehr, et al. *"Finger Vein Recognition using PCA-based Methods."* World Academy of Science, Engineering and Technology 6.6 (2012): 1079-1081.

[21] Jenssen, Robert. *"Kernel entropy component analysis."* Pattern Analysis and Machine Intelligence, IEEE Transactions on 32.5 (2010): 847-860.

[22] Shekar, B. H., et al. *"Face recognition using kernel entropy component analysis."* Neurocomputing 74.6 (2011): 1053-1057.

[23] J. Mercer, *Functions of Positive and Negative Type and Their Connection with the Theory of Integral Equations,* Philosophical Trans. Royal Soc. London, vol. A, pp. 415-446, 1909.

[24] K.R. Mu ller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, *An Introduction to Kernel-Based Learning Algorithms,* IEEE Trans. Neural Networks, vol. 12, no. 2, pp. 181-201, Mar. 2001.

[25] C.K.I. Williams, *On a Connection between Kernel PCA and Metric Multidimensional Scaling,* Machine Learning, vol. 46, pp. 11-19, 2002.

[26] Grgic, Mislav, Kresimir Delac, and Sonja Grgic. *"SCfacesurveillance cameras face database."* Multimedia tools and applications 51.3 (2011): 863-879.

[27] S.Damavandinejadmonfared, and V.Varadharajan. *Finger Vein Recognition in Row and Column Directions Using Two Dimensional Kernel Principal Component Analysis* Proceedings of the 2014 International Conference on Image Processing, Computer Vision Pattern Recognition, IPCV14, July 2014, USA.

Cloth Simulation Using Soft Constraints

Mihai Frâncu
Politehnica University
Bucharest, Romania
mihai.francu@cs.pub.ro

Florica Moldoveanu
Politehnica University
Bucharest, Romania
florica.moldoveanu@cs.pub.ro

Abstract

This paper describes a new way of using projective methods for simulating the constrained dynamics of deformable surfaces. We show that the often used implicit integration method for discretized elastic systems is equivalent to the projection of regularized constraints. We use this knowledge to derive a Nonlinear Conjugate Gradient implicit solver and a new projection scheme based on energy preserving integration. We also show a novel way of adding damping to position based dynamics and a different view on iterative solvers. In the end we apply these fresh insights to cloth simulation and develop a constraint based finite element method capable of accurately modeling thin elastic materials.

Keywords

Implicit integration, constraints, projection, PBD, damping, iterative solver, cloth, FEM

1 INTRODUCTION

For decades now the preferred method of simulating elastic systems in computer graphics has been the implicit integration of the equations of motion. The method is very attractive due to its unconditional stability and large time steps. It has been widely used for simulating cloth and finite difference or finite element soft bodies in general. Constraint based methods on the other hand have not received that much attention, with the exception of rigid body dynamics. Only recently there has been an increase in the number of papers on the subject in relation to soft bodies and we believe there is room for improvement. Many regard the method as being an inaccurate approximation of natural phenomena which are better described by elasticity theory. In this paper we aim to reconcile the two methods and show that they are two faces of the same problem; this can prove useful for the further development of both approaches.

1.1 Related work

Constraint based methods have appeared originally in their acceleration based formulation for rigid body dynamics [Bar94]. Later on, velocity or impulse based methods gained more popularity [AH04, Erl07]. Position based methods are actually a nonlinear version of velocity based ones, in the sense that they can still be expressed as velocity filters, but constraints are enforced at positional level [ST96]. Such a method was made popular in games by [Jak01] and was later refined and extended by [MHHR07] under the name Position Based Dynamics (PBD). Part of the inspiration for this method came from molecular dynamics where methods like SHAKE or RATTLE are widely used [BKLS95]. A more detailed study for the application to cloth simulation in computer graphics was done in [Gol10]. Here the method of *fast projection* is developed based on an implicit treatment of constraint directions [HCJ*05] and a better energy preserving integrator is also derived. A similar method was used to develop the unified Nucleus solver in Autodesk Maya [Sta09]. Position based methods rely on *projection* for solving differential algebraic equations (DAE), which is ultimately an optimization problem [HLW06]. Another part of inspiration came from *strain limiting* techniques used in elastic cloth simulation [Pro96, BFA02].

Constraint based methods are often criticized for the fact they simulate only nearly inextensible materials and are prone to *locking*. In order to address this [EB08] use fast projection in conjunction with a BDF-2 integrator on a conforming triangular mesh. They also give a brief proof for fast projection being the limit of infinitely stiff elastic forces. Other authors prefer to use quad-predominant meshes or diamond subdivision [Gol10].

Constraint *regularization* was employed mainly in [Lac07] for making rigid dynamics with contact and friction more tractable numerically. We take the name *soft constraints* from [Cat10] where an older idea is used: regularization under the mask of Constraint

Force Mixing (CFM) [Smi06]. Recently constraint regularization has been used for particle based fluid simulation [MM13]. Another application was intended for the simulation of deformable elastic models using a constraint based formulation of the linear Finite Element Method (FEM) [SLM06]. Similar position based approaches can be found in [BML*14] and [BKCW14]. The FEM constraint approach is similar in philosophy with *continuum strain limiting* [TPS09].

The implicit integration of the equations of motion has become pervasive for cloth since the seminal work of [BW98]. The method was also applied for FEM simulation [MSJT08]. Its main attraction is its unconditional stability for very stiff equations and large time steps. By implicit integration we usually mean the Implicit Euler (IE) method, but other implicit integrators were also employed, like BDF-2 [CK02], Implicit Midpoint [OAW06] or Newmark [SSB13]. These integration methods offer better energy conservation and more responsive simulation, in contrast to IE which artificially dampens out high frequency details in exchange for stability. Other variations include approximations made to the force Jacobian [HET01] or an implicit-explicit (IMEX) approach [EEH00, BMF03]. Most approaches however use only one Newton solver iteration. More recently a new view on IE as an optimization problem was presented in [LBOK13].

A special class of integrators labeled *variational* can be deduced directly from the discretization of the Euler-Lagrange equations of motion [SD06]. They are also symplectic integrators, i.e. they preserve area in phase space, which also means they are closer to preserving energy and momenta [HLW06]. Many of them are explicit methods (e.g. Symplectic Euler, Verlet, Leapfrog) so care must be taken to the time step size. Variational implicit methods like Implicit Midpoint or Newmark are more stable and can be converted to projection schemes (e.g. through our constraint space transformation). This is why we used them as inspiration for our energy conservation strategy.

A new alternative that is totally different from implicit integration of elastic systems or our approach is *exponential integration* [MSW14] which relies on evaluating trigonometric matrices (in terms of exponential functions).

1.2 Contributions

We present in this paper a constraint based simulator that is able to reproduce fully the elastic properties of cloth. We base our results on the fact that constraint projection methods are in fact equivalent to implicit integration of stiff springs (Section 2). Our approach is not entirely new as it is based on the idea of constraint regularization [Lac07]. We chose to use a PBD method instead as it corresponds to the nonlinear case [ST96] and it handles fast deforming bodies more robustly. Catto [Cat10] uses Baumgarte stabilization (ERP) and CFM and relates them to the stiffness and damping of an implicit harmonic oscillator. We give a more general and accurate correspondence to elastic parameters.

In Section 3 we derive a simple implicit integration solver based on the Variational Implicit Euler approach in [LBOK13] and the Nonlinear Conjugate Gradient method which is very similar to PBD. From it we obtain the equations of regularized constraint projection (Section 4). Using this transformation we derive a new projection method with better energy conservation (Section 5). In Section 6 we present a novel and effective way of adding more damping to PBD. Section 7 shows how *relaxation* can be used for block solving and regularization and how to transform the Conjugate Gradient method into Conjugate Residuals. In Section 8 we present constrained mass-spring systems for cloth and how to prevent locking. In Section 9 we present a nonlinear Saint Vennant-Kirchoff (StVK) elasticity model implemented through soft constraints. We take the area and planar strain constraint from [SLM06] and derive a method that takes into account the discretization and elastic properties of cloth. The closest approach to our method is [BKCW14] but they use energy as a constraint instead of a minimization objective. [BML*14] is using the same energy objective as us but their numerical method is different.

2 OPTIMIZATION EQUIVALENCE

In this section we would like to show that implicitly integrating stiff elastic forces is no different than using a constraint based formulation with regularization. The most general way to show this is by employing an optimization formulation for both methods. Let us start with Implicit Euler:

$$\mathbf{M}\Delta\mathbf{v} = h\mathbf{f}(\mathbf{x}_0 + \Delta\mathbf{x}), \tag{1}$$

$$\Delta\mathbf{x} = h\Delta\mathbf{v} \tag{2}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{x}$ are positions, $\mathbf{v}$ are velocities, h is the time step, $\mathbf{x}_0 = \mathbf{x}^{(n)} + h\mathbf{v}^{(n)}$ and $\mathbf{f}(\mathbf{x})$ are conservative forces, i.e. $\mathbf{f}(\mathbf{x}) = -\nabla_{\mathbf{x}} U(\mathbf{x})$. We can reformulate (1) as:

$$\tfrac{1}{h^2}\mathbf{M}\Delta\mathbf{x} = -\nabla_{\mathbf{x}} U(\mathbf{x}_0 + \Delta\mathbf{x}), \tag{3}$$

This is a nonlinear equation which is typically solved using the Newton method, but it can also be regarded as the optimality condition of an optimization problem:

$$\text{minimize } \tfrac{1}{2h^2}\Delta\mathbf{x}^T\mathbf{M}\Delta\mathbf{x} + U(\mathbf{x}_0 + \Delta\mathbf{x}). \tag{4}$$

It is shown in [GHF*07] that a similar formulation can be used for a *projection* method using Lagrange multipliers and implicit constraint directions:

$$\text{min. } \tfrac{1}{2h^2}\Delta\mathbf{x}^T\mathbf{M}\Delta\mathbf{x} - \lambda^T\mathbf{c}(\mathbf{x}_0 + \Delta\mathbf{x}) + U_{ext}(\mathbf{x}^{(n)}), \tag{5}$$

where $\mathbf{c}(\mathbf{x})$ are the constraint functions that need to be zero. Note that we modified the formulation so that the external forces potential is included in the objective function. Note that the external forces are treated explicitly, i.e. using the position at the beginning of the frame, which is also the case for (4).

The potential term $U_c = \lambda^T \mathbf{c}(\mathbf{x})$ gives us the internal constraint forces. If we take the gradient of this constraint potential we get the *principle of virtual work*:

$$\mathbf{f}_c = \nabla_{\mathbf{x}} \mathbf{c}(\mathbf{x}) \lambda = \mathbf{J}^T \lambda. \tag{6}$$

If we express the total potential in general as $U = U_{int} + U_{ext}$, then so far the objectives in (4) and (5) are the same in the first (inertial) term and U_{ext}. For U_{int} we have an expression in the second case, but we have not yet specified one for the implicit integration. And we are not forced to provide one, but in reality, following the approach in [BW98], this internal potential is usually made up of quadratic elastic potentials with the purpose of enforcing certain constraints [Lan70]:

$$U_e(\mathbf{x}) = \tfrac{k}{2} \|\mathbf{c}(\mathbf{x})\|^2. \tag{7}$$

More generally we can replace stiffness k by a matrix:

$$U_e(\mathbf{x}) = \tfrac{1}{2} \mathbf{c}(\mathbf{x})^T \mathbf{E} \mathbf{c}(\mathbf{x}),$$

which is extremely useful when dealing with different stiffnesses in a mass-spring system or continuum based constitutive laws.

The potential energy in (7) gives forces of the form:

$$\mathbf{f}_e(\mathbf{x}) = -k \mathbf{J}^T \mathbf{c}(\mathbf{x}). \tag{8}$$

By comparing (6) and (8) we can see that they act in the same direction and by requiring that they have the same magnitude we obtain the regularization condition:

$$\mathbf{c}(\mathbf{x}) + \varepsilon \lambda = 0, \tag{9}$$

where $\varepsilon = 1/k$. We call this a *soft constraint* and by enforcing it we basically set the internal potential energy in (5) to be the same as in (4), thus making the two problems equivalent. It is clear now that when stiffness k goes to infinity ($\varepsilon \to 0$) implicit integration of springs becomes the constrained dynamics problem in (5). In the general case ε gets replaced by $\mathbf{E}^{-1}$.

In conclusion, not only is constraint based dynamics a limit case of implicit integration, but it can be made equivalent by replacing the strict constraint condition with a "softer" one. This permits us to solve a mass-spring system or any other discretized elastic system by casting the problem into the following form:

$$\begin{aligned} \mathbf{M}\Delta\mathbf{x} &= h^2 \left(\mathbf{J}^T \lambda - \nabla_{\mathbf{x}} U_{ext}(\mathbf{x}^{(n)}) \right), \\ 0 &= \mathbf{c}(\mathbf{x}_0 + \Delta\mathbf{x}) + \varepsilon \lambda \end{aligned}$$

This equivalence opens up a whole range of opportunities, especially for bringing results from implicit integration into the world of constraint based simulation. This was not considered possible in the past, as projection methods were regarded as an approximation of true elasticity based ones [Lac07, LBOK13].

3 NONLINEAR CONJUGATE GRADIENT SOLVER

The most important analogy we make in this paper is that between Implicit Euler integration and PBD. PBD starts from a candidate position (that includes the effect of external forces) and then runs an iterative process that does not involve the second derivative of the constraint function. This process is actually a minimization algorithm based on *sequential quadratic programming* (SQP) [WN99] that involves solving a linear system at every iteration, called *fast projection* [GHF*07]. This process can be further optimized by employing an inexact one step SOR-Newton scheme [Jak01, MHHR07] that reduces the cost of each iteration by running only one relaxation step.

The same logic can be applied to the Implicit Euler method expressed as the quadratic minimization problem in (4). If we choose the initial guess state to be one that incorporates the external forces, i.e. positions and velocities after an unconstrained step, we arrive at an approach similar to fast projection. The only difference is that the former works in configuration space, while the latter works in constraint space.

If we consider the initial candidate state consisting of $\tilde{\mathbf{v}} = \mathbf{v}^{(n)} + h\mathbf{f}_{ext}$ and $\tilde{\mathbf{x}} = \mathbf{x}^{(n)} + h\tilde{\mathbf{v}}$ we can rewrite (1) using a first order Taylor expansion around $\tilde{\mathbf{x}}$:

$$\mathbf{M}\delta\mathbf{v} = h\left(\mathbf{f}(\tilde{\mathbf{x}}) + \mathbf{K}\delta\mathbf{x}\right), \tag{10}$$

where $\mathbf{K} = \nabla_{\mathbf{x}}\mathbf{f} = -\frac{\partial^2 U}{\partial \mathbf{x}^2}$ is the *tangential stiffness matrix* and $\delta\mathbf{x} = h\delta\mathbf{v}$. Most authors choose to solve the implicit integration problem using only one Newton step, meaning we only need to solve one single linear system: $\mathbf{S}\delta\mathbf{v} = \mathbf{t}$. This works well in practice, but only if $\mathbf{K}$ contains second derivatives of the constraint function. This is because these terms contain information about the change of the constraint direction, so without them we need an iterative algorithm that keeps updating the constraint gradient. By dropping the second derivative term from $\mathbf{K}$ (see [BW98]) we get:

$$\mathbf{K} = -k\mathbf{J}^T\mathbf{J}. \tag{11}$$

This is equivalent to linearizing the force in (8) as in [EB08]. Using this formula at every Newton iteration we get the series of linear systems we need to solve:

$$(\mathbf{M} + h^2 k \mathbf{J}_i^T \mathbf{J}_i)\delta\mathbf{v}_{i+1} = h\mathbf{f}(\mathbf{x}_i), \tag{12}$$

where $\mathbf{J}_i^T = \nabla_{\mathbf{x}}\mathbf{c}(\mathbf{x}_i)$ and $\mathbf{x}_{i+1} = \mathbf{x}_i + h\delta\mathbf{v}_{i+1}$.

Uncostrained step to $\tilde{\mathbf{x}}, \tilde{\mathbf{v}}$
Compute Jacobian $\mathbf{J}$ and forces $\mathbf{f}$ using (8)
Compute residual $\mathbf{r} = \mathbf{d} = h\mathbf{f}$ and its square $\delta = \mathbf{r}^2$
for iter = 1:maxIter **do**
  Compute $\mathbf{q} = \mathbf{S}\mathbf{d} = (\mathbf{M} + h^2 k \mathbf{J}^T \mathbf{J})\mathbf{d}$
  Compute impulse $\mathbf{p} = \alpha\mathbf{d}$, where $\alpha = \delta / \mathbf{q}^T\mathbf{d}$
  Integrate: $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{p}$, $\mathbf{x} \leftarrow \mathbf{x} + h\mathbf{p}$
  Recompute Jacobian $\mathbf{J}$ and forces $\mathbf{f}$
  Compute residual $\mathbf{r} = h\mathbf{f}$ and its square $\delta' = \mathbf{r}^2$
  Compute $\beta = \delta'/\delta$ and then $\delta' \leftarrow \delta$
  Compute new search direction $\mathbf{d} = \mathbf{r} + \beta\mathbf{d}$

Algorithm 1: NCG implicit solver

Nonlinear Conjugate Gradient (NCG) [She94] is a natural solution for solving the above problem, given its linear version is very popular for solving the one Newton step approach. The only changes we need to make to linear CG is to replace the system matrix at every step with $\mathbf{S}_i = \mathbf{M} + h^2 k \mathbf{J}_i^T \mathbf{J}_i$ (the Hessian of the objective function) and the residual with $\mathbf{r}_i = h\mathbf{f}(\mathbf{x}_i)$. We use a Fletcher-Reeves formula and perform the inner line search in only one iteration - see Algorithm 1.

Note that the NCG method is not necessarily faster than traditional CG linear implicit solvers (we found that it takes roughly 40% more time without optimizations). We can also add back the second derivative term if we want. Also, visually there is no big difference between the two methods. The only advantages you would get with the NCG method are smaller spring elongations and more stability for large time steps. But the main reason for devising the scheme is the similarity with PBD which we further exploit in the next section.

4 CONSTRAINT SPACE

Given that we already know that the regularized projection method is equivalent to implicit integration and that the formulation in the previous section is already very similar to PBD, we would like to transform the system in (12) to one corresponding to fast projection. So by multiplying (12) on the left-hand side by $\mathbf{T} = \frac{1}{hk}\mathbf{A}^{-1}\mathbf{J}$, where $\mathbf{A} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T$, we get:

$$(h^2\mathbf{A} + \varepsilon\mathbf{I})\delta\lambda + \mathbf{c}(\mathbf{x}) = \mathbf{0}, \tag{13}$$

which is precisely the system we need to solve at every iteration of fast projection for the regularized constraints in (9). In order to get this result we made the substitution $\delta\mathbf{v} = h\mathbf{M}^{-1}\mathbf{J}^T\lambda$ which derives from the optimality conditions of the constraint projection optimization problem. To back our claims you can see in Figure 1 that NCG and PBD behave almost the same and very closely to the exact and CG semi-implicit solvers.

We call $\mathbf{T}$ a *constraint space transformation* from configuration space and show that the inverse transform is also possible by multiplying (13) to the left hand side by $\mathbf{Q} = hk\mathbf{M}^{-1}\mathbf{J}^T$. Note that $\mathbf{TQ} = \mathbf{I}_m$ (m - number of constraints) and $\mathbf{QT} = \mathbf{I}_n$ (n - degrees of freedom). We thus found a quick way of switching from one interpretation to the other. Also, by setting $\varepsilon \to 0$ (infinite stiffness) we recover the classic iterative projection formula:

$$h^2\mathbf{A}\delta\lambda + \mathbf{c}(\mathbf{x}) = \mathbf{0}, \tag{14}$$

Note though that not all implicit formulations can be converted this way to a constraint based formulation and that is because the methods are equivalent as optimization problems but the numerical methods used may differ in significant ways, e.g. the use of the second derivative. Still their results converge towards the same solution.

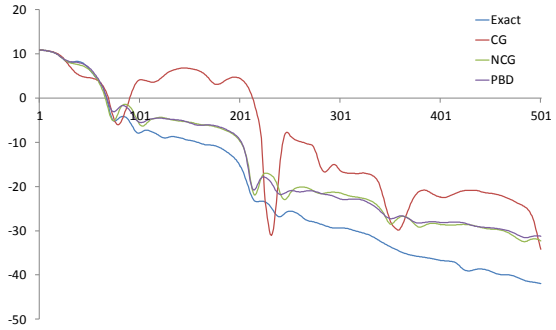


Figure 1: The energy evolution over 500 frames of a 15x15 piece of cloth using NCG (green), PBD (purple) and CG (red) and exact (blue) semi-implicit solvers.

5 ENERGY CONSERVATION

Implicit methods in general suffer from artificial numerical dissipation, whether they are used for an elasticity based formulation or a constraint based one. This is usually regarded as a good stability property and the extra damping is considered beneficial by the computer graphics community. Still in many cases like the example of cloth, this integration technique acts like a low-pass filter that removes high frequency motion and thus prevents the formation of high-detail wrinkles and responsive folds.

In the projection methods literature there exist energy preserving solutions like *symmetric projection* [HLW06] or the Velocity Verlet method proposed in [Gol10]. Another popular integration method that can conserve energy exactly is the Implicit Midpoint method [OAW06]. There exist other explicit variational integrators (e.g. Symplectic Euler) with very good energy conservation properties but they suffer from the same time step limitations as any other explicit method.

Given the fact that we are not able to solve the nonlinear equations generated by implicit methods exactly, the

accumulated errors will make Implicit Midpoint much less stable than Implicit Euler. One could alleviate this problem by using techniques suited for explicit methods, e.g. smaller/adaptive time steps or adding in a damping term. Our solution is to employ an integration scheme taken from [Lac07] which gives us more flexibility:

$$\tfrac{1}{h}\mathbf{M}\Delta\mathbf{v} = \mathbf{f}_{ext} + (1-\alpha)\mathbf{f}(\mathbf{x}^{(n)}) + \alpha\mathbf{f}(\mathbf{x}^{(n+1)}), \quad (15)$$

$$\tfrac{1}{h}\Delta\mathbf{x} = (1-\beta)\mathbf{v}^{(n)} + \beta\mathbf{v}^{(n+1)}, \quad (16)$$

where α and β are between 0 and 1 and we can distinguish the following special cases: Explicit Euler ($\alpha=\beta=0$), Implicit Euler ($\alpha=\beta=1$), Implict Midpoint ($\alpha=\beta=\frac{1}{2}$), and Symplectic Euler ($\alpha=0,\beta=1$ or $\alpha=1,\beta=0$).

Moving slightly away from the Implicit Midpoint method and making it more implicit permits us to have low artificial numeric dissipation while still being able to solve the system approximately and obtain a stable, yet responsive simulation. Using the constraint space transformation , i.e. multiplying (15) and (16) on the left hand side by $\mathbf{T}$, we obtain the following regularized projection:

$$(h^2\alpha\beta\mathbf{A}_i + \varepsilon\mathbf{I})\delta\lambda_{i+1} + \alpha\mathbf{c}(\mathbf{x}_i) = \mathbf{0}, \quad (17)$$

where $\mathbf{x}_{i+1} = \mathbf{x}_i + \beta h\mathbf{M}^{-1}\delta\mathbf{f}_{i+1}$, $\mathbf{v}_{i+1} = \mathbf{v}_i + h\mathbf{M}^{-1}\delta\mathbf{f}_{i+1}$ and $\delta\mathbf{f}_{i+1} = \mathbf{J}_i^T\delta\lambda_{i+1}$. Also the integration of the candidate state changes to:

$$\tilde{\mathbf{x}} = \mathbf{x}^{(n)} + h\mathbf{v}^{(n)} + h^2\mathbf{g} + h^2\beta(1-\alpha)\mathbf{M}^{-1}\mathbf{J}^T\lambda^{(n)},$$
$$\tilde{\mathbf{v}} = \mathbf{v}^{(n)} + h\mathbf{g} + h(1-\alpha)\mathbf{M}^{-1}\mathbf{J}^T\lambda^{(n)},$$

where we replaced external force by gravity for brevity and the Jacobian $\mathbf{J}$ is computed at position $\mathbf{x}^{(n)}$.

At the limit $\varepsilon \to 0$ this whole procedure is of course equivalent to projecting the candidate positions using the same system (14) as in fast projection and PBD. The difference appears only in the regularization term which gets replaced by $\varepsilon\alpha^{-1}\beta^{-1}$. The above formulation is also good when stepping both positions and velocities at the same time. Alternatively we could estimate the new velocity only at the end using equation (16):

$$\beta\mathbf{v}^{(n+1)} = \tfrac{1}{h}(\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}) - (1-\beta)\mathbf{v}^{(n)}$$

We could have reached a similar result using the fast projection formalism in [Gol10] but our constraint space transformation method ensures that we also obtain the correct regularization term. For example, we can apply the same technique to obtain the BDF-2 based projection method presented in [EB08] and find that the new regularization term is $\frac{9}{4}\varepsilon$. Still we prefer our method as we do not need to store previous positions and velocities and, being a one step scheme, it is better suited for non-smoothness. Actually more related to ours is the Newmark scheme [SSB13] for which we can use the same projection method regularized with ε/β, using the Newmark β factor between 0 and $1/2$.

6 DAMPING

Now that we have reduced the amount of artificial damping, we can add back some real damping. Our method will be based on the damping force expression used in [BW98] which is also a special case of the widely used method of *Rayleigh damping* [SSB13]. In order to extend (10) to contain the damping force we need to consider the total force as $\mathbf{f}(\mathbf{x},\mathbf{v}) = \mathbf{f}_e(\mathbf{x}) + \mathbf{f}_d(\mathbf{v})$, i.e. sum of elastic and damping forces:

$$(\mathbf{M} - h^2\mathbf{K} - h\mathbf{D})\delta\mathbf{v} = h\mathbf{f}(\tilde{\mathbf{x}}),$$

where $\mathbf{D} = \frac{\partial\mathbf{f}}{\partial\mathbf{v}} = \nabla_{\mathbf{v}}\mathbf{f}$. This is equivalent to having a damping force:

$$\mathbf{f}_d = d\nabla_{\mathbf{x}}\mathbf{c}(\mathbf{x})\dot{\mathbf{c}}(\mathbf{x}) = d\mathbf{J}^T\mathbf{J}\mathbf{v},$$

where d is the damping coefficient and $\dot{\mathbf{c}}(\mathbf{x}) = \mathbf{J}\mathbf{v}$.

Rayleigh damping makes the approximation $\mathbf{D} = \zeta\mathbf{M} + \gamma\mathbf{K}$, but we will only be using the second term as it makes the derivations simpler and it is only damping along the constraints we are interested in (we can achieve drag friction in other ways). The implicit integration formula (10) now becomes:

$$(\mathbf{M} - h(h+\gamma)\mathbf{K})\delta\mathbf{v} = h\mathbf{f}(\tilde{\mathbf{x}}). \quad (18)$$

If using the approximation in (11) we notice that $\mathbf{f}_d = \gamma\mathbf{K}\mathbf{v}$, where $\gamma = d/k$ is the ratio between the damping and the stiffness coefficients.

We can incorporate this damping force into the optimization formulation using *Rayleigh dissipation functions* [Lac07]. So we transform (18) to constraint space and get the following projection:

$$h(h+\gamma)\mathbf{A}\delta\lambda_{i+1} + \mathbf{e}_i = \mathbf{0}, \quad (19)$$

where $\mathbf{e}_i = \mathbf{c}(\mathbf{x}_i) + \gamma\mathbf{J}\mathbf{v}_i$; the second term is nothing more than the relative velocity along the constraint times γ. So this is a simple way to add more damping along the constraint directions which looks more natural than plain viscous drag in all directions. The downside is that you may have to pay the price of more iterations in order to keep the same amount of constraint violation as before damping. The final formula after adding regularization and energy preservation is:

$$(h\alpha(h+\gamma)\mathbf{A}_i + \varepsilon\mathbf{I})\delta\lambda_{i+1} + \alpha\mathbf{e}_i = \mathbf{0}. \quad (20)$$

As you can see now the projection formula includes both a stiffness parameter ($\varepsilon = 1/k$) and a damping parameter ($\gamma = d/k$) and so the method is fully equivalent to the implicit integration of a damped elastic system. Note that even in the case of infinite stiffness, the damping parameter can remain finite and we get (19).

7 ITERATIVE SOLVERS

As we already mentioned, the most popular methods for solving PBD are nonlinear relaxation methods: Jacobi, Gauss-Seidel or Successive Over Relaxation (SOR). Jacobi for instance has the following update formula: $\delta\lambda_{i+1} = \delta\lambda_i + \mu\mathbf{r}_i$, where $\mu_j = \omega/A_{jj}$, $\omega < 1$ and A_{jj} comes from the diagonal of $\mathbf{A}$. What these methods actually do is they solve each equation and its corresponding unknown separately (local solve) and iterate over all equations for a number of times in order to refine the solution. The local solve formula for one constraint c in the PBD method is:

$$h^2\nabla c(\mathbf{x})^T\mathbf{M}\nabla c(\mathbf{x})\delta\lambda + c(\mathbf{x}) = 0, \qquad (21)$$

where $\mathbf{M}$ and $\mathbf{x}$ correspond to the $s \geq 1$ particles involved in the constraint. We could also solve for more than one constraints simultaneously and we would obtain a system $\mathbf{L}\delta\lambda + \mathbf{c}(\mathbf{x}) = \mathbf{0}$ that we could solve using direct matrix inversion or a direct solver.

Looking more closely at the equation (13) we notice that it is not that different from (14). We can regard the change to the diagonal of the system matrix as a scaling through a relaxation factor as in [Jak01]. As noted in [MHHR07] this factor is highly nonlinear and we are now able to express this exact nonlinear relationship to the linear spring stiffness value: $\omega_j = (1 + (h^2 k_j A_{jj})^{-1})^{-1} < 1$, where k_j is the stiffness of spring j. On the other hand, if we use $\omega > 1$ we obtain SOR which may converge faster and produce better inextensibility, i.e. stiffer cloth for less iterations.

Another application of the equivalence between implicit integration and regularized PBD is to see what happens to the Conjugate Gradient method when we transform from configuration to constraint space. Let us start with the update formula for Steepest Descent:

$$\delta\mathbf{v}_{i+1} = \delta\mathbf{v}_i + \alpha_i\rho_i,$$

where ρ_i is the residual of the system in (12) which is related to the residual $\mathbf{r}_i$ by $\rho_i = \mathbf{T}\mathbf{r}_i$ and $\mathbf{r}_i = \mathbf{Q}\rho_i$, and

$$\alpha_i = \frac{\rho_i^T\rho_i}{\rho_i^T\mathbf{S}\rho_i}.$$

We would like to see how the constraint space transformation affects the update formula in constraint space. So we write α_i in terms of $\mathbf{r}_i$:

$$\alpha_i = \frac{\mathbf{r}_i^T\mathbf{Q}^T\mathbf{Q}\mathbf{r}_i}{\mathbf{r}_i^T\mathbf{Q}^T\mathbf{S}\mathbf{Q}\mathbf{r}_i}.$$

For the nominator we find that $\mathbf{Q}^T\mathbf{Q} = h^2k^2\tilde{\mathbf{A}}$, where $\tilde{\mathbf{A}} = \mathbf{J}\mathbf{M}^{-2}\mathbf{J}^T$ and the denominator is $\mathbf{Q}^T\mathbf{S}\mathbf{Q} = h^2k^3\mathbf{A}\mathbf{S}$. By using the properties of matrices $\mathbf{T}$ and $\mathbf{Q}$ we get that $\mu_i = k\alpha_i$ and by considering infinite stiffness, i.e. $\mathbf{S} \to h^2\mathbf{A}$ when $\varepsilon \to 0$, we get:

$$\mu_i = \frac{\mathbf{r}_i^T(h^2\tilde{\mathbf{A}})\mathbf{r}_i}{\mathbf{r}_i^T(h^2\mathbf{A})^2\mathbf{r}_i}.$$

If we ignore the tilde (or all the masses are 1) we see that we have obtained the formula for the Minimum Residual method which was used in [FM14]. It may be that the method breaks when $\mathbf{M}^{-1}$ is far different from its square and the correct formula needs to be used. Still we think this derivation is a strong argument for why CG does not work in constraint based methods and we need to transform it to a method that looks more like a minimum residual version of CG, e.g. Conjugate Residuals (CR) [Saa03].

8 CLOTH MODEL

The most straightforward application of the presented methods to cloth is through the use of a model made of particles and springs or rigid links connecting them. Such links correspond to a constraint function like $\mathbf{c}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| - l_{ij}$, where l_{ij} is the initial rest length of the link and i and j are the indices of the two particles. Using three types of links for stretching, shearing and bending, one can obtain a full model of cloth. Details on how to build such links for quad meshes are given in many papers [Pro96, OAW06]. The main advantage of our method of *soft constraints* is that the stiffness of each constraint can now be expressed naturally as an elastic parameter (related to Young's modulus) instead of using non-linear attenuation factors like in [Jak01] and [MHHR07].

Figure 2: Simulation of a cloth model consisting of 6910 vertices and 13674 triangles using soft constraints

For irregular triangle meshes, one can also build bending links [Erl05], but one cannot distinguish between stretching and shearing links anymore. For quad meshes shearing is simple to express and is usually less stiff than stretching. Using the same stiffness coefficient for shearing as for stretching (infinite in the

case of PBD) leads to locking artifacts, as the cloth does not have enough degrees of freedom to move naturally. Lowering the stiffness value may help with the locking problem, but this causes stretching, which must be avoided at all cost for cloth.

Many other solutions have been proposed for locking [EB08], but we chose to use the model presented in [BW98] which separates out the stretching components (warp and weft) from shearing for each triangle. In general a constraint involving a number of particles implies calculating gradients corresponding to each particle $\nabla_i c(\mathbf{x}) = \frac{\partial c(\mathbf{x})}{\partial \mathbf{x}_i}$. Then we can solve that constraint independently using (21):

$$\delta\lambda = -\frac{c(\mathbf{x})}{\sum_{i=1}^{s} m_i^{-1} \|\nabla_i c(\mathbf{x})\|^2} = -\frac{c(\mathbf{x})}{\xi}. \tag{22}$$

The action of the bending links is dependent on stretching so we might want to use other measures for the curvature of the cloth. We could use directly the constraint between two triangles (4 vertices) defined in [MHHR07], as it expresses the same dihedral angle as in [BW98]. Still we chose to derive our own formulas using (22) and the assumption made in [BW98] that the lengths of the normals remain constant.

9 FINITE ELEMENT METHOD

The continuum formulation in [BW98] is actually a special treatment of the finite element method. The three constraints correspond to the strain components ε_{uu}, ε_{vv} and ε_{uv} that make up the planar symmetric Green-Lagrange strain tensor:

$$\varepsilon(\mathbf{x}) = \tfrac{1}{2}(\nabla\mathbf{w}\nabla\mathbf{w}^T - \mathbf{I}), \tag{23}$$

where $\mathbf{w} : \mathbb{R}^2 \to \mathbb{R}^3$ is a mapping from an undeformed mesh parametrization (u, v coordinates) to deformed vertex positions. Then the actual components are:

$$\varepsilon_{uu} = \tfrac{1}{2}(\mathbf{w}_u^T\mathbf{w}_u - 1), \tag{24}$$

$$\varepsilon_{vv} = \tfrac{1}{2}(\mathbf{w}_v^T\mathbf{w}_v - 1), \tag{25}$$

$$\varepsilon_{uv} = \varepsilon_{vu} = \mathbf{w}_u^T\mathbf{w}_v, \tag{26}$$

where by the subscript of $\mathbf{w}$ we signify partial derivation with respect to to u and v. By considering strain constant over a triangle (linear FEM) we can derive simple formulas for $\mathbf{w}_u$ and $\mathbf{w}_v$ like in [BW98] or [VMTF09].

The integral of the strain energy over a triangle is:

$$U_{\text{fem}} = \tfrac{a}{2}\hat{\varepsilon}(\mathbf{x})^T\mathbf{E}\hat{\varepsilon}(\mathbf{x}), \tag{27}$$

where a is the area of the triangle, $\hat{\varepsilon}^T = (\varepsilon_{uu}, \varepsilon_{vv}, \varepsilon_{uv})$ and $\mathbf{E}$ is a matrix that depends on the Young modulus E and the Poisson ratio ν (or equivalently on the Lamé coefficients) like the one given in [VMTF09] or [TWS07]. Note that the former expresses isotropic elasticity while the latter expresses orthotropic elasticity, i.e. different stiffness along warp and weft directions.

We use (27) to derive the true constraint function using the regularization framework as in [SLM06]:

$$\mathbf{c}(\mathbf{x}) = a^{\frac{1}{2}}\hat{\varepsilon}(\mathbf{x}). \tag{28}$$

The resulting three constraints (c_u, c_v, c_s) are are similar to the ones in [BW98] and their gradients form the Jacobian $\mathbf{J}_{\text{fem}} = (\nabla c_u^T, \nabla c_v^T, \nabla c_s^T)$. Note that in the most rigorous approach the area of the triangle $a(\mathbf{x})$ is also varying and its derivative should also be considered. We chose not to do so in our computations, but alternatively we could add an extra area constraint. Some authors use area and volume constraints together with edge constraints to improve on mass-spring soft body models [THMG04].

Now we can formulate the regularization condition:

$$a^{\frac{1}{2}}\hat{\varepsilon}(\mathbf{x}) + \mathbf{E}^{-1}\lambda = \mathbf{0}. \tag{29}$$

In the end we can apply the block local solve formula from Section 7, which is equivalent to other StVK linear FEM approaches like the one in [VMTF09]. We choose to apply this block approach only for the stretch components together, as the shear stress component is related only through a diagonal term to strain, and thus decoupled from the normal directions. The resulting 2x2 local linear system for the two stretching constraints is:

$$(h^2\mathbf{A} + \tilde{\mathbf{E}}^{-1})\delta\lambda + \hat{\varepsilon}(\mathbf{x}) = 0,$$

where in the case of isotropic materials

$$\tilde{\mathbf{E}}^{-1} = \frac{1}{E\sqrt{a}}\begin{pmatrix} 1 & \nu \\ -\nu & 1 \end{pmatrix}.$$

Notice that we divided equation (29) by the constant area term $a^{1/2}$ and obtained a CFM matrix that contains all the relevant continuous material parameters: Young's modulus, Poisson ratio and the triangle area (discretization measure). We can also add damping through the Rayleigh damping technique presented in Section 6 or the projection in Section 5 for better energy conservation. In the end we obtain a very accurate (iterations permitting) and physically correct model for simulating thin nonlinear elastic materials like the one in [VMTF09] based only on constraints (Figure 3).

10 RESULTS

We implemented a cloth simulator solely on a modular constraints architecture using C++ (single threaded). Depending on the level of accuracy or performance the user can choose between different constraint types, e.g. links or FEM triangles for stretching and shearing, different types of bending constraints, static collision or

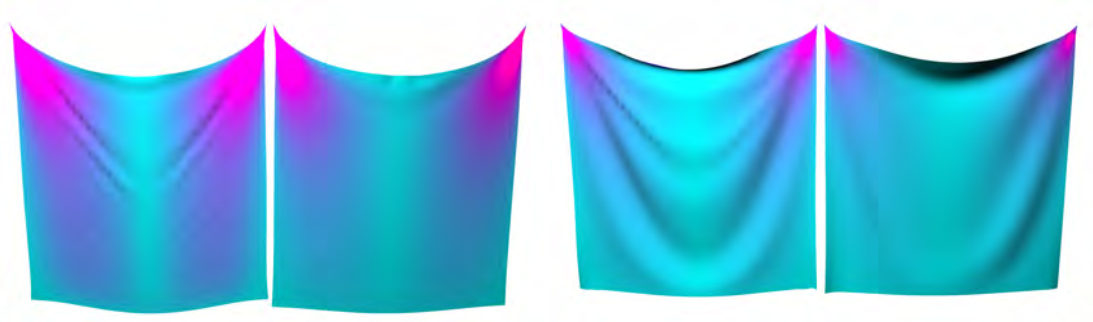


Figure 3: Two snapshots of a side by side real-time simulation of two 40x40 cloth pieces with the same Young's modulus E: regularized FEM constraints (left) and soft links (right); superimposed in purple is the strain map. FEM offers more realistic folds and the strain is better distributed throughout the cloth.

self collision constraints etc. Note that all these constraints are treated in the same solver loop. Regularization was implemented by modifying the diagonal of the system matrix using equation (13) or as described in Section 7. The resulting scheme can be modified to use the projection in Section 5 or we could add more damping (Section 6). These options can be added depending on the needs of the simulator and we denote them collectively as soft constraint methods for enhancing PBD. This also is why we do not provide any pseudo-code and hope that the readers will assemble themselves the simulator of choice.

Given our simulator is fully constraint based our collision response techniques are the same as the ones used in [Jak01, MHHR07] and for self collision we adapted the methods in [BFA02]. Friction is treated more accurately by being solved at every contact iteration in a similar fashion to cone complementarity programming [TA10]. We implemented cloth-mesh collisions by testing for triangle-point and edge-edge intersections between two triangles. For acceleration we used AABB trees for both the static mesh (pre-computed) and the cloth (rebuilt at every frame). A similar approach was used for accelerating self-collisions too.

We tested simulations mostly visually looking for obvious artifacts like jittering or instability. Our most common test scenario was a piece of cloth hanging by two corners, falling from a horizontal or vertical position, with different parameters or tessellation. Given the multitude of methods used and the differences between them it is hard to find a metric that measures well the quality of the simulation. We opted to measure the total energy - kinetic and potential (gravitational and elastic) and no damping, and chart its evolution in time (Figure 4). The NCG solver behaves well and has good convergence, but decays non-monotonically. The regularized PBD method is smoother, dissipates energy slower, but the Gauss-Seidel solver is less accurate. Energy preserving projection with $\alpha = \beta = 0.6$ offers even slower energy decay while the higher energy line is due to the kinetic energy of the oscillation. This is a good energy preserving property but it looks jittery and unnatural and we may need to add extra (non-artificial) damping. The closer we get to $\alpha = \beta = 0.5$ the more horizontal the graph becomes, i.e. full energy conservation, but this is dangerous territory for stability. We get the same results with a Symplectic Euler integrator with very small time steps and a small damping factor.

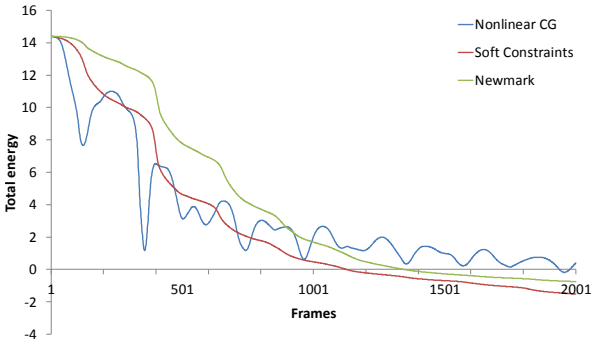


Figure 4: Total energy evolution in time for the simulation of a 10x10 rubber cloth (k = 2 N/m, 25 iterations) using NCG implicit integration (blue), regularized PBD (red) and regularized energy preserving projection (green).

For our damping method we measured the total energy minus the elastic potential in order to give a clearer picture of the velocity reduction (Figure 5). As you can see a damping factor of $\gamma = 10h$ gives a significant energy dissipation compared to soft projection (or PBD just as well). Reaching this level of dissipation so quickly is not possible using the method we compared against, i.e. reducing the relative velocity along the constraint direction (basically velocity projection). We set for the energy preserving projector $\alpha = \beta = 0.55$ and $\gamma = h$ in order to obtain as little artificial damping as possible while at same time damping the simulation just a bit less than PBD would normally do.

All simulations were performed in real time at 60 Hz, i.e. under 16 ms of computation time depending on

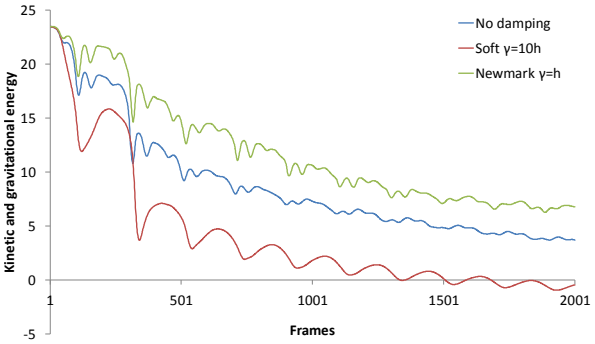


Figure 5: Damping response for the simulation of a 40x40 piece of cloth ($k = 2000$ N/m, 25 iterations) using regularized PBD (blue), aggressive damping (red) and slightly damped energy preserving projection (green).

cloth size, and with lower framerate for the dress in Figure 2 (up to 20 ms or more for the solver only).

11 CONCLUSIONS

We have shown that implicit integration of elastic systems is equivalent as an optimization problem to fast projection. Based on the analogy to PBD we derived a Nonlinear Conjugate Gradient implicit solver. Its drawback is that it is using an approximated force Jacobian but this is compensated by running more than one inexact Newton iterations.

After developing a method of switching between the two representations (configuration and constraint space) we proved that the regularized PBD method (soft constraints) replicates elastic behavior. We also showed how to preserve energy better or how to dissipate more when solving constraints. Note though that the viscous drag term of the Rayleigh damping matrix cannot be treated implicitly in this framework. Also, one can use a parallel version of the Conjugate Residuals algorithm to speed up the simulation. Finally we showed that accurate FEM simulation of cloth using constraints is possible and is no different from implicit integration. We believe that these are new and useful results for PBD.

We hope to use the Kawabata evaluation system in the future for real fabric modeling like in [VMTF09]. We also intend to optimize our simulator using parallel algorithms for multi-core and GPGPU.

12 ACKNOWLEDGMENTS

We would like to thank Framebreed animation studio for the 3D model of the dress and the body in Figure 2. We also want to thank Lucian Petrescu and Dongsoo Han for their input. Part of the research presented in this paper was supported by the Sectoral Operational Program Human Resources Development, 2014-2015, of the Ministry of Labor, Family and Social Protection through a Financial Agreement between University POLITEHNICA of Bucharest and AM POS DRU Romania, project ID 132395.

13 REFERENCES

[AH04] M. Anitescu and G. D. Hart. A Constraint Stabilized Time-Stepping Approach for Rigid Multibody Dynamics with Joints, Contact and Friction. Int. J. Numer. Meth. Eng., 60(14):2335–2371, 2004.

[Bar94] D. Baraff. Fast Contact Force Computation for Nonpenetrating Rigid Bodies. In ACM SIGGRAPH 1994 Papers, pages 23–34, 1994

[BW98] D. Baraff and A. Witkin. Large Steps in Cloth Simulation. In ACM SIGGRAPH 1998 Papers, pages 43–54, 1998.

[BKLS95] E. Barth, K. Kuczera, B. Leimkuhler, and R. D. Skeel. Algorithms for constrained molecular dynamics. J. Comp. Chem, 16:1192–1209, 1995.

[BKCW14] J. Bender, D. Koschier, P. Charrier, and D. Weber. Position-based simulation of continuous materials. Computers & Graphics, 44:1–10, 2014.

[BML*14] S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly. Projective dynamics: Fusing constraint projections for fast simulation. ACM Trans. Graph., 33(4):154:1–154:11, 2014.

[BFA02] R. Bridson, R. Fedkiw, and J. Anderson. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. In ACM SIGGRAPH 2002 Papers, pages 594–603, 2002.

[BMF03] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 28–36, 2003.

[Cat10] E. Catto. Soft Constraints: Reinventing the Spring. Game Developer Conference, 2010.

[CK02] K.-J. Choi and H.-S. Ko. Stable but responsive cloth. In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '02, pages 604–611, 2002.

[EEH00] B. Eberhardt, O. Etzmuß, and M. Hauth. Implicit-explicit schemes for fast animation with particle systems. In In Eurographics Computer Animation and Simulation Workshop, pages 137–151, 2000.

[EB08] E. English and R. Bridson. Animating developable surfaces using nonconforming elements. ACM Trans. Graph., 27(3):66:1–66:5, 2008.

[Erl05] K. Erleben et al. Physics-based Animation. Charles River Media, 2005.

[Erl07] K. Erleben. Velocity-based Shock Propagation for Multibody Dynamics Animation. ACM Trans. Graph., 26, 2007.

[FM14] M. Francu and F. Moldoveanu. An improved jacobi solver for particle simulation. In VRIPHYS 14, pages 125–134, 2014.

[Gol10] A. R. Goldenthal. Implicit Treatment of Constraints for Cloth Simulation. PhD thesis, 2010.

[GHF*07] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun. Efficient Simulation of Inextensible Cloth. In ACM SIGGRAPH 2007 Papers, 2007.

[HLW06] E. Hairer, C. Lubich, and G. Wanner. Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations. Springer, 2006

[HET01] M. Hauth, O. Etzmuss, and U. Tübingen. A high performance solver for the animation of deformable objects using advanced numerical methods. In In Proc. Eurographics 2001 (2001), pages 319–328, 2001.

[HCJ*05] M. Hong, M.-H. Choi, S. Jung, S. Welch, and J. Trapp. Effective constrained dynamic simulation using implicit constraint enforcement. In Robotics and Automation, ICRA 2005, pages 4520–4525, 2005.

[Jak01] T. Jakobsen. Advanced Character Physics. In Game Developers Conference Proceedings, pages 383–401, 2001.

[Lac07] C. Lacoursière. Ghosts and machines: regularized variational methods for interactive simulations of multibodies with dry frictional contacts. PhD thesis, Umeå University, 2007.

[Lan70] C. Lanczos. The Variational Principles of Mechanics. Dover Publications, 1970.

[LBOK13] T. Liu, A. W. Bargteil, J. F. O'Brien, and L. Kavan. Fast simulation of mass-spring systems. ACM Trans. Graph., 32(6):214:1–214:7, 2013.

[MM13] M. Macklin and M. Müller. Position based fluids. ACM Trans. Graph., 32(4):104:1–104:12, July 2013.

[MSW14] D. L. Michels, G. A. Sobottka, and A. G. Weber. Exponential integrators for stiff elastodynamic problems. ACM Trans. Graph., 33(1):7:1–7:20, Feb. 2014.

[MHHR07] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position Based Dynamics. J. Vis. Comun. Image Represent., 18(2):109–118, 2007.

[MSJT08] M. Müller, J. Stam, D. James, and N. Thürey. Real time physics: Class notes. In ACM SIGGRAPH 2008 Classes, pages 88:1–88:90, 2008.

[OAW06] S. Oh, J. Ahn, and K. Wohn. Low damped cloth simulation. Vis. Comput., 22(2):70–79, 2006.

[Pro96] X. Provot. Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior. In In Graphics Interface, pages 147–154, 1996.

[Saa03] Y. Saad. Iterative Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, 2nd edition, 2003.

[SLM06] M. Servin, C. Lacoursière, and N. Melin. Interactive simulation of elastic deformable materials. In SIGRAD 2006 Conference Proceedings, pages 22–32, 2006.

[She94] J. R. Shewchuk. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. Technical report, Pittsburgh, PA, USA, 1994.

[SSB13] F. Sin, D. Schroeder, and J. Barbic. Vega: Nonlinear fem deformable object simulator. Comput. Graph. Forum, 32(1):36–48, 2013.

[Smi06] R. Smith. Open dynamics engine v0.5 user guide. 2006.

[Sta09] J. Stam. Nucleus: Towards a unified dynamics solver for computer graphics. In Computer Aided Design and Computer Graphics, pages 1–11, 2009.

[SD06] A. Stern and M. Desbrun. Discrete geometric mechanics for variational time integrators. In ACM SIGGRAPH 2006 Courses, SIGGRAPH '06, pages 75–80, 2006.

[ST96] D. Stewart and J. C. Trinkle. An Implicit Time-Stepping Scheme for Rigid Body Dynamics with Coulomb Friction. Int. J. Numer. Meth. Eng., 39:2673–2691, 1996.

[TA10] A. Tasora and M. Anitescu. A Convex Complementarity Approach for Simulating Large Granular Flows. J. Comput. Nonlinear Dynam., 5, 2010.

[THMG04] M. Teschner, B. Heidelberger, M. Muller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In Proceedings of the Computer Graphics International, pages 312–319, 2004.

[TPS09] B. Thomaszewski, S. Pabst, and W. Straßer. Continuum-based strain limiting. Comput. Graph. Forum, 28(2):569–576, 2009.

[TWS07] B. Thomaszewski, M. Wacker, and W. Straßer. Advanced topics in virtual garment simulation – part 1, 2007.

[VMTF09] P. Volino, N. Magnenat-Thalmann, and F. Faure. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. ACM Trans. Graph., 28(4):105:1–105:16, 2009.

[WN99] S. J. Wright and J. Nocedal. Numerical optimization, volume 2. Springer New York, 1999.

Hybrid ROI-Based Visualization of Medical Models

Lazaro Campoalegre
Trinity College Dublin
College Green
Dublin 2, Ireland
campoall@tcd.ie

Isabel Navazo
Universitat Politècnica de Catalunya
C. Jordi Girona 31
08034 Barcelona, Spain
isabel@cs.upc.edu

Pere Brunet
Universitat Politècnica de Catalunya
C. Jordi Girona 31
08034 Barcelona, Spain
pere@cs.upc.edu

ABSTRACT

There is an increasing interest on tele-medicine and tele-diagnostic solutions based on the remote inspection of volume data coming from multimodal imaging. Client-server architectures meet these functionalities. The use of mobile devices is sometimes required due to the portability and easy maintenance. However, transmission time for the volumetric information and low performance hardware properties, make quite complex the design of efficient visualization systems on these devices. In this paper, we present a hybrid approach which is based on regions of interest (ROIs) and on a transfer-function aware compression scheme. It has a good performance in terms of bandwidth requirements and storage needs in the client device, being flexible enough to represent several materials and volume structures in the ROI. Clients store a low-resolution version of the volume data and ROI-dependent high resolution segmented information. Data must be only sent whenever a new ROI is requested, but interaction in the client is autonomous - without any data transmission - while a certain ROI is inspected. A benchmark is presented to compare the the proposed scheme with three existing approaches, on two different volume data models. The results show that our hybrid approach is compact, efficient and scalable, with compression rates that decrease when the size of the volume model increases.

Keywords

Volume rendering, client-server, mobile devices, medical data, region of interest, ray-casting, volume data compression

1 INTRODUCTION

Recently, several important research areas in three-dimensional techniques for multimodal imaging have appeared. Applications include neurological imaging for brain surgery, tissue characterization, medical school teaching, plastic surgery and others. At the same time, scientists are more familiarized with three-dimensional structures reconstruction from Two-dimensional images.

The reconstruction of a volumetric model is generally achieved by using a voxel representation of datasets. According to the structure to be highlighted during the visualization, a transfer function is applied to assign color and opacity to the density value which represents the structure properties.

The handling of three-dimensional information requires efficient systems to achieve fast data transmission and interactive visualization of high quality images. Client-server applications allow these functionalities. Sometimes the use of mobile devices is necessary due to the portability and easy maintenance. However, transmission time for the volumetric information and low performance hardware properties, complicate the design of efficient visualization systems on these devices.

The main contribution of our work is a Hybrid visualization approach that inherits the advantages of some previous algorithms like the ones presented in [1] and [2], while keeping a good performance in terms of bandwidth requirements and storage needs in client devices. The scheme is flexible enough to represent several materials and volume structures in the Region of Interest (ROI) at high resolution and very limited information transmission cost.

2 PREVIOUS WORK

Client-server architectures have grown in popularity. Mobile devices as well as desktop computers can both function as clients requesting and receiving information over the network. Many authors have published research results in the remote volume visualization area. However there is still scarce specific bibliography for volume visualization in mobile devices. The majority of the proposals use known algorithms like Ray-Casting, 2D Textures, and isosurface modeling to render volume data. In order to compensate limitations in low performance devices or to reduce costs, the number of client-server schemes have been proposed.

In some client-server approaches the dataset is compressed on the server side and sent to the client where

the transfer function is applied after decompression and before the rendering of the recovered data. Moser and Weiskopf [3] proposed a 2D texture-based method which uses compressed texture atlas to reduce interpolation costs. Nogera et al. [4] proposed a webGL application to visualize very large 3D volumes by using multi-texturing to encode volumetric models on a set of RGBA 2D textures. A recent application developed by Balsa et al. [5] allows to interact with volume models using mobile devices hardware. Their scheme is not compressing the volume data.

In other schemes, the transmitted data is a compressed image, the transfer function is applied at the beginning of the pipeline, followed by a 2D rendering on a texture, all done on the server side. A compressed image is sent to the client where decompression and image rendering takes place. This scheme is frequently named "Thin Clients" [6]. The idea in multiresolution model schemes [7] is to render only a region of interest at high resolution and to use progressively low resolution when moving away from that region. Both bricking and multiresolution approaches [8] need a high memory capacity on the CPU for storing the original volume dataset. Moreover, bricking requires a high amount of texture transfers as each brick is sent once per frame; multiresolution techniques have been built for CPU purposes and its translation to GPUs is not straightforward due to the required number of texture accesses.

Preprocessing of data is also a useful technique, as it ensures the reduction of the information, combined with different techniques for quantization, encoding and multiresolution representation [8].

Efficient schemes require optimized algorithms to reduce and send data through the network. The algorithms must achieve the maximum compression possible while allowing an easy decompression in the client side, where sometimes hardware and memory constraints decrease performance [8].

Wavelet transforms offer considerable compression ratios in homogeneous regions of an image while conserving the detail in non-uniform ones. The idea of using 3D wavelets for volume compression was introduced by Muraki [9]. Ihm and Park [10] proposed an effective 3D 16^3-block-based compression/decompression wavelet scheme for improving the access to random data values without decompressing the whole dataset. Guthe et al. [11] proposed a novel algorithm that handles a hierarchical wavelet representation where decompression takes place in GPU.

Some techniques advocate the use of hybrid region-based volume rendering, by applying different shading algorithms inside the volume model [12], or by implementing multiresolution region-based schemes [1]. Luo *et al.* [13] developed a technique for focusing on a user-driven ROI while preserving context information. The approach uses a distance function to define the region of interest. This function controls voxel opacity, exploits silhouette enhancement and non-photorealistic shading.

In this paper, we propose a hybrid framework that exploits the use of standard transfer functions as an alternative to compress volume dataset. Our scheme is a transfer function-aware scheme for client/server techniques. It combines Wavelet-preprocessed volume data to reduce information outside the ROI, and highlighted segmented data in regions of interest (ROI), (Gradient Octree shceme). From the best of our knowledge this possibility has not been considered by any of the described approaches in this previous work.

3 OVERVIEW OF THE APPROACH

Let us assume that we are interested in inspecting a volume data model V which is too large in terms of network transmission and/or client storage facilities. Wavelet compression algorithms like the ones presented in [1, 10, 11] are able to support block-based regions of interest (ROIs). Other approaches like Gradient Octrees [2] can be rendered with advanced illumination models and at a higher visual quality level. Gradient Octrees are specific data structures for multiresolution volume datasets. Gradient Octrees $G(V)$ include an specific data structure S and a compact data array D. The octree structure S can be sent to the client devices in a lossless way with only one bit per node, whereas data is compacted to 3 Bytes per octree node, including material information and volume gradients. Both approaches, however, have advantages and drawbacks:

(I) TF-aware wavelet compression schemes succeed in sending to the clients a very limited amount of information in the areas outside the region of interest (ROI). High quality volume information in the ROI is also compact [1], because the 3D texture is smaller and restricted to the blocks in the ROI area. Ray-casting visualization in the client can use compact 3D textures which are suitable for many client devices. The main drawback of this approach, however, is twofold. First, changing the transfer function requires sending a new version of the compress volume model to the client, and second, it is not well suited for illumination computations that would require too many texture accesses.

(II) Approaches like Gradient Octrees overcome these limitations by supporting multiple transfer functions and materials and by precomputing gradients on the server. They support advanced illumination models, thus achieving a higher visual quality level. However, they are not direct candidates for ROI-based visualization paradigms, as their low level volume representations present a flat-face appearance with poor gradients. These representations at coarse tree levels are

well suited for progressive transmission but they perform worse than similar-quality low-level wavelet reconstructions.

Our hybrid scheme inherits the best of both approaches. In this case, apart from the volume model V, the user must supply a set of transfer functions $\{TF_k\}$ and selects one of them as a canonical transfer function. The server starts by computing a Gradient Octree $G(V)$ from V and for the set $\{TF_k\}$, also computing the quantified representation $W(V)$ of the wavelet transform of V with the canonical transfer function, as described in Section 4. $G(V)$ encodes materials and gradients only in the subset of voxels of V which are relevant to same of the transfer functions in $\{TF_k\}$.

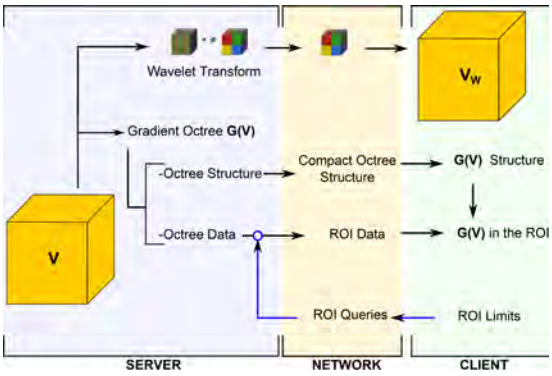


Figure 1: Overview of the proposed scheme, showing the preprocess on the server, the data transfer through the network and the data structures in the client device

Users at the client side can interactively define regions of interest, ROIs. Information over the network can be classified into *static information* (being send only once per volume model) and *dynamic information*. Dynamic information must be re-sent whenever the ROI is redefined by the user. Static information is compact, including $W(V)$ and a set of arrays defining the tree structure of $G(V)$. In cases where the size of the volume model V is too large and the volume data at the deepest level of $G(V)$ does not fit into the client's CPU memory, the portion of this data belonging to the ROI is generated from the octree data on demand, as *dynamic information*, whenever the user asks for a different ROI.

In the client side, a low-resolution volume model V_W is reconstructed by de-quantizing and computing a few inverse wavelet steps in each block. Let us note as V_R the subvolume corresponding to the ROI. A two-level ray-casting rendering algorithm in the client GPU (Section 5) succeeds at showing a high-quality Gradient Octree rendering in V_R together with a visualization of V_W in the parts of the volume outside the ROI, also supporting a number of interaction facilities.

The corresponding compression and decompression algorithms are detailed in Section 4.

4 COMPRESSION AND DECOMPRESSION ALGORITHMS

We start by computing the wavelet transform $W(V)$ of the volume model V and its gradient octree $G(V)$ on the server, Figure 1. We use a localized, block-based transform with a previous smoothing step to achieve local behaviour and a better compression rate. We assume standard piecewise linear transfer functions [**?**]. By considering these transfer functions, we virtually segment the volume V in as many regions as linear segments defined by the $\{TF_k\}$ functions. Voxels with a density d such that the opacity of $\{TF_k\}$ is zero for all k, belong to null regions and are simply represented by a null code. Our implementation uses a block size of 16 together with a 4-steps Haar transform, being rather efficient in compression while supporting block-aware interaction paradigms in the client. As already mentioned, the wavelet information that is sent over the network to the client is a low resolution volume model V_W, obtained by computing a few w_l inverse wavelet steps in each block. Observe that in the usual case of $w_l = 2$, the size of the information in V_W will always be lower than 1/64 of the size of the initial model V. The low-resolution model for $w_l = 2$ is compressed more than a 98.5%. In what follows, we will use the term *compression rate* to name the relative size of the compressed model, which in this case is 1.5%.

The gradient octree $G(V)$ information includes the octree structure and the octree data. Creating $G(V)$ involves three compression steps. The first is transfer-function aware and uses V and the set of $\{TF_k\}$ to compute an Edge Volume model $VE(TF)$ which only encodes voxels that are relevant to the transfer functions $\{TF_k\}$. Non-relevant voxels in $VE(TF)$ are assigned a Nil value. On a second step, we compress gradient information to a total of three bytes per Grey tree node (including material information) in a set of data arrays, one per octree level, [2]. We use a GPU-oriented encoding of the proposed hierarchical data structure with explicit volume gradient information in octree nodes, to avoid gradient computations during GPU ray-casting. The final Gradient Octrees representation, shown in Figure 2, consists on a small volume model V_{32} with pointers and two sets of per-level arrays, O_l and D_l. For the sake of clarity, octree levels in Figure 2 and in what follows will be identified by their resolution. The example shown in Figure 2 corresponds to the complete octree representation of a volume V of resolution $r = 512$, with gradient and materials stored in the data array D_{512}. Data arrays of coarser octree levels (D_{256}, D_{128}, D_{64} and D_{32}) store gradient and materials data of Grey octree nodes at these levels. In short, the octree structure S of $G(V)$ consists of the pointers volume V_{32} and the set of per-level arrays O_l. The octree data D of $G(V)$ includes the set of per-level arrays D_l.

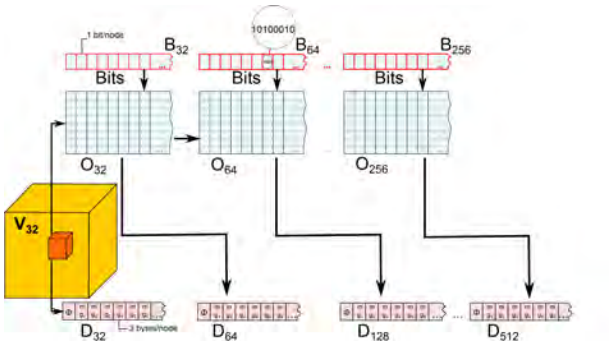


Figure 2: Encoding Gradient Octrees

For transmission purposes, it is possible to compact the Gradient Octree structure (not the volume data) in a lossless way. Instead of sending the arrays O_l, we send a set of arrays B_l with one byte per parent node as shown in red in Figure 2. Note that this is equivalent to store and transmit only one bit per node: the components in B_l simply represent the type of each child in one bit (0 if Nil, 1 if Grey). Compressed arrays B_l are computed on the server and sent to the client. For every received level, the client is able to generate a B_l-driven, increasing sequence of indexes to create a local copy of the array of indexes to child nodes O_l (for a detailed discussion, see [2]). Note that O_l indexes point simultaneously to D_{2l} and to O_{2l}. The volume V_{32} is sent to the GPU as a 3D texture, whereas arrays O_l and D_l are encoded as 2D and 1D textures. In short, we succeed in sending the tree structure S in a lossless way and with only one bit per node, through a sequence of compact arrays B_l. Moreover, compressing gradients and materials in three bytes is efficient, supports GPU decompression and suffers from a very limited loss in visual quality.

Although ROI-dependent localizations of the octree structure S could be defined [14], we have observed that the corresponding compression improvements (mainly in the information over the network) are negligible. In our present implementation we have therefore considered a hybrid model consisting of the low-resolution volume V_W, the gradient octree structure S and ROI-dependent octree data D. This hybrid model information is sent from the server to the client (or clients) in two parts: (I) The static information is sent only once, at the beginning of the interaction session. It consists on the low-resolution volume V_W, the $32 \times 32 \times 32$ pointers volume V_{32} of the computed Gradient Octree, the set of arrays B_l which encode the S octree structure and the materials look-up table of the Gradient Octree, Figure 1. The size of this last table is very small and we will not consider it in our compression computations. (II) The dynamic information is sent on demand whenever the client changes the ROI. The client sends a query with the new ROI limits (bounding box) and the server generates and sends a subset D_R of the data arrays D of the Gradient Octree, as we know in advance that only voxels in the ROI will be retrieved and rendered, Figure 1.

In our present implementation we assume that users are only interested in gradient octree data at the deepest octree level r, as lower resolutions are already shown outside the ROI. This makes the whole process easier, as we can just send a compact D_r array containing only those voxels with a non-Nil gradient value in the deepest octree level. This results in a very compact data transmission. We compute and keep a temporal, ROI-dependent version of the pointers volume V_{32} which we name VR_{32}. VR_{32} has Nil pointers outside the ROI and sequential pointers for the Grey nodes inside the ROI. The textures VR_{32} and O_k are now ROI-dependent, and must be recomputed in the client from the $G(V)$ Structure (see Figure1) whenever the ROI is changed during the interaction, with a very efficient algorithm which only involves array traversal and counting.

5 RENDERING AND INTERACTION IN THE CLIENT DEVICES

Reconstruction of any of the blocks within the non-ROI partion of the volume can be performed at one, two, three or four wavelet levels. The four-level reconstruction of a block generates a full piece of $16 \times 16 \times 16$ voxels that represent the corresponding part of the volume. Reconstructions of the same block at three, two or one levels generate pieces of $8 \times 8 \times 8$, $4 \times 4 \times 4$ or $2 \times 2 \times 2$ voxels, representing the same part of the volume at lower resolutions.

A usual interactive session starts by inspecting the whole volume model at a low resolution. In this case, all blocks are usually reconstructed at one or two levels, the corresponding 3D Texture is sent to the client GPU and ray-casting rendered. Observe that the size of this 3D texture, in the case of two reconstruction levels, is 1/64 of the size of the original volume model V.

Alternatively, the user can decide to inspect the whole volume model at a low resolution (two levels of reconstruction, for instance) with the ROI showing predefined structures at maximum level of detail by ray-casting the Gradient Octree. To achieve this last interactive visualization, two structures, one for the non-ROI volume (3D texture) and the other for the ROI volume (Gradient Octree), are sent to the client GPU where an adaptive ray-casting algorithm is performed, as detailed below. Since the whole model is available at the client side, rotation and zooming operations can be autonomously performed in the client without any further transmission from the server. If a region of the model needs to be detailed, the Gradient Octree can be displayed on demand.

We use a standard ray casting algorithm in the client GPU, the main difference with the classical algorithms

being that rays traverse a low resolution 3D texture representing the whole volume. In the point samples along the ray that do not belong to the ROI, the ray-casting uses density values from the low-resolution model V_W. Samples in the ROI retrieve densities from a virtual volume with the same resolution R as V_R, instead of traversing V_R itself. In Figure 3, an example with resolution $R = 512$ is shown. Ray-casting proceeds as usual by advancing along rays r from the observer with a uniform sampling of the volume along r. Then, for each sample s of r addressing a virtual voxel (i, j, k) in the ROI, its volume information is found in D_R.

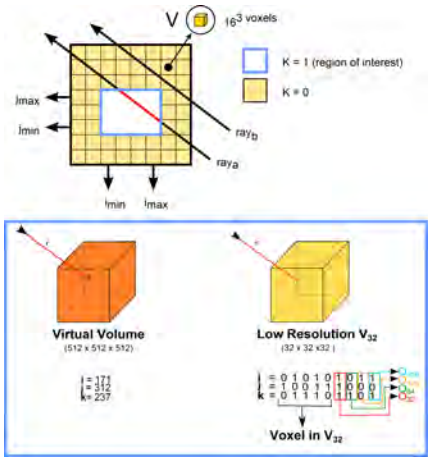


Figure 3: The block structure of the model, a region of interest (in white) and the octree-based ray-casting.

Ray casting within the ROI is based on the octree addressing properties. The octree search of any virtual voxel (i, j, k) is directly driven by the base-2 representation of i, j and k, as shown in Figure 3. In this case, their first 5 binary digits point to the corresponding voxel in the low-res texture V_{32}. The index i_{32} found in this V_{32} voxel element points to the low resolution data in D_{32} (which we don't use if a higher resolution is required) and also to the array of its eight child indexes in O_{32}. A well-known property of binary octree subdivision ensures that next "three bit columns" in the binary representation of i, j, k are in fact child indexes s_l. Son indexes point to deeper octree levels and are able to drive the octree traversal to the right element in D_R containing data in the virtual volume voxel. Subtree traversal from the low-res voxel in V_{32} to the virtual voxel data is based on the recursion equation,

$$i_{2l} = O_l[i_l][s_l] \tag{1}$$

for l=32, 64, 128, .. $R/2$

The final index i_R points to the high-res data in D_R, but tree traversal can stop earlier if the virtual voxel is void and any index i_l in the chain is found to be zero.

Observe that only virtual voxels in V_R in the ROI will be addressed. This means that the client must only store a restriction of $G(V)$ in V_R. In our present implementation we initially send the whole octree structure of $G(V)$ to the client, but high-resolution data D_R (restricted to V_R) is only sent on demand when the user changes the ROI

Let's assume that ray r is crossing voxel $i = 171$, $j = 312$, $k = 237$ in the virtual volume of the ROI, Figure 2. In this case, the octree search starts in the voxel $(10, 19, 14)$ of V_{32} and is then driven by four child indexes: $s_{32} = 7$, $s_{64} = 1$, $s_{128} = 4$ and $s_{256} = 5$ which recursively generate the indexes i_{64}, i_{128}, i_{256} and i_{512}. Reaching the deepest level information in a Gradient Octree of resolution $R = 512$ involves a maximum of six texture queries, to V_{32}, O_{32}, O_{64}, O_{128}, O_{256} and finally to D_{512}.

After retrieving high-res data in D_{512}, materials and the gradient vector are decompressed on the fly in the GPU. Obviously, everything also works when lower resolution virtual volumes are considered.

6 RESULTS AND DISCUSSION

To perform a complete comparative study , we selected the following accessible frameworks:

VrMed Viewer, an integration of libraries and functionalities, designed to achieve interactive visualization in PCs and Virtual Reality Systems, using a GPU-based Ray-casting algorithm [15].

Volume Viewer, an Android based application [5], implemented to run on mobile devices. Allows interactive visualization of models with a transfer function editor with easy handling. In this case, the whole volume model is sent to the client device.

VrMed-Thin Client The approach is based on [16]. It achieves remote visualization of volume models with basic user interaction tools in mobile devices. A server running VRMed Viewer on Linux operative system, renders images which are sent to the client through the wireless. Clients generate control commands as OpenGL parameters which are sent to the server using a TPC/IP socket.

Tables 1 and 2 show a comparison of [2], [1], our technique (**Hybrid Approach**), and the previously described schemes, using two models: The skull model with a $256 \times 256 \times 112$ resolution and the thorax model with a 512^3 resolution. Density values are in the rang [0...255], hence each voxel is codified using only 1 byte. Table rows show for each scheme whether multiresolution and progressive transmission is allowed and the compression rate achieved for each case. Both tables also show the size of the transmitted data trough the network and the client requirements to perform volume rendering followed by an estimation of the average frame rates in two cases: rendering in the PC server and rendering in the client (mobile device).

Figure 4 shows some snapshots of the interaction with the hybrid skull model. In all cases, two wavelet reconstruction steps ($wl = 2$) have been used without lighting computation in the low resolution area: In (a), (b) and (d) the Wavelet have been computed by using a skin tranfer function, whereas the image in (c) represents a bone transfer function both in the low resolution area and in the Region of Interest.

Some snapshots of the interaction with the hybrid thorax model are shown in Figure 5. In all cases two wavelet reconstruction steps ($wl = 2$) have also been applied without lighting computation, in the low resolution area: ROI showing ribs and lungs (a), internal gases and lungs (b) and skin, ribs and lungs (d). The snapshot in (e) shows the alveoli in the ROI, magnified in (f) by interacting with zoom and a section plane. In these cases wavelets are precomputed after applying to the model a TF covering all structures in the low resolution area. The snapshot in (c), shows a TF for bones visualization in both, the low resolution area, and the ROI. Image in (d) shows a zoom-in of (c) for showing up the quality of the hybrid model. The server application runs on PC with 6 GB of RAM, Intel Core 2 Duo at 3.16 GHz and a client with 4 GB of RAM, Intel Core 2 Duo at 3.06 GHz and Nvidia GeForce GTX z80. Client tests were performed on the HTC One smartphone whith a screen resolution of 1080 x 1920 pixels 2 GB RAM and an Adreno 320 Graphics processor.

A ROI-based visualization has been considered in the Wavelets-based scheme and in the Hybrid approach, while in the rest of columns, the whole volume V has been rendered at a uniform resolution. This is valid in both cases (tables 1 and 2). Zoom has been adjusted in a way that the total amount of rendered ROI pixels in the application viewport is a 25% of the total of viewport pixels. The amount of ROI pixels in the viewport is relevant, as it measures the total amount of required high-quality casted rays during ray-casting rendering.

Compression rates correspond to the amount of data sent over the network, and relate this amount to the total memory requirements of the volume models, which are 7.4 MBytes in Table 1 and 128 MBytes in the case shown in Table 2. In contrast to the previous schemes, our techniques allow multi-resolution rendering with progressive transmission of volume data. For the Wavelet based approach, the presented figures on the amount of data over the network represent the necessary information to reconstruct four levels of wavelet compression, $w_l = 4$. The compression rate in this case is between 21% and 32%.

In case of the Gradient Octree approach, data includes the octree structure plus material and gradient information at its deepest, maximum resolution level. In the Hybrid scheme, the *information over the network* represent both the necessary data to reconstruct two levels of wavelet compression for the low resolution model and the nodes representing the Gradient Octree leaves in the selected region of interest (ROI). This approach also requires a client GPU being able to manage 3D textures. The compression rate in this case is between 20% and 22%, with an average frame rate in the mobile device between 7 and 16 fps.

The proposal in this paper **Hybrid approach** results in a compression rate which is between 4% and 18%, with an average frame rate in the mobile device between 8 and 16 fps when $w_l = 2$. It also requires a client GPU being able to manage 3D textures.

The VrMed viewer is presented for comparison purposes. Some of the parameters in the tables do not apply to this case, as VrMed is a stand-alone application without network transmission. The average frame rates, 48 and 20 fps, are obviously higher than those in the previous cases but these figures show that our proposed approaches are performing within reasonable efficiency limits.

The Thin Client based approach sends a maximum of 0.18 MB of data through the network per frame during an interactive session with a single client (of course, the total amount of transmitted data depends on the number of interaction frames). This is due to the fact that the technique requires the transmission of rendered images from the server when the user interacts with the model in the client side. This fact makes this scheme network dependent, with framerates which decrease in network congestion cases. We have observed that our thin-client implementation becomes useless when the number of clients is above 8. On the other side, this scheme does not require sophisticated client GPUs, as clients must only decompress and show pre-rendered images. This can be an advantage for basic client devices, but result in an under-utilization of client GPUs in the case of most present devices. The asterisks in the *Thin client* column in tables 1 and 2 mean that data sizes are per frame sizes. The compression rates obviously depend on the number of transmitted frames.

The Hybrid approach is specially well suited in the case of large models. The comparison between tables 1 and 2 show that this is a scalable scheme, with compression rates that decrease when the size of the volume model increases. The corresponding frame rates are larger than in the case of Gradient Octrees, being of the same order of magnitude than Thin Clients.

Comparing Thin clients to Wavelets, Gradient Octrees and the Hybrid approach, we can define the break-even interaction period as the number of frames required to have an equivalent amount of information sent over the network. Break-evens are computed as the ratio between the size of the compressed model as sent over the network in our approaches and the size of a single Thin Client frame image. In the case of the skull model

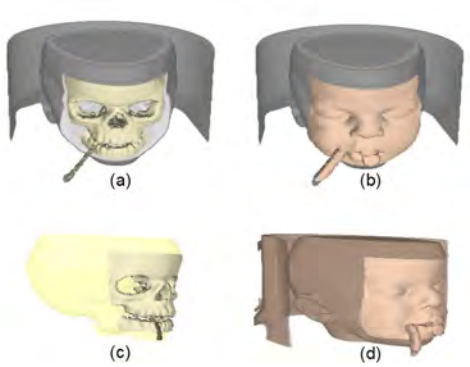


Figure 4: Hybrid Visualization. Interaction with the hybrid skull model. ROI size: $(128 \times 64 \times 64)$.

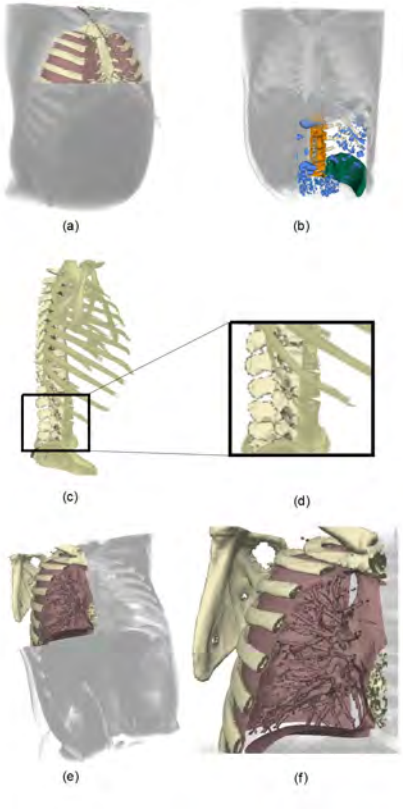


Figure 5: Hybrid Visualization. Interaction with the hybrid thorax model. ROI size: $416 \times 224 \times 224$ (a), $256 \times 160 \times 256$ (b), $96 \times 128 \times 480$ (c), and $256 \times 512 \times 256$ (d), (e) and (f).

in Table 1, this break-even is 11 frames for Wavelets, 21 frames for Gradient Octrees and 11 frames for the Hybrid approach.

In the case of the thorax model in Table 2, the break-even is 51 frames for Wavelets, 79 frames for Gradient Octrees and 17 frames for the Hybrid approach. By considering the number of frames per second in each case, we can conclude that the information we are sending is equivalent to the total information sent by the Thin Client approach during an interaction period in between 1 and 10 seconds. In the case of the Hybrid approach, break-evens are 11 frames and 17 frames, meaning this Hybrid scheme outperform Thin Clients in interaction cases longer than around 20 frames.

Thin Clients can also be compared with the presented approach in terms of frame rates. Frame rates depend on the network bandwidth, the present approach being better than Thin Client approaches in geographic regions with a limited bandwidth. In fact, the presented proposal can be specially useful in world regions with limited network infrastructures but requiring fast access to 3D medical data, like non-urban areas.

The Volume Viewer approach as presented in the last column of both tables does not require sophisticated client GPUs, as clients are rendering stacks of 2D textures. Frame rates in the client are reasonable. The main drawback in this case, however, is the amount of information being sent over the network, which makes it unusable in the case of large volume models.

The proposed hybrid scheme allows interactive inspection by rotating and zooming volume models. Users are able to select ROI portions of the visualized model, as well as choosing a transfer function from the set of transfer functions ($\{TF_k\}$) inside the the selected ROI. Interface options include also, planar section selection and offsets structures visualization in front of the selected section plane. Users can also choose the resolution of the low resolution region, by reconstructing models, by using one, two or three wavelet reconstruction steps.

	Wavelets Sheme	Gradient Octrees	Hybrid Approach	VrMed Viewer	VrMed Thin Client	Volume Viewer
Multi-resolution	✓	✓	✓	no	no	no
Progressive transmission	✓	✓	✓	no	no	no
Compression rate (%)	32	22	18	-	*	100
Data over the network(MB)	1.91	3.8	2.04	-	0.18*	7.3
Client requirements	3D Tex	3D Tex	3D Tex	-	Image	Stack 2D Tex
Frame rate (PC version)	52	24.12	46.53	48.24	48.24	-
Frame rate (mobile)	24.2	-	16.32	-	20.23	17.0

Table 1: Comparison between the approach presented in this paper (blue column) and some previous schemes for remote visualization of volume models. Study of the Skull model with a resolution of $256 \times 256 \times 112$ and 7.3 MB of size.

	Wavelets Sheme	Gradient Octrees	Hybrid Approach	VrMed Viewer	VrMed Thin Client	Volume Viewer
Multi-resolution	✓	✓	✓	no	no	no
Progressive transmission	✓	✓	✓	no	no	no
Compression rate (%)	21	20	4.2	-	*	100
Data over the network(MB)	16.4	27	5.3	-	0.32*	128
Client requirements	3D Text	3D Tex	3D Tex	-	Image 2D Tex	Stack
Frame rate (PC version)	14	10.31	18.03	20.34	20.34	-
Frame rate (mobile)	13.20	-	8.07	-	20.23	-

Table 2: Comparison between the approach presented in this paper (blue column) and some previous schemes for remote visualization of volume models. Study of the Thorax model with a resolution of 512^3 and 128 MB of size.

7 CONCLUSIONS & FUTURE WORK

We have proposed a Hybrid approach that inherits the advantages of the algorithms presented in [1] and [2] while keeping a good performance in terms of bandwidth requirements and storage needs in client devices. Information over the network consists on static information (being only set once) and dynamic information. Dynamic information must be re-sent whenever the ROI is redefined by the user. The complexity (memory and data transmission requirements) of the static and dynamic information has been discussed. The main conclusion is that the hybrid scheme is flexible enough to represent several materials and volume structures in the ROI area at a very limited static and dynamic information transmission cost.

The Hybrid approach has been proved to be specially well suited in the case of large models. The presented experimental tables show that the Hybrid approach is a scalable scheme, with compression rates that decrease when the size of the volume model increases. Corresponding frame rates are larger than in the case of Gradient Octrees, being of the same order of magnitude than Thin Clients. Our compression results are better than similar client server schemes for volume rendering, and compare favourably to Marching Cubes based approaches. While these last schemes must send an average of three triangles per voxel in segmented volumes, the presented approach only sends three bytes/voxel. We consider that our approach may enrich the user experience during the inspection of volume medical models in these low performance devices.

ACKNOWLEDGMENT

This work has been partially supported by the Project TIN2013-47137-C2-1-P of the Spanish Ministerio de Economía y Competitividad.

8 REFERENCES

[1] Lázaro Campoalegre, Pere Brunet, and Isabel Navazo. Interactive visualization of medical volume models in mobile devices. *Personal Ubiquitous Comput.*, 17(7):1503–1514, 2013.

[2] Lázaro Campoalegre, Pere Brunet, and Isabel Navazo. Gradient octrees: A new scheme for remote interactive exploration of volume models. *Proc. of the CAD/Graphics*, 2013.

[3] Manuel Moser and Daniel Weiskopf. Interactive volume rendering on mobile devices. In *Vision, Modeling, and Visualization VMV*, volume 8, pages 217–226, 2008.

[4] Jose Maria Noguera and Juan Roberto Jiménez. Visualization of very large 3d volumes on mobile devices and webgl. In *WSCG international conference on computer graphics, visualization and computer vision*, 2012.

[5] Marcos Balsa and Pere Vázquez. Practical volume rendering in mobile devices. In *Advances in Visual Computing*, pages 708–718. Springer, 2012.

[6] Klaus Engel and Thomas Ertl. Texture-based volume visualization for multiple users on the world wide web. In *Virtual Environments*, pages 115–124. Springer, 1999.

[7] Eric LaMar, Bernd Hamann, and Kenneth I. Joy. Multiresolution techniques for interactive texture-based volume visualization. In *IEEE Visualization*, pages 355–361, 1999.

[8] Enrico Gobbetti, José Antonio Iglesias Guitián, and Fabio Marton. Covra: A compression-domain output-sensitive volume rendering architecture based on a sparse representation of voxel blocks. In *Computer Graphics Forum*, volume 31, pages 1315–1324, 2012.

[9] Shigeru Muraki. Volume data and wavelet transforms. *Computer Graphics and Applications, IEEE*, 13(4):50–56, 1993.

[10] Insung Ihm and Sanghun Park. Wavelet-based 3D compression scheme for interactive visualization of very large volume data. *Computer Graphics Forum*, pages 3–15, 1999.

[11] Stefan Guthe, Michael Wand, Julius Gonser, and Wolfgang Straßer. Interactive rendering of large volume data sets. In *Visualization, 2002. VIS 2002. IEEE*, pages 53–60. IEEE, 2002.

[12] Jianlong Zhou, Manfred Hinz, and Klaus D Tönnies. Hybrid focal region-based volume rendering of medical data. In *Bildverarbeitung für die Medizin 2002*, pages 113–116. Springer, 2002.

[13] Yanlin Luo, José Antonio Iglesias Guitián, Enrico Gobbetti, and Fabio Marton. Context preserving focal probes for exploration of volumetric medical datasets. In *Proceedings of the International Conference on Modelling the Physiological Human*, pages 187–198, 2009.

[14] Lazaro Campoalegre. Contributions to the interactive visualization of medical volume models in mobile devices. In *PhD Thesis, UPC*, 2014.

[15] Eva Monclús, José Díaz, Isabel Navazo, and Pere-Pau Vázquez. The virtual magic lantern: An interaction metaphor for enhanced medical data inspection. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 119–122, 2009.

[16] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl. Interactive volume on standard pc graphics hardware using multi-textures and multi-stage rasterization. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 109–118, 2000.

Joint Bilateral Mesh Denoising using Color Information and Local Anti-Shrinking

Oliver Wasenmüller
oliver.wasenmueller@dfki.de

Gabriele Bleser
gabriele.bleser@dfki.de

Didier Stricker
didier.stricker@dfki.de

DFKI GmbH - German Research Center for Artificial Intelligence
Augmented Vison Department
Trippstadter Str. 122
67663 Kaiserslautern, Germany

ABSTRACT

Recent 3D reconstruction algorithms are able to generate colored meshes with high resolution details of given objects. However, due to several reasons the reconstructions still contain some noise. In this paper we propose the new Joint Bilateral Mesh Denoising (JBMD), which is an anisotropic filter for highly precise and smooth mesh denoising. Compared to state of the art algorithms it uses color information as an additional constraint for denoising; following the observation that geometry and color changes often coincide. We face the well-known mesh shrinking problem by a new local anti-shrinking, leading to precise edge preservation. In addition we use an increasing smoothing sensitivity for higher numbers of iterations. We show in our evaluation with three different categories of testdata that our contributions lead to high precision results, which outperform competing algorithms. Furthermore, our JBMD algorithm converges on a minimal error level for higher numbers of iterations.

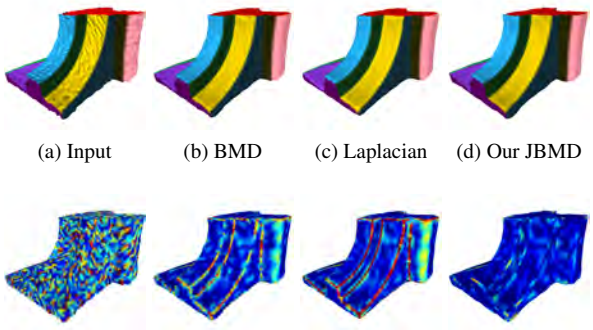
Keywords

Mesh Denoising, Smoothing, Fairing, Joint Bilateral Filter, Local Anti-Shrinking, Color Information.

1 INTRODUCTION

Many applications, such as urban planning, industrial measurement or human anthropometry, require reconstructed 3D models of the respective objects with very high precision. The traditional approach is to acquire these models by laser scanners, since they promise high quality results. However, they are expensive, impractical to use and contain still some noise. Meanwhile 3D reconstruction algorithms, such as e.g. [Agi, FP10, NIH+11, SSC14], are able to generate colored mesh models from devices like standard color cameras and\or depth cameras, which are widely spread, cheap and easy to use. These reconstructions contain color information together with high-resolution details, but also suffer from noise in their 3D geometry. To get rid of this noise, several (mostly iterative) methods for mesh denoising were proposed in the literature. Sometimes they are also referred to as smoothing, filtering or fairing methods. They use directly the 3D geometry or derived measures, like distances or normals of the mesh, to estimate new vertex positions. However, none of them explicitly uses the color information provided by e.g. one of the above mentioned algorithms. Thus, we present in this paper a new anisotropic method called Joint Bilateral Mesh Denoising (JBMD), which uses - besides geometric information - the color information as an additional constraint for edge preserving denoising.

Another well-known problem of mesh denoising are shrinking effects. They occur mostly in curves regions of the mesh and are caused by homogeneous shifts of vertices in a neighborhood into one major direction. Current approaches try to compensate that ef-



(a) Input (b) BMD (c) Laplacian (d) Our JBMD

Figure 1: Comparison of different mesh denoising algorithms for the *fandisk* mesh. Top row: meshes. Bottom row: color-coded error distribution.

fect, whereas they focus more on visually appealing results than on precision. Thus, in our JBMD algorithm we propose a new approach to avoid this effect by a precise local anti-shrinking. Furthermore, many current algorithms suffer from their high dependence on the number of iterations. Therefore our new algorithm increases the denoising sensitivity per iteration leading to constantly low errors. A short overview of our algorithm is also given in the following video: `https://youtu.be/odm8kr2rKPA`

Summarizing, the main contributions of our proposed JBMD algorithm are:

- explicit usage of color information as an addition constraint for denoising,
- precise local anti-shrinking and
- increasing denoising sensitivity.

The remainder of this paper is organized as follows: Section 2 gives an overview of existing methods for mesh denoising. The proposed JBMD algorithm is motivated and explained in detail in Section 3, while it is evaluated regarding precision in Section 4. The work is concluded in Section 5.

2 RELATED WORK

Image and mesh denoising is an ongoing research topic in the image processing and computer vision community.

Often, mesh denoising methods are related to image denoising approaches. State of the art approaches in image denoising include methods such as anisotropic diffusion [PM90], total variation [ROF92], wavelet denoising [Don95], robust diffusion [BSMH98], bilateral filter [TM98] and joint bilateral filter [KCLU07, HSJS08]. In particular the joint bilateral filter uses, similar to our new approach, the color information as an additional constraint.

State of the art algorithms for mesh denoising are amongst others Laplacian [Fie88, Tau95, VMM99] and bilateral [FDCO03, JDD03, ZFAT11] mesh denoising (BMD) methods. Laplacian mesh denoising is an iterative isotropic procedure, where the new vertex positions are directly calculated from the positions of the neighboring vertices. In contrast, bilateral mesh denoising is an iterative edge preserving anisotropic approach. New vertex positions are estimated from the vertex's neighborhood, where the influence of neighboring vertices depends on their distance and on their offset to the tangent plane. Parts of this approach are also used for our new algorithm.

A general and well-known problem of mesh denoising is that the mesh shrinks in convex regions with each application of the particular algorithm, which is a huge problem especially for iterative approaches. [Tau95] solves this problem by alternating shrinking and expansion steps. Admittedly the precision of this approach depends heavily on the geometry of the particular mesh [DMSB99]. Another common approach, which is e.g. used in the Bilateral Mesh Denoising [FDCO03], is to preserve the volume of the mesh by a global correction step as proposed in [DMSB99]. The algorithm estimates the volume V^n of a mesh after the n-th iteration by the sum of volumes of all ordered pyramids centered at the origin and with a triangle of the mesh as base. Each vertex of the mesh is then scaled by the factor β, which is defined by

$$\beta = \left(\frac{V^0}{V^n}\right)^{\frac{1}{3}}, \tag{1}$$

to achieve the original volume V^0. However, as mesh shrinkage occurs only in convex regions contrary to flat regions, a global correction has indeed appealing effects but is not precise. Thus, we propose in this paper a precise local shrinkage correction.

3 METHOD

In this paper we propose the Joint Bilateral Mesh Denoising (JBMD), which is a filtering method for meshes using local neighborhoods. The method can be subdivided into two parts: the denoising itself and the subsequent local anti-shrinking.

The main idea of our new denoising algorithm is related to image processing, namely motivated trough the Joint Bilateral Filter (JBF) [KCLU07]. This anisotropic edge-preserving filter is often used to denoise depth images by using color images as additional constraints. The main idea is to compute a new depth value as a weighted average of surrounding depth values, where the weights depend on their deviation in position (space) and color value (range). The assumption of JBF are coherent depth and color discontinuity, meaning that edges in the color image coincide with edges in the depth image and vice versa. This coherence assumption was validated in many image processing publications [KCLU07, WBS15] and we show in Section 4 that it also holds for meshes.

The intention of our new local shrinkage correction is - contrary to alternating [Tau95] or global [DMSB99] correction - to adjust vertex positions only where shrinkage effects occurred. This effect arises only in convex regions, whereas flat regions are not affected. We observed that the weighted mean signed shift of vertices in the neighborhood, which were estimated by the denoising in the first step, equalize in noisy flat regions, whereas in convex (and thus shrunk) regions it is a precise local measure for a shrinkage correction.

Like in many other mesh denoising algorithms [FDCO03, ZFAT11] we estimate in our JBMD new

vertex positions v'' in a mesh by shifting along the normal direction n. This has the positive effect that irregularities in the resulting mesh are avoided. Our two-step algorithm can be described by

$$v'' = v + (x' \cdot n) - (x'' \cdot n), \tag{2}$$

where $(x' \cdot n)$ is the denoising part, $(x'' \cdot n)$ the correction part and $x'|x''$ refer to the magnitude of the shift. The algorithm is illustrated in detail in Figure 2 and defined in the following:

Algorithm 1 Joint Bilateral Mesh Denoising (JBMD) for the m-th iteration

Require: Vertex v, Normal n

1: $\{q_i\}$ = neighborhood(v)
2: $sum, norm, sum', norm' = 0$
3: **for all** i **do**
4: $d_i = \|v - q_i\|$
5: $o_i = \langle n, v - q_i \rangle$
6: $c_i = (v.r - q_i.r)^2 + (v.g - q_i.g)^2 + (v.b - q_i.b)^2$
7: $w_i^d = exp(-d_i^2/2\sigma_d^2)$
8: $w_i^o = exp(-o_i^2/2(\sigma_o \cdot \lambda^m)^2)$
9: $w_i^c = exp(-c_i/2\sigma_c^2)$
10: $sum += (w_i^d \cdot w_i^o \cdot w_i^c) \cdot o_i$
11: $norm += w_i^d \cdot w_i^o \cdot w_i^c$
12: **end for**
13: $x' = sum/norm$
14: $v' = v + x' \cdot n$
15: **for all** i **do**
16: $d_i' = \|v' - q_i'\|$
17: $w_i^d = exp(-(d_i')^2/2\sigma_d^2)$
18: $sum' += w_i^d \cdot x_i'$
19: $norm' += w_i^d$
20: **end for**
21: $x'' = sum'/norm'$
22: $v'' = v' - x'' \cdot n$
23: **return** v"

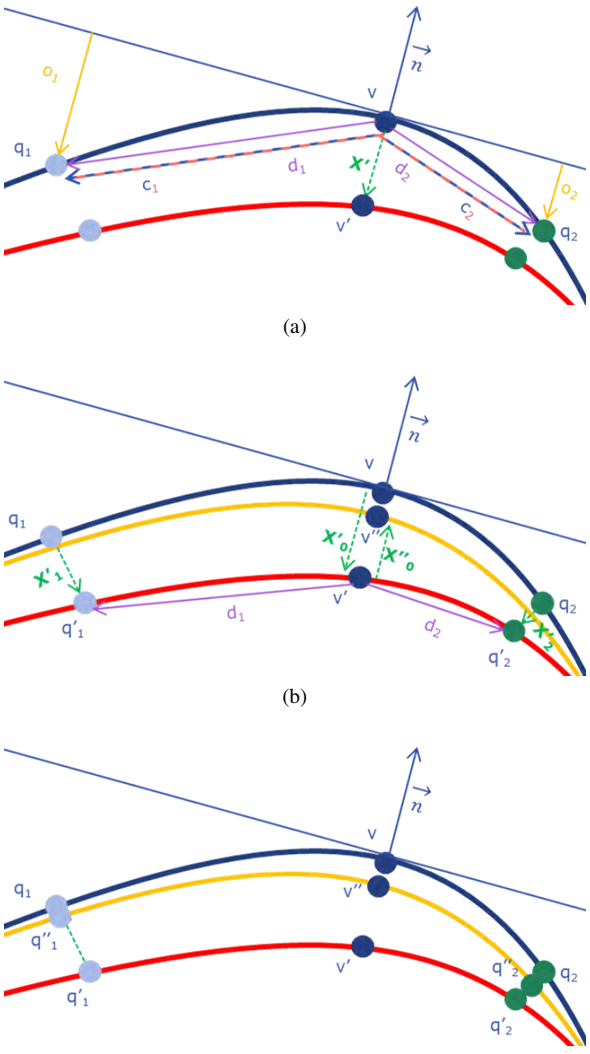


Figure 2: Joint Bilateral Mesh Denoising (JBMD) applied to vertex v: (a) Denoising step ($v \rightarrow v'$). (b) Shrinking correction step ($v' \rightarrow v''$). (c) Final result.

In the first step of our algorithm (line 3-14) the new position v' of a vertex v is estimated as a weighted average of neighboring vertex position q_i, where the weights depend on three influencing factors: distance (w_i^d), offset (w_i^o) and color difference (w_i^c). For computing the distance d_i between a vertex v and neighboring vertex q_i, the geodesic distance on the smooth surface would be the correct measure. However, for efficiency reasons we approximate d_i using the Euclidean distance in line 4, since [FDCO03] demonstrated already a sufficient impact. The offset o_i is defined as the distance of vertex q_i to the tangent plane of vertex v. The intention of using this offset o_i is that neighboring points in flat regions should have a higher influence than in convex or edge regions. As described in line 5, o_i can be easily estimated using the dot product. The last influence factor is the color difference c_i between a vertex q_i and vertex v, which is estimated in line 6. To map the influence factors d_i, o_i and c_i to weights w_i^d, w_i^o and w_i^c, we use the Gaussians of lines 7-9. The final shift x' of vertex v is the normalized weighted sum of offsets o_i of neighboring vertices q_i.

In the second step of our algorithm (line 15-22) we correct the position of a vertex v' due to possible shrinking effects. As already mentioned before, we observed that the weighted mean signed shift x'' of vertices in the neighborhood, which were estimated by the denoising in the first step, is a good local measure for a shrinkage correction. For the estimation of the weights for x'' we use the distance d_i' between a vertex q_i' and v' together with a mapping function (line 16-17) similar to the first step of the algorithm. The weighted mean signed shift x'' is calculated by summing up the weighted signed sifts x_i' (line 18) and normalizing afterwards (line 21).

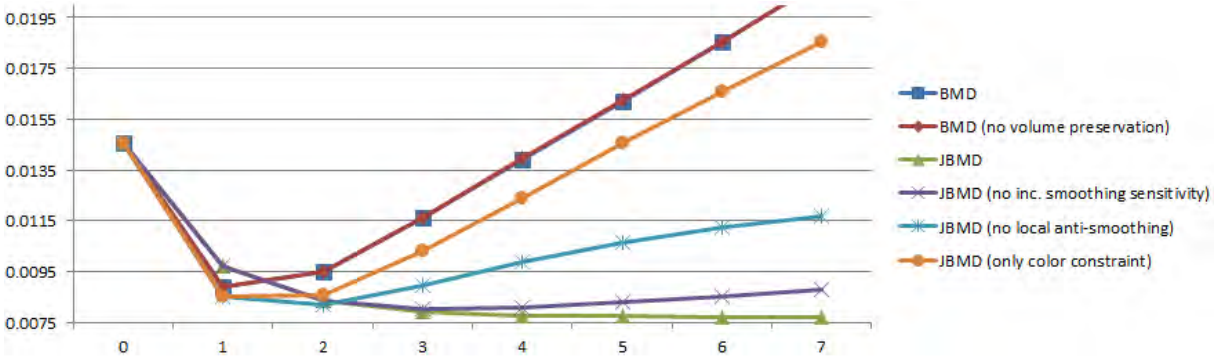


Figure 3: Mean errors of the *fandisk* mesh for different numbers of iterations. Quantitative comparison of different features of our new JBMD algorithm and comparison against BMD.
Dark blue: BMD [FDCO03]. Red: BMD without volume preservation. Green: Our new JBMD. Purple: JBMD without increasing smoothing sensitivity. Light blue: JBMD without local anti-shrinking. Orange: JBMD without increasing smoothing sensitivity and without local anti-shrinking.

The described JBMD algorithm is applied locally for each vertex of the mesh. However, vertices at a boundary of a mesh do not have a well defined neighborhood (line 1). In our algorithm we define the size of the neighborhood as a fixed number k. The neighborhood of a vertex v is then defined by the k closest vertices q_i. Obviously, the shape of our neighborhood changes from vertex to vertex, but since the distance d_i between vertices v and q_i is an influencing parameter, this artifact has negligible influence.

Our JBMD algorithm is - like many other [FDCO03, ZFAT11] - an iterative approach. In the first iteration major noise is eliminated, whereas with higher number of iterations the overall level of noise decreases. Thus, we consider this aspect by an increasing smoothing sensitivity. In our JBMD algorithm the noise influences the result via the offset o_i, whereas the corresponding mapping function depends on σ_o. Therefore, we decrease the parameter σ_o by a constant factor λ with each iteration; leading to constantly low error.

4 EVALUATION

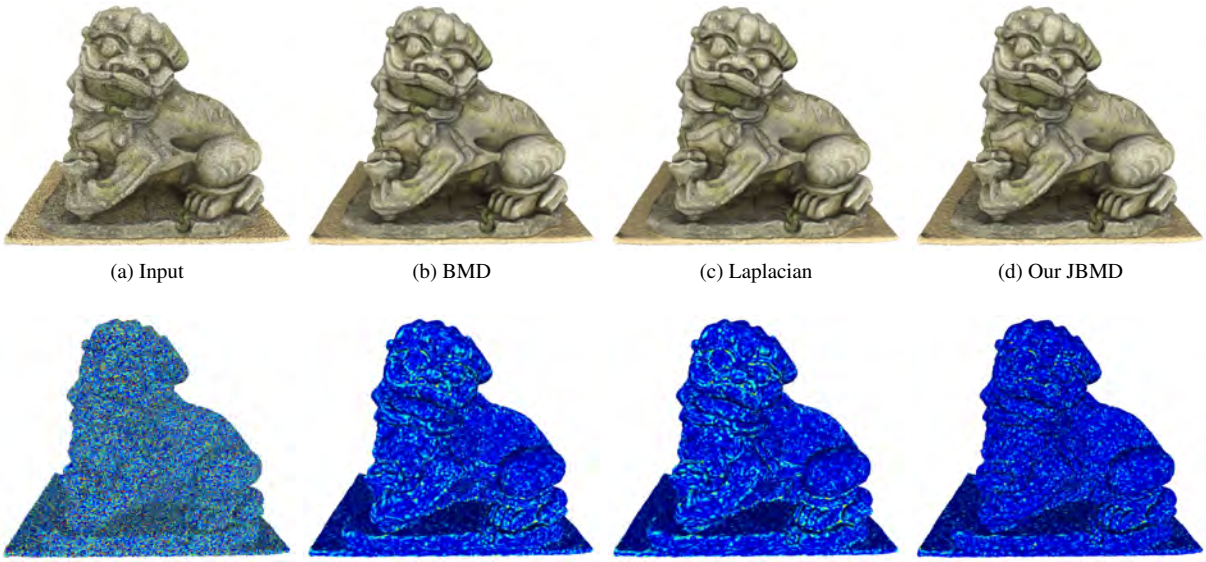
In this section we benchmark our JBMD algorithm by comparing it to competing algorithms, namely Laplacian Denoising and Bilateral Mesh Denoising (BMD). These algorithms are described in more detail in Section 2. All methods - including ours - depend on some parameters. For our JBMD algorithm these are σ_d, σ_o σ_c and λ. Thus, we run each algorithm with a huge number of possible parameter combinations to detect the optimal setting. All results (Figures, diagrams, etc.) shown in this paper are generated with optimal parameter settings and numbers of iterations. For our JBMD algorithm we used the parameter settings of Table 1 for the given datasets. Note, these parameters depend highly on the mean vertex distance (MVD) of the given mesh. According to our experiments $MVD \approx 2\sigma_d \approx 4\sigma_o$ can be used as a rough guideline for setting the parameters.

Unfortunately, a groundtruth comparison on real world data is very difficult, since no datasets are available, which provide both real noisy data and real denoised data. Thus, in the recent literature it is common to use precise models of an object and generate the noise on it synthetically. For this paper we decided to use three categories of testdata in our evaluation.

The first category are colored synthetic meshes with sharp edges together with an artificially noisy version of this mesh. We use here the well-known *fandisk* mesh, where each part of the surface has another color. Furthermore, we add a Gaussian noise, where the standard deviation is roughly half of the vertex distance. The second category of testdata are highly precise reconstructions of real objects acquired by a camera-projector-system [KNRS13]. We also added here a Gaussian noise with a standard deviation of approximately half vertex distance. The meshes used in this paper are the *lion* and *allegorie* reconstructions. The third category of testdata are reconstructed meshes, which are generated by standard cameras and Agisoft PhotoScan [Agi]. We use in this paper the heads of two persons: *person 1* and *person 2*. These reconstructions include partially strong noise due to the lack of characteristic features. Obviously, for these reconstructions no ground truth is available, but they are a real world scenario, where the application of a mesh denoising algorithm is required.

	MVD	σ_d	σ_o	σ_c	λ
fandisk	0.1897	0.1	0.05	30	0.8
lion	0.4401	1.0	0.5	30	0.6
allegorie	0.0003	0.0008	0.0004	35	0.7
person 1	0.0030	0.01	0.005	20	0.7
person 2	0.0041	0.01	0.005	20	0.7

Table 1: Parameter settings in the evaluation of our new JBMD algorithm for the given datasets with specified mean vertex distance (MVD).



(a) Input (b) BMD (c) Laplacian (d) Our JBMD

Figure 4: Comparison of different mesh denoising algorithms for the *lion* mesh. Top row: meshes. Bottom row: color-coded error distribution.

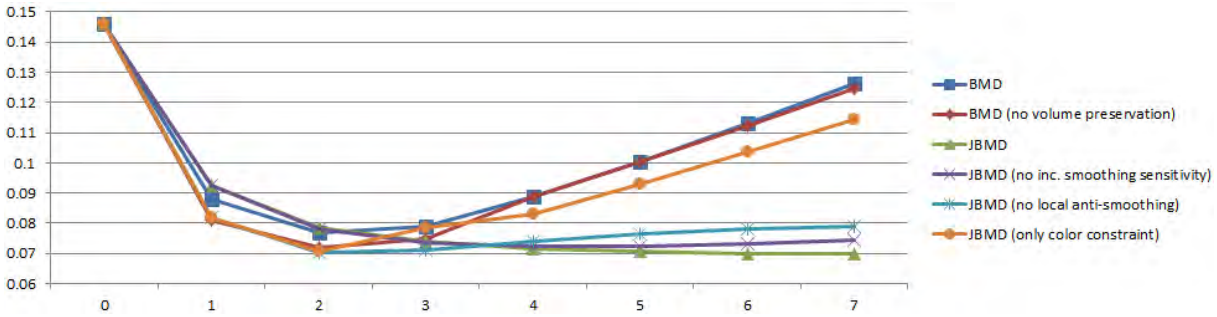


Figure 5: Mean errors of the *lion* mesh for different numbers of iterations. Quantitative comparison of different features of our new JBMD algorithm and comparison against BMD.

As a quality measure for denoising we use the mean error (ME), which is defined for a denoised mesh X and a ground truth mesh G by

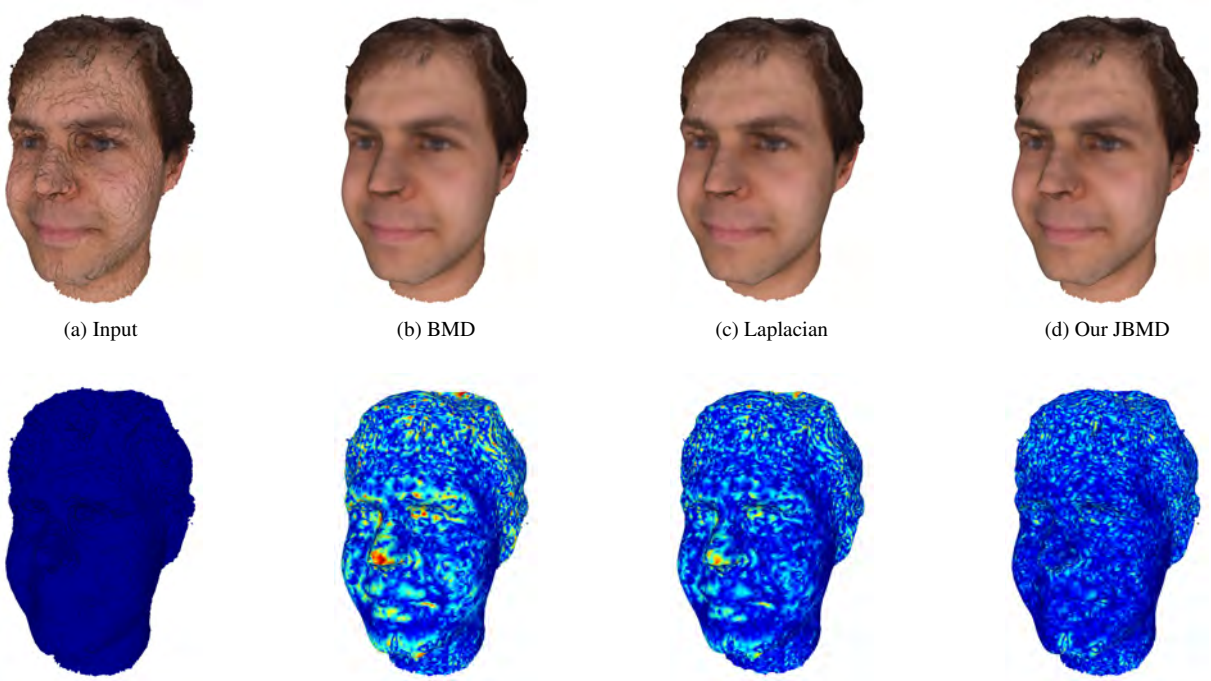
$$ME(X,G) = \frac{1}{n}\sum_{i=1}^{n} \|x_i - g_p\| \qquad (3)$$

$$x_i \in X; g_p \in G; \forall i\ p = \arg\min_p \|x_i - g_p\|,$$

where n is the number of vertices in the mesh X. More intuitive, it is defined by the mean distance of each vertex in the mesh X to the respective closest vertex in the ground truth mesh G.

First of all we evaluate our algorithm with the *fandisk* mesh in Figure 1, which has ideal preconditions for our JBMD, since all sharp edges coincide with color changes. From a visual point of view, all three mesh denoising algorithms provide smooth results without visible noise. However, they differ strongly in their precision, as visible in the color-coded error distribution in the bottom row. The blue color indicates low errors, whereas red represents high errors. Both BMD and Laplacian denoising have imprecise vertices at the edges of the mesh, whereas our JBMD has only some minor inaccuracy. Figure 3 depicts the mean error of the *fandisk* mesh depending on the number of iterations. Our JBMD has the lowest error and converges in particular on this low error level. If our JBMD is used without the increasing smoothing sensitivity, the mean error increases again from the fourth iteration on. If we switch off our shrinking correction, we achieve better results for a small number of iterations. This is caused by the inhibiting effect of the shrinking correction, since it reverts the denoising to some extent. However, for larger numbers of iterations superior results can be achieved with our new local anti-shrinking. Looking at the effects of using the color information as an additional parameter, we see that our JBMD has - even without increasing smoothing sensitivity and without local anti-shrinking (orange line) - always a lower mean error than the BMD.

The *lion* and *allegorie* meshes, which correspond to the testdata category of real reconstructions with synthetic noise, are depicted in Figure 4 and 7 respectively. Again, from a visual point of view all three mesh de-



(a) Input (b) BMD (c) Laplacian (d) Our JBMD

Figure 6: Comparison of different mesh denoising algorithms for the *person 1* mesh. Top row: meshes. Bottom row: color-coded comparison against the input mesh.

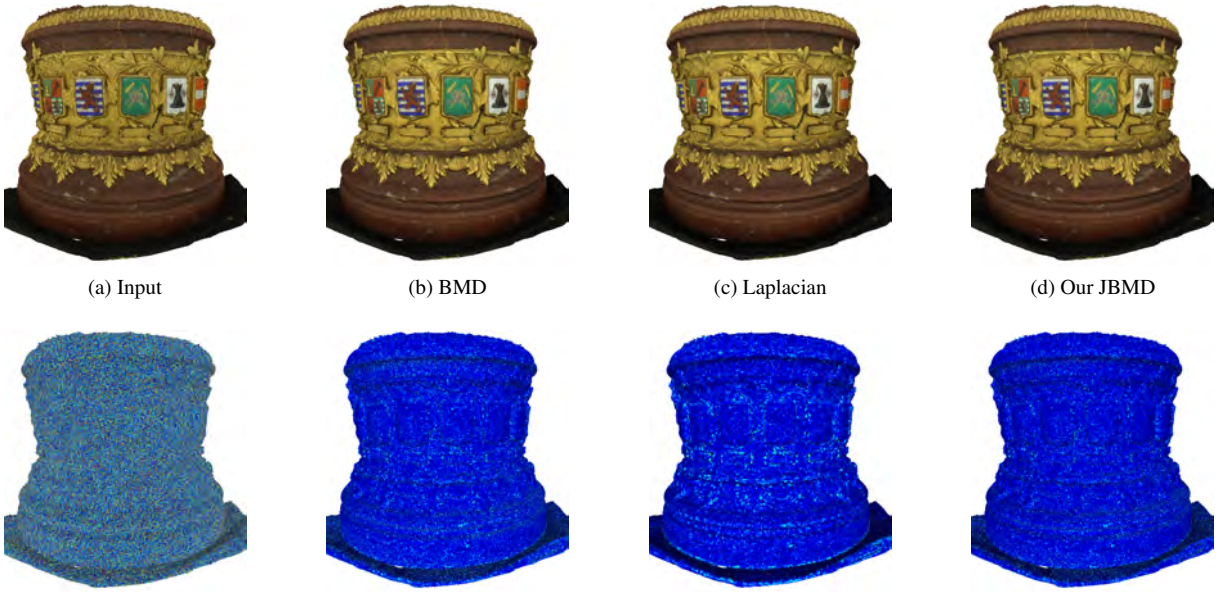
noising algorithms deliver smooth results. However, for both datasets at the edges of the mesh BMD and Laplacian denoising are less precise than our JBMD. Figure 5 and Figure 8 depict the respective mean error of the *lion* and *allegorie* meshes depending on the number of iterations. Again, our JBMD outperforms the competing algorithms and converges at the lowest error level.

The *person 1* and *person 2* meshes correspond to the testdata category of reconstructions with real world noise. Since no groundtruth data is available for these datasets, we compare the denoised meshes against the original mesh in Figure 6 and 9 respectively. The BMD algorithms results in the biggest differences to the original mesh. Especially the nose, but also the eyebrows and mouth, have a huge deviation and are not precise. The Laplacian smoothing shows less deviation, but is by far not as close to the original mesh as our JBMD. Of course, smaller deviations to the original do not mandatory result in a better quality, but from a visual point of view all results are similarly smooth. Thus, also for this category of testdata our JBMD outperforms the competing algorithms.

Summarizing the evaluation results, we found out that our JBMD algorithm outperforms competing algorithms for all tested datasets in terms of precision while creating smooth results. Notably is in particular that our JBMD converges on the lowest error level for higher numbers of iterations. This is the achievement of all three main contributions of our paper: Using color information as additional constraint, correcting shrinking effects locally and increasing the smoothing sensitivity with each iteration. Like illustrated in Figure 3, 5 and 8 this is only possible with the combination of all these three contributions. As long as at least one of them is not activated, the mean error is not minimal and does not converge. With the local anti-shrinking it is possible to denoise especially edges very precisely. Furthermore, we verified with our convincing result that the coherence assumption of coinciding geometry and color changes holds also for meshes.

5 CONCLUSION

In this paper we proposed the new Joint Bilateral Mesh Denoising (JBMD), which is an anisotropic filter for highly precise and smooth mesh denoising. Under the assumption of coinciding geometry and color changes it uses color information as an additional constraint for denoising. This assumption is adapted from the Joint Bilateral Filter (JBF) of the recent image processing research and we showed in this paper that this coherence assumption also holds for meshes. Furthermore, we proposed a precise local anti-shrinking, which leads to precision improvements especially at the edges of the mesh. Our third contribution increases the smoothing sensitivity for higher numbers of iterations. In our evaluation we compared our new JBMD algorithm against competing algorithm based on three categories of test data. We showed that our contributions lead to high precision results with lowest errors. In addition our algorithm converges to the minimum error level for higher numbers of iterations.



(a) Input (b) BMD (c) Laplacian (d) Our JBMD

Figure 7: Comparison of different mesh denoising algorithms for the *allegorie* mesh. Top row: meshes. Bottom row: color-coded error distribution.

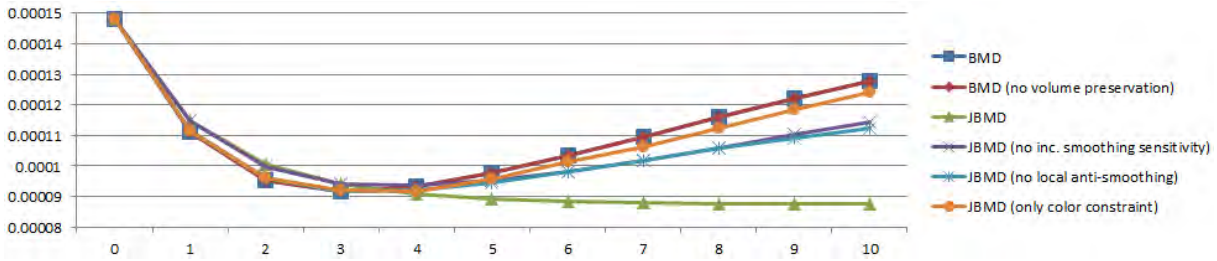
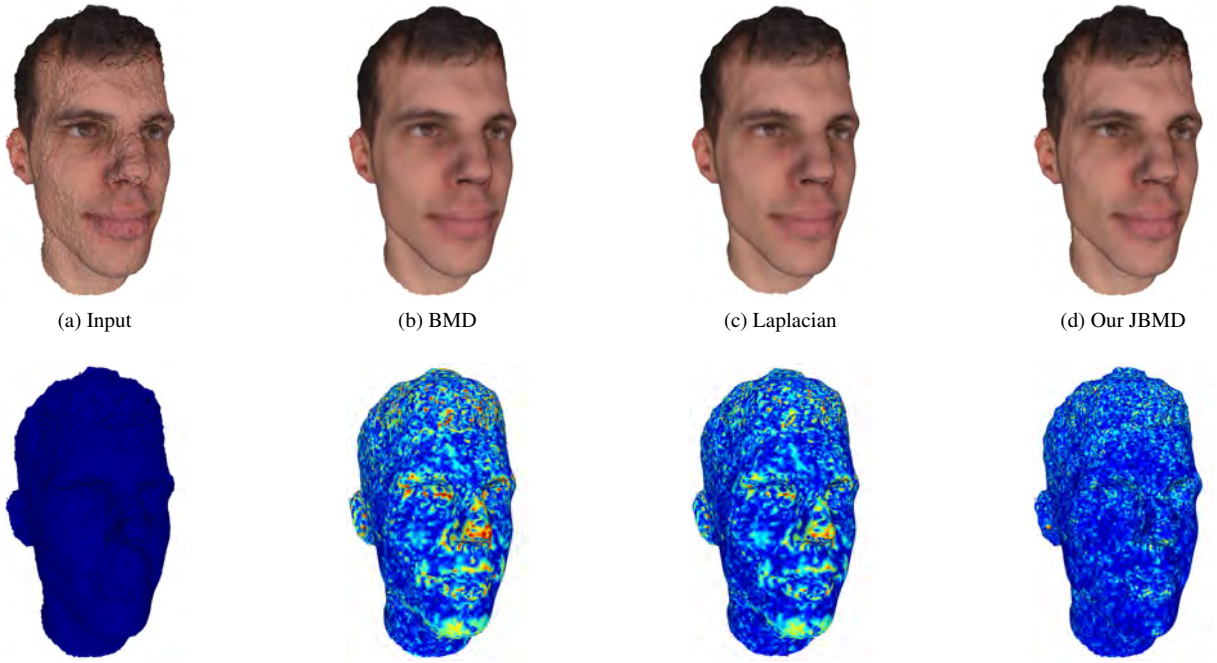


Figure 8: Mean errors of the *allegorie* mesh for different numbers of iterations. Quantitative comparison of different features of our new JBMD algorithm and comparison against BMD.



(a) Input (b) BMD (c) Laplacian (d) Our JBMD

Figure 9: Comparison of different mesh denoising algorithms for the *person 2* mesh. Top row: meshes. Bottom row: color-coded comparison against the input mesh.

ACKNOWLEDGEMENTS

This work was partially funded by the Federal Ministry of Education and Research (Germany) in the context of the project DENSITY (01IW12001).

6 REFERENCES

[Agi] Agisoft. PhotoScan. http://www.agisoft.com/.

[BSMH98] Michael J Black, Guillermo Sapiro, David H Marimont, and David Heeger. Robust anisotropic diffusion. *Image Processing, IEEE Transactions on*, 7(3):421–432, 1998.

[DMSB99] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324. ACM Press/Addison-Wesley Publishing Co., 1999.

[Don95] David L Donoho. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3):613–627, 1995.

[FDCO03] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 950–953. ACM, 2003.

[Fie88] David A Field. Laplacian smoothing and delaunay triangulations. *Communications in applied numerical methods*, 4(6):709–712, 1988.

[FP10] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1362–1376, 2010.

[HSJS08] Benjamin Huhle, Timo Schairer, Philipp Jenke, and Wolfgang Straßer. Robust non-local denoising of colored depth data. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–7. IEEE, 2008.

[JDD03] Thouis R Jones, Frédo Durand, and Mathieu Desbrun. Non-iterative, feature-preserving mesh smoothing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 943–949. ACM, 2003.

[KCLU07] Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. In *ACM Transactions on Graphics (TOG)*, volume 26, page 96. ACM, 2007.

[KNRS13] Johannes Kohler, Tobias Noll, Gerd Reis, and Didier Stricker. A full-spherical device for simultaneous geometry and reflectance acquisition. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 355–362. IEEE, 2013.

[NIH+11] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[PM90] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):629–639, 1990.

[ROF92] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[SSC14] Frank Steinbrucker, Jurgen Sturm, and Daniel Cremers. Volumetric 3d mapping in real-time on a cpu. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2021–2028. IEEE, 2014.

[Tau95] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM, 1995.

[TM98] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[VMM99] Jörg Vollmer, Robert Mencl, and Heinrich Mueller. Improved laplacian smoothing of noisy surface meshes. In *Computer Graphics Forum*, volume 18, pages 131–138. Wiley Online Library, 1999.

[WBS15] Oliver Wasenmüller, Gabriele Bleser, and Didier Stricker. Combined bilateral filter for enhanced real-time upsampling of depth images. *International Conference on Computer Vision Theory and Applications*, 2015.

[ZFAT11] Youyi Zheng, Hongbo Fu, OK-C Au, and Chiew-Lan Tai. Bilateral normal filtering for mesh denoising. *Visualization and Computer Graphics, IEEE Transactions on*, 17(10):1521–1530, 2011.

Using Mutual Independence of Slow Features for Improved Information Extraction and Better Hand-Pose Classification

Aditya Tewari*†
Aditya.Tewari@dfki.de

Bertram Taetz*
Bertram.Taetz@dfki.de

Didier Stricker*
Didier.Stricker@dfki.de

Frédéric Grandidier†
Frederic.Grandidier@iee.lu

ABSTRACT

We propose a Slow Feature Analysis (SFA) based classification of hand-poses and demonstrate that the property of mutual independence of the slow feature functions improves the classification performance. SFA extracts functions that describe trends in a time series data and is capable of isolating noise from information while conserving high-frequency components of the data which are consistently present over time or in the set of data points. SFA is a useful knowledge extraction method that can be modified to identify functions which are well suited for distinguishing classes. We show that by using the orthogonality property of SFA our information about classes can be increased. This is demonstrated by classification results on the well known MNIST dataset for hand written digit detection.

Furthermore, we use a hand-pose dataset with five possible classes to show the performance of SFA. It consistently achieves a detection rate of over 96% for each class. We compare the classification results on shape descriptive physical features, on the Principal Component Analysis (PCA) and the non-linear dimensionality reduction (NLDR) for manifold learning. We show that a simple variance based decision algorithm for SFA gives higher recognition rates than K-Nearest Neighbour (KNN), on physical features, PCA and non-linear low dimensional representation. Finally, we examine Convolutional Neural Networks (CNN) in relation with SFA.

Keywords

Slow Feature Analysis, Hand-Pose Identification, Knowledge extraction, Feature Learning

1 INTRODUCTION AND BACKGROUND

The hand is probably the most effective tool for indicating and gesticulating. Estimating the hand-pose in frames of a sequence to detect a gesture is a common step used in various gesture recognition approaches. Hand gesture recognition is steadily gaining popularity in tasks like navigation, selection and manipulation in Human Computer Interactions [BVBC04]. While complex applications like surgical simulation and training systems require dynamic hand gesture recognition [LTCK03], simpler command and control interfaces often employ hand-poses.

The hand-pose at each frame is treated as a feature in some approaches [CGP07], while some methods use this information to describe the states of a state machine [GMR+02]. A sensor free, vision based detection of pose is a challenging task because of the large degree of freedom in the movement of hand parts and self occlusion that might occur, moreover the calculation of local edge or corner based features is prone to noise [CGP08]. Some methods use physical features of the hand like the gravitational center of the palm region and the finger location [RYZ11]. Other features include convexity that describes the curvature of the palm hull. The works of [PKK09, CLEL12] describe the use of geometrical descriptors for posture detection. We argue that because of occlusion and the high degree of freedom, high level features learnt from hand-pose data can help in improving the classification. In [LCP12] a method of manifold embedding for articulated hand configuration detection is proposed. This method learns one of the global description of data by identifying the manifold on which the data resides.

The SFA allows unsupervised learning of invariant or slowly varying features. It can learn translation, scale and rotational invariances [WS02]. The SFA technique has been modified to achieve supervised learning to

* DFKI, Augmented Vision, Trippstadter Str.122, Germany, 67663 Kaiserslautern.
† IEE-SA, Weiergewan, 11-rue Edmond Reuter, Luxembourg, 5326 Contern.

achieve classification [Ber05]. It provides mutually orthogonal features thus the prominent features carry independent information about the data even though they remain invariant to size, rotation and translation. Another important property of the SFA is the guaranteed optimisation to the slowest changing function which allows for easy extension when learning a new class. We propose to learn several slow feature functions for each class to improve classification further. To achieve this we employ the property of mutual orthogonality of features learnt from a class. The mutual orthogonality of SFA features result in aggregation of information thus it increases the effective information that a classifier receives.

Section 2.1 describes the basic ideas behind SFA, further we discuss its use as a classifier in section 2.2. In section 2.3 we explain the use of orthogonality to increase information and describe its effect on the classification task on the MNIST dataset in section 3.1. Finally in section 3.2 we apply the technique on hand-pose classification.

We ascertain the applicability of SFA as an information extraction method, by demonstrating better classification rates as compared to the standard PCA and manifold learning methods. The manifolds representation of data compensates for non-linearities. The better performance of SFA over manifold learning proves its strong capability of identifying the consistent properties of signals in a dataset. Apart from the comparison with PCA and manifold learning methods we also make comparisons to classification performed by using shape descriptors and geometrical features calculated from the hand-pose images. We report a substantial improvement over the classification done with these features. The improvement over physical features indicates that SFA is capable of information extraction while the improved classification compared to manifold embedding establishes the ability to handle non-linearities in a dataset.

The broad contributions made through this work are:

- The applicability of SFA for hand-pose classification using data obtained from a time of flight camera.
- SFA classification based on several slow feature functions and not just the principal slow feature.
- A comparison of classification based on physical features and SFA features that indicates the superior information extraction capability of SFA.
- The demonstration of improved classification performance on the MNIST hand written digit dataset and the hand-pose dataset using a modified SFA.

2 SLOW FEATURE ANALYSIS

2.1 Slow Feature Analysis as a Learning Problem

Low level features are short duration features and are often misleading. High level features of the data carry information that extends beyond small neighbourhoods. SFA learns functions that represent such high level features. These high level representation can better explain the property of the data space. A feature that does not vary rapidly, yet has a slow consistent change promises to describe the behaviour of a function in better detail [Föl91]. The slow features thus provide a consistent trend in the data. The SFA is originally designed for detection of trends in temporal data [WS02]. It has been modified to provide consistent trends within elements belonging to a static dataset [Ber05]. We first discuss the SFA procedure for temporal data and shall later explain the modifications for classification in static datasets.

If a vectorial input $\mathbf{X}(\mathbf{t}) \in \mathbb{R}^d$ is a time series, one of the slow features is the function $\mathbf{g}(\cdot)$, such that $\mathbf{y}(\mathbf{t}) = \mathbf{g}(\mathbf{X}(\mathbf{t}))$, varies as slowly as possible while avoiding trivial responses.

The problem is formally described by [Wis03] as minimising the absolute differential

$$\Delta(y_j) := \langle \dot{y_j}^2 \rangle. \tag{1}$$

Here y_j is the j^{th} component of $\mathbf{y}(\mathbf{t})$ and $\dot{y_j}$ is the derivative of y_j with respect to time t and $\langle\cdot\rangle$ denotes average over time. The absolute differential is minimised under the following conditions:

$$\langle y_j \rangle = 0 \tag{2}$$

$$\langle y_j^2 \rangle = 1 \tag{3}$$

$$\langle y_i y_j \rangle = 0 \quad i \neq j. \tag{4}$$

While the minimisation selects invariant features, (3) forces some variance and removes the possibility of obsolete solutions like a constant function and (4) forces independence among the calculated slow features. These constraints are forced by sphering the data [LZ98].

Sphering of $\mathbf{X} \in \mathbb{R}^d$ means we transform $\mathbf{X}$ such that the covariance matrix of the transformed random variable $\mathbf{X}^*(\mathbf{t})$ is an identity matrix. $\mathbf{X} = (x_1, x_2 ..., x_n)$, represents a data matrix and $x_1, x_2, x_3, ..., x_n$ are n vectors belonging to it. If $(\mathbf{X} - \mu)$ and Σ are respectively the centered data matrix and the covariance matrix, then the sphered data is expressed as:

$$\mathbf{X}^*(\mathbf{t}) = B_n(\mathbf{X} - \mu), \quad with \quad B_n^T B_n = \Sigma^{-1}. \tag{5}$$

The sphered data $\mathbf{X}^*(\mathbf{t})$ is projected into a quadratic space, resulting in data $\mathbf{Z}$. The derivative $Z(t+1) - Z(t)$, is represented by $\dot{Z}$. Let $\mathbf{W}$ be the

eigenvectors of the covariance matrix of the derivative matrix $\dot{\mathbf{Z}}$,

$$\langle \dot{\mathbf{Z}}\dot{\mathbf{Z}}^{\mathbf{T}} \rangle \mathbf{W} = \lambda_{\mathbf{W}}. \quad (6)$$

The eigenvectors corresponding to the smallest eigenvalues are the direction of the slowest change in differential of the data. These eigenvectors compose the slow feature functions. These functions are the weighted linear sums over the components of the expanded signal, where weights are the components of eigenvectors w,

$$g_j(x) = {w_j}^T . Z(t). \quad (7)$$

Where w_j is the j^{th} column of the matrix $\mathbf{W}$. The m smallest eigenvalues correspond to the m primary slow feature functions: $g_1, g_2, g_3 \ldots g_m$.

2.2 Slow Feature Analysis for Classification

The slow features describe intrinsic features of a long time series. It is the property of slow features to conserve variations over time, this property can be exploited for classification. The data for classification is not temporal and thus the absolute differential described in (1) is modified to perform a supervised classification. To perform a supervised classification, functions resulting in minimum inter-element difference within each class are identified. As in case of time series SFA, the conditions of zero mean, constant variance and linear independence are imposed. Once again these conditions are satisfied by sphering the data. Furthermore, the optimisation process tries to increase the variance outside a class, to identify the slow feature functions.

For the dataset $\mathbf{X}$, we define a matrix $\mathbf{Z}$, such that $\mathbf{Z}$ is the quadratic expansion of the sphered transform of $\mathbf{X}$. Accordingly, the differential term for a vector z^{el} belonging to the expanded dataset $\mathbf{Z}$ is represented as:

$$\nabla_{el} := \sum_{C=1}^{N} \sqrt{\sum_{n=1}^{N_C} (z_C^n - z^{el})^2}. \quad (8)$$

Thus average differential for the data $\mathbf{Z}$ can be re-represented as:

$$\nabla := \langle \nabla_{el} \rangle. \quad (9)$$

Where, z^{el} is the vector corresponding to the element for which the differential is calculated. z_C^n is the n^{th} element of a class C, N is the number of classes and N_C is the number of datapoints in the class C. We now minimise the value of ∇. This minimisation condition returns functions that forces slow variance within classes. Each of the slow features correspond to one of the classes, to further improve the extracted feature functions, (9) is extended to maximise the variance between classes while minimising it within the class [ZT12].

To achieve this we subtract the average of the absolute difference of the in-class element with elements outside the class (∇_{el}^o) from the average differential within the class (∇_{el}), that yields

$$\nabla_{el}^o := \sum_{C=1}^{N} \sqrt{\sum_{\{c=1, c\neq C\}}^{N} \sum_{n=1}^{N_c} (z_c^n - z^{el})^2}. \quad (10)$$

The calculation of the slow feature function is modified to minimising the cost function O, where O is defined as:

$$O = \langle \nabla_{el} \rangle - \langle \nabla_{el}^o \rangle. \quad (11)$$

2.3 Using Orthogonality to Increase Information

The classification process described above returns (N=number of classes) functions. These functions are learnt from the entire dataset using the optimisation function of (11). This procedure results in a set of functions which provide low variance response. The constraint of decorrelation between different slow features creates the possibility of learning many functions corresponding to one class.

The ready availability of features after doing an SFA procedure, and there mutual independence motivates us to find more features within a class. Thus we calculate multiple slow features corresponding to each class. Rather than learning slow features over the entire dataset we learn a set of function for every class. Slow features are learnt by restricting the dataset to elements of one class, this is repeated for all classes.

As each function is orthogonal, we have more than one function representing intrinsic properties of the specific class. These linear functions are decorrelated on the expanded space. Learning slow features in every class requires a larger training dataset, meanwhile it also results in adding information for classification. The optimisation function (11) is further modified to minimise variance within a class, while maximising out-of-class variation using all other classes (13). This modification extends (10) as follows:

$$\nabla_{el_C} := \sqrt{\sum_{n=1}^{N_C} (z_C^n - z^{el_C})^2}, \quad (12)$$

$$\nabla_{el_C}^o := \sqrt{\sum_{\{c=1, c\neq C\}}^{N} \sum_{n=1}^{N_c} (z_c^n - z^{el_C})^2}, \quad (13)$$

$\nabla_{el_C}^o$ in (13) is the sum of out-of-class variances calculated over the training dataset.

$$O_C = \langle \nabla_{el_C} \rangle - \langle \nabla^o_{el_C} \rangle. \quad (14)$$

el_C represents that the calculation for the differential is done for elements belonging to the class C. The optimisation for class C is achieved by minimising O_C.
The functions are collected as matrix $\mathbf{W_C}$ where C is the class for which these functions are learnt. $w_{\lambda_{C_j}}$ is the vector corresponding to the j^{th} eigenvalue λ_{C_j} of class matrix $\mathbf{W_C}$. For a test input vector $\mathbf{P}$ the functional $\mathbf{G}$ returns an output vector $\mathbf{G}(\mathbf{W},\mathbf{P})$. The functional $\mathbf{G}$ has m linear functions in the space corresponding to the dimension of vector expanded in data space,

$$G(\mathbf{W_C},\mathbf{P}) = \mathbf{P} \cdot \bar{\mathbf{W}}_{\mathbf{c}}{}^T. \quad (15)$$

The variance for the output of the function is calculated as,

$$Var_C \quad = \sum_j (\mathbf{P} \cdot \bar{w}_{\lambda_{C_j}})^2 \quad = \sum G(\mathbf{W_C},\mathbf{P})^2. \quad (16)$$

The final classification is performed as follows:

$$class = \underset{C}{\operatorname{argmin}}(Var_C). \quad (17)$$

While doing an N class classification using m functions for each class, we have Nm functions. Some of these functions are very similar even though they belong to separate classes. This does not affect the minimum variance choice, because of aggregation.
The value of functions corresponding to a class when applied to an element from the same class is centred around a constant value. When a function is applied on a mismatched class, the result is random. This randomness likely results in a wrong identification.
In the case of multiple centred functions, corresponding to a class, the resulting output for a matching sample has all the function outputs centred around zero. Some functions from non-matching classes may return centred responses close to zero but, the aggregated variance for a mismatch element is higher, resulting in clearer distinction from the matching class.

3 EXPERIMENTS

3.1 Effect of Increased Information on MNIST Dataset

MNIST dataset [LC12] is one of the most popular dataset for evaluating classification problems. The Lecun network [LJB+95] has achieved an error rate of less than 0.3% on the MNIST dataset. [Ber05] also describes the original classification technique on the MNIST Hand written digit dataset. We further tested and compared both methods of using SFA for classification described earlier on the same dataset. Each datapoint in the MNIST dataset is a 28x28 pixel image. We reduce it to a 35 dimensional vector by employing PCA and then project it into a quadratic space. The quadratic expansion of the 35 dimensional PCA vector results in a vector of size 630. We calculate 10 slow features functions for the full dataset. Also, we calculate 10 slow feature functions for each class. It was observed that the identification performance for every class improved when we used the property of orthogonality to calculate slow feature functions. The comparative results are listed in Table 1.

class	0	1	2	3	4	5	6	7	8	9
Full Dataset	81	93	79	83	77	72	77	80	73	84
Class Separation	91	96	82	85	79	81	89	91	83	84

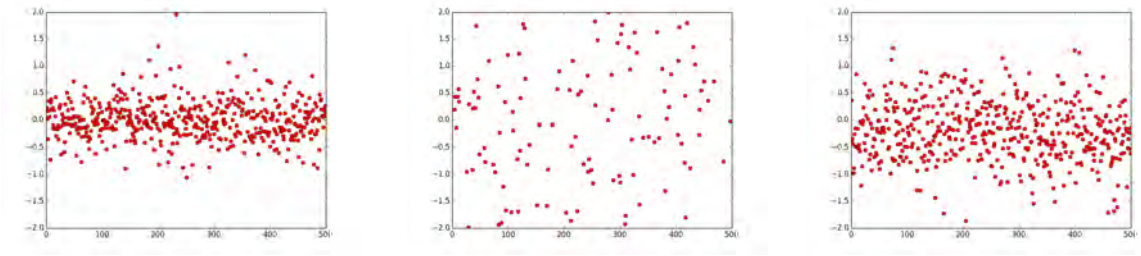
Table 1: Classification accuracy in % for each digit mentioned on the top row. The second row values are accuracy percentages when slow feature functions are learnt from the entire dataset, the third row shows the accuracy percentages when several functions are learnt independently for each class

Figure 1 and Figure 2 show the difference between the two methods for classification. Figure 1 is based on identification of feature function from the entire data while Figure 2 is based on the classification approach where multiple corresponding functions are learnt from each class. The Y axis represents the distance of the response from the mean response calculated during the training stage, the X axis marks the index of input element on the dataset. The input elements are stacked in order of the classes that they belong to.
Figure 1 shows the centred response of the first three classes to the function corresponding to class with digit 0. The deviation of elements of class '0' from the origin are smaller as compared to other classes. This fits our hypothesis that SFA looks for feature functions that minimise the in-class variance. The Figure 2 shows the response of each data point to three functions learnt for class 0. The response of the data points of each class is shown in the same figure, with dark blue (the first cluster) representing class 0. The lower variance of function value to the matching class is clearly visible in these figures, the aggregation of function 1, 2 and 3 results in a deviation which is smaller for the matching class, but higher for mismatch. Averaging over these function values reduces the likely possibility of error in the first method because of randomness of non matching function response.

3.2 Hand-pose Experiments

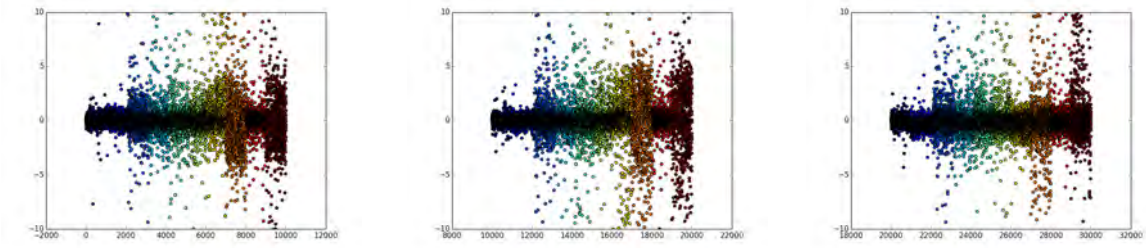
3.2.1 Hand-Pose Data Collection

A 3D Time-of-light, PMD-Nano camera has been used to collect a dataset of hand-poses. The camera is fixed vertically above the palm. The output of the PMD-Nano time of flight camera is an 120x165x2 image. The two channels of the image are the amplitude



(a) class:0 (b) class:1 (c) class:2

Figure 1: Response of class 0, 1 and 2 to function learnt from class 0.



(a) function:1 (b) function:2 (c) function:3

Figure 2: Response of all classes to the first 3 functions learnt from class 0.

value and the depth map image data. We cover the arm region with absorbent clothing and use the reflectance of skin to identify the palm. The reflectance constraint does not entirely remove the background and thus the closest contour greater than a threshold area is chosen as the palm region. The segmented palm region is then converted into a binary image which is further used for hand-pose identification.

We then learn slow feature functions for five hand-pose classes labelled as "Fist", "Flat", "Index", "Open" and "Grab", see Figure 3. Slow features or invariances are learnt from a dataset of 3,000 frames of each class from 3 subjects. 1000 frames in each class are randomly selected and rotated in either direction, by an angle between 10° and 20°. These rotated frames are added to the training dataset along with the original frames. Note that, this spreads the poses such that they cover the whole rotational axis, it also increases the dataset and generates samples which train the SFA for rotational invariances.

Three hundred frames are selected for each class through random partitioning of the original dataset. These samples are used as test dataset, while the remaining original dataset is used for training. The preprocessing follows the same procedure as described for the training dataset.

3.2.2 Hand-pose Identification

Before learning slow features from the dataset of segmented hands, the image is scaled down to one-third of its original size. This is followed by a PCA which reduces each image to a 35 dimension vector that is projected to its quadratic space to allow the learning of non-linear invariances in the principal components of the training data.

During the SFA learning process the covariance matrix

(a) Fist (b) Flat (c) index (d) Open (e) Grab

Figure 3: The hand-pose samples.

of the differential data as well as the eigenvectors corresponding to the largest eigenvalues are recorded. The eigenvectors corresponding to the ten largest eigenvalues correspond to the linear functions used for classification. Each function is centred around the mean values learnt during the training process. It is observed that the samples of matching classes are tightly spread around the mean values of the classes. The class which corresponds to the function has much smaller variance as compared to other classes. Figure 4 shows the response of the test dataset on the most prominent function of the "Fist" class. The data points for each class are represented by a unique color. The "Fist" class which is represented by blue in the figure has relatively tight packing of the data-points as compared to any other class. Like in the previous figures the X axis of the plot represents the data points which are arranged by their labels, and the Y axis represents the centered value of the learnt function.

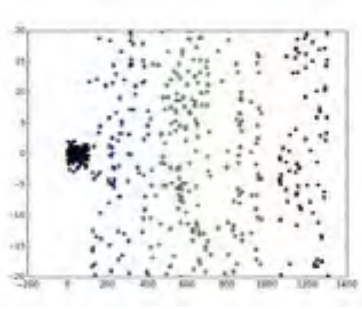


Figure 4: Slow function response for class 'Fist'.

This pattern is visible over the entire set of *Nm* functions, 5x10=50 in the present case. Table 2 shows the response of each data-point to the first five of the 10 learnt functions of each class. Each rows represents a set of functions corresponding to the class.
It can be observed that the functions learnt for one class have lower variance in the same class, while higher variance in other classes. This observation is used to differentiate classes. Thus we calculate the variance of the function response over all the functions calculated for a class.

The three hundred frames of each class in the dataset are used for evaluations. While learning models that are saved include, PCA mapping for each class, the sphering matrix, *m* eigenvectors and the covariance matrices for each class.

4 RESULTS AND COMPARISONS

We compare the results of the classification using slow feature analysis with results from KNN on physical features extracted from each frame. The physical features include coordinates of the tip of the finger (or the tip of the palm), the coordinates of the palm centroid, the convex ratio and the concave depth of the image and the polar and azimuth angle of the finger [RYZ11, PKK09, CLEL12] . We also compare the results to KNN applied on the PCA of the data and the low dimension manifold of the raw binary image [LCP12].
The KNN models for the physical features are generated using 1500 samples from each class and are modelled by simple euclidean distances. The Manifold is learned by Isomap algorithm [TDSL00] and the learning is done by the same training data as used for slow feature analysis.
Slow Feature Analysis based classification works better than the physical feature based classification evaluated in the KNN model. It also outperforms the KNN evaluation done with 35-dimensional (35-D) PCA and 9-dimensional (9-D) manifold representation of the dataset. We chose 35-D PCA because it is used as the basis for SFA calculation and 9-D isomap because the classification by KNN performs best for it. Table 3 shows the confusion matrix for the SFA based classification, Table 4 shows the confusion matrix for classification on KNN model trained on the hand crafted physical features. Table 5 is the confusion matrix for classification results from KNN model trained on the 35-D PCA representation of the image data. While classifying on the 9-D element vector received from the isomap done on the palm region as described earlier, the results are improved as compared to KNN on physical features and PCA based KNN. Table 6.
The results from the SFA are considerably better than the results from the physical features. These features are carefully selected for hand-pose estimation. This underlines the ability of the method to search for relevant features in a class. This improvement also suggests that SFA is capable of reducing the effect of local noise and distortion.
We compare SFA with KNN on the lower dimension representation of the data computed by PCA. The confusion matrices of Tables 3 and 5 clearly demonstrate that SFA performs far better. Thus the process of calculating the slow feature functions after doing PCA on the data further refines the knowledge that we are able to extract from the dataset.
SFA classification also performs better than a KNN model trained on manifold representation of the dataset. While the identification of the "Flat" hand-pose is better than the SFA in case of the isomap representation, the overall performance of SFA is superior. It is notable that KNN is a far more complex model as compared to simpler variance based classification of SFA. This result suggests that SFA is capable of managing non-linearities in the data, this can be attributed to the step in which the PCA data is projected onto a quadratic space.
The improvement from PCA to isomap modelling is a result of better handling of non-linearities in the data. The KNN model based on euclidean distances suffers from the inability to compensate for non-linearities, this is overcome when we use the isomap projection. It is also important to note that while the KNN model is learnt over the isomap projection, SFA classification provides better results by simple variance calculations. It is worth mentioning that the performance improvement in the quality of classification was minimal when we scaled the palm region by distances. This observation can be attributed to the characteristic of SFA that, it explores multidimensional linear functions which encompasses the invariances over the data points.

Discussion

The SFA, as demonstrated in the last section, performs well for the classification task. Even though the total labelled data available to us was small, we compared the performance of SFA classification for hand-pose with CNN. The CNNs have resulted in exceptional classification results. As mentioned earlier Lecun network

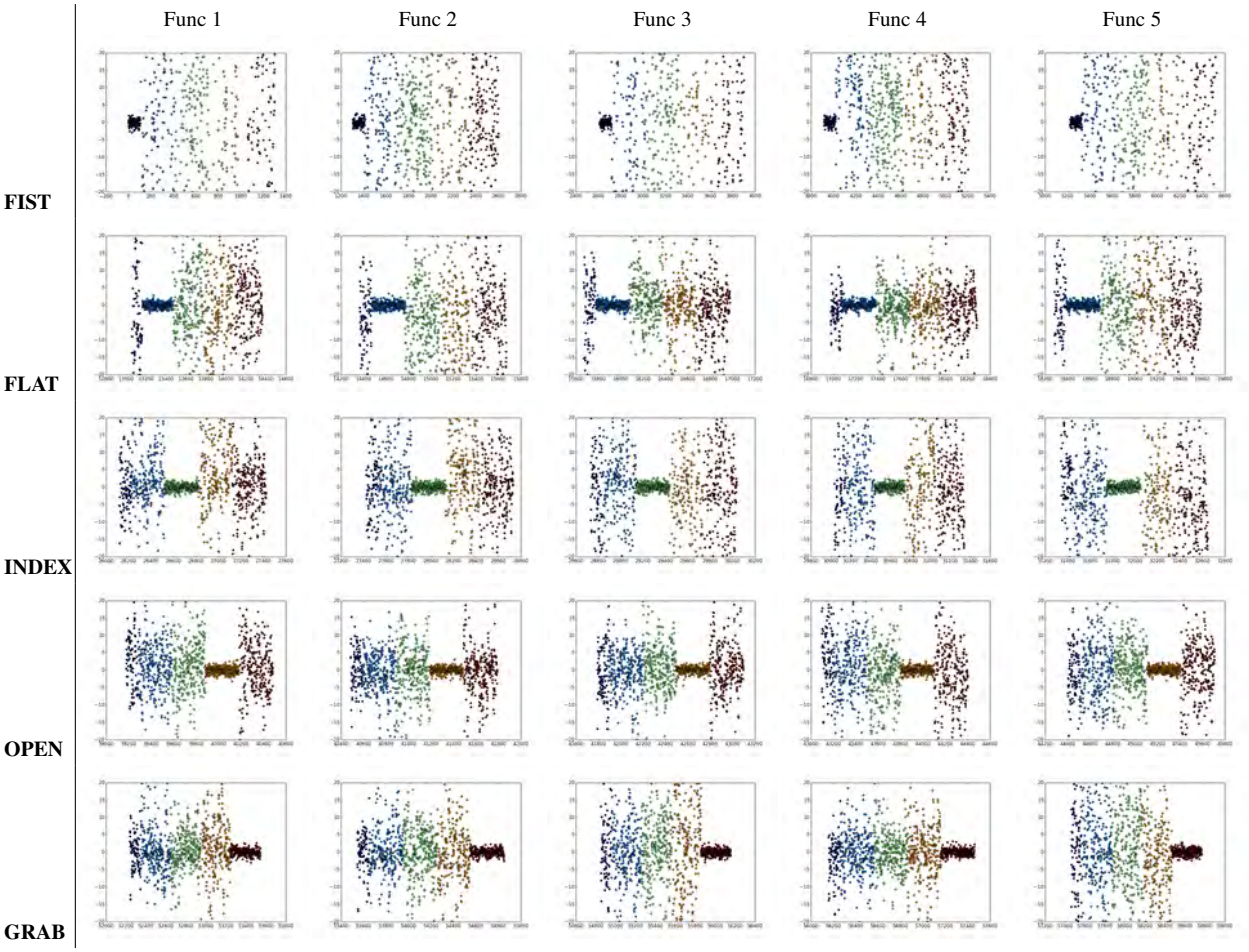


Table 2: Scatter map showing the value for 5 SFA functions for every class on the test dataset, different colors represent different classes.

based on CNN has achieved an error rate of less than 0.3% on the MNIST dataset, this compares favourably with human accuracy. We tested our hand-pose dataset for training a CNN with two convolution layers and two Max pooling layers. Using 15000 data-points after rescaling. The accuracy of classification reached over 98% after 30,000 iteration with a batch size of 50 images. Although, it was observed that because of the relatively small amount of data the CNN model starts over-fitting. The use of easily available, less specific hand-pose datasets for pre-training the CNN is one of the possible methods of overcoming the problem of over-fitting with the present data. SFA also requires a large dataset but lesser than CNN, we demonstrate that it is capable of learning functions for each class of hand-poses with 3000 data points. It can be argued that the SFA learning process results in learning of information that defines the class of the dataset, but the convolutional features learnt by a CNN using the classification based method contain information that distinguishes different classes. SFA results in lesser classification accuracy than CNN on a large dataset, but SFA gives interpretation about the nature of the class independent, which seems to be harder to identify in a CNN model.

5 CONCLUSION

In this paper we used SFA for classification on two datasets. SFA was tested on MNIST dataset and a hand-pose dataset. We approached the classification by training SFA separately for each class and demonstrated that, the property of orthogonality of SFA helps in extracting more information about the class. We showed that SFA outperforms hand picked physical features for hand-pose classification. This confirms the recent trend of preferring global features which are learnt from the data over extracting features by intuition. Training and test data has considerable variances of rotation and scale, in our experiments SFA remains robust to such variances.

The use of slow feature analysis also reduces the on line processing required on the test sample. SFA based classification requires a relatively large dataset for training, additionally it employs an expensive batch learning algorithm which requires large computer memory to run. Yet, it displays a remarkable ability to extract information and identify trends in a dataset. Usually calculating

%	FIST	FLAT	INDEX	OPEN	GRAB
FIST	**97.0**	1.0	0.0	1.7	0.3
FLAT	0.0	**96.7**	2.3	1.0	0.0
INDEX	0.0	0.0	**98.7**	1.3	0.0
OPEN	1.0	0.0	1.3	**97.6**	0.0
GRAB	0.7	2.3	0	0.3	**96.7**

Table 3: Confusion matrix for SFA classification. Bold values are accuracy values for the class corresponding to the respective row.

%	FIST	FLAT	INDEX	OPEN	GRAB
FIST	**97.0**	0.7	1.3	0	1.0
FLAT	0.7	**95.7**	3.0	0	0.7
INDEX	2.7	5.7	**91.7**	0.3	0.0
OPEN	3.0	2.3	0	**94.3**	0.3
GRAB	0.7	4.7	0	0.3	**94.3**

Table 4: Confusion Matrix for KNN classification based on physical features. Bold values are accuracy values for class corresponding to the respective row.

%	FIST	FLAT	INDEX	OPEN	GRAB
FIST	**78.3**	12.2	2.9	3.8	2.9
FLAT	1.3	**80.7**	6.3	6.6	5.0
INDEX	0.0	3.3	**81.7**	2.0	14.0
OPEN	0.0	7.7	4.7	**85.3**	2.3
GRAB	0.3	3.3	7.7	3.0	**85.7**

Table 5: Confusion Matrix for KNN classification results on 35-D PCA. Bold values are accuracy values for the class corresponding to the respective row.

%	FIST	FLAT	INDEX	OPEN	GRAB
FIST	**97.0**	0.3	2.0	0.7	0
FLAT	0.3	**98.3**	1.3	0	0.3
INDEX	3.7	0.3	**96.0**	0	0
OPEN	1.7	0.0	1.0	**96.3**	1.0
GRAB	2.7	0.3	0.3	0.7	**96.0**

Table 6: Confusion Matrix for KNN classification on 9-D isomap on raw images. Bold values are accuracy values for the class corresponding to the respective row.

features at run time is a hard task, it consumes considerable computing and development effort. Whereas, SFA requires few linear operations to calculate the slow features. Thus, it does not only improve the robustness towards the data but also improves the performance of the machine when compared with processes that use physical features.

We showed the performance on global SFA features in this work and compared it to physical (local) features. Note that when we tested the SFA for classification of fixed length time series sequences, local features like peaks and inflexion, when combined with slow features, improved the classification performance. Classification was made using a logistic regression classifier. However, this fusion requires online feature calculation and a more complex classifier model.

It will be interesting to further study and quantify the effect of noise and poor segmentation on these features. Also further experiments with various data sources and the influence of an increasing number of classes on the orthogonality property of SFA will be of interest. We plan to extend the present approach of pose detection to gesture recognition. The batch learning approach is not suitable for the gesture classification and recently developed incremental SFA [KLS11] is a promising solution to the problem.

6 ACKNOWLEDGMENT

This work is supported by the National Research Fund, Luxembourg, under the AFR project 7019190.

7 REFERENCES

[Ber05] Pietro Berkes. Pattern recognition with slow feature analysis, February 2005.

[BVBC04] Volkert Buchmann, Stephen Violich, Mark Billinghurst, and Andy Cockburn. Fingartips: gesture based direct manipulation in augmented reality. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 212–221. ACM, 2004.

[CGP07] Qing Chen, Nicolas D Georganas, and Emil M Petriu. Real-time vision-based hand gesture recognition using haar-like features. In *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1–6. IEEE, 2007.

[CGP08] Qing Chen, Nicolas D Georganas, and Emil M Petriu. Hand gesture recognition using haar-like features and a stochastic context-free grammar. *Instrumentation and Measurement, IEEE Transactions on*, 57(8):1562–1571, 2008.

[CLEL12] J-F Collumeau, Rémy Leconge, Bruno Emile, and Hélène Laurent. Hand gesture recognition using a dedicated geometric descriptor. In *Image Processing Theory, Tools and Applications (IPTA), 2012 3rd International Conference on*, pages 287–292. IEEE, 2012.

[Föl91] Peter Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.

[GMR+02] Namita Gupta, Pooja Mittal, S Dutta Roy, Santanu Chaudhury, and Subhashis Banerjee. Developing a gesture-based interface. *Journal of the Institution of Electronics and Telecommunication Engineers*, 48(3):237–244, 2002.

[KLS11] Varun Raj Kompella, Matthew D Luciw, and Jürgen Schmidhuber. Incremental slow feature analysis. In *IJCAI*, volume 11, pages 1354–1359, 2011.

[LC12] Yann LeCun and Corinna Cortes. The mnist database of handwritten digits, 1998, 2012.

[LCP12] Chan-Su Lee, Sung Yong Chun, and Shin Won Park. Articulated hand configuration and rotation estimation using extended torus manifold embedding. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 441–444. IEEE, 2012.

[LJB+95] Yann LeCun, LD Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, UA Muller, E Sackinger, Patrice Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261:276, 1995.

[LTCK03] Alan Liu, Frank Tendick, Kevin Cleary, and Christoph Kaufmann. A survey of surgical simulation: applications, technology, and education. *Presence: Teleoperators and Virtual Environments*, 12(6):599–614, 2003.

[LZ98] Guoying Li and Jian Zhang. Sphering and its properties. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 119–133, 1998.

[PKK09] Giorgio Panin, Sebastian Klose, and Alois Knoll. Real-time articulated hand detection and pose estimation. In *Advances in Visual Computing*, pages 1131–1140. Springer, 2009.

[RYZ11] Zhou Ren, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 1093–1096, New York, NY, USA, 2011. ACM.

[TDSL00] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[Wis03] Laurenz Wiskott. Slow feature analysis: A theoretical analysis of optimal free responses. *Neural Computation*, 15(9):2147–2177, 2003.

[WS02] Laurenz Wiskott and Terrence Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

[ZT12] Zhang Zhang and Dacheng Tao. Slow feature analysis for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):436–450, 2012.

Localized Search for High Definition Video Completion

Jocelyn Benoit
École des arts numériques, de l'animation et du design
Montreal, Canada
jbenoit@nad.ca

Eric Paquette
Multimedia Lab, École de technologie supérieure
Montreal, Canada
eric.paquette@etsmtl.ca

ABSTRACT

This paper presents a new approach for video completion of high-resolution video sequences. Current state-of-the-art exemplar-based methods that use non-parametric patch sampling work well and provide good results for low-resolution video sequences. Unfortunately, because of memory consumption problems and long computation times, these methods handle only relatively low-resolution video sequences. This paper presents a video completion method that can handle much higher resolutions than previous ones. First, to address the problem of long computation times, a dual inpainting-sampling filling-order completion method is proposed. The quality of our results is then significantly improved by a second innovation introducing a coherence-based matches refinement that conducts intelligent and localized searches without relying on approximate searches or compressed data. Finally, with respect to the computation times and memory problems that prevent high-resolution video completion, the third innovation is a new localized search completion approach, which also uses uncompressed data and an exact search. Combined together, these three innovations make it possible to complete high-resolution video sequences, thus leading to a significant increase in resolution as compared to previous works.

Keywords

Video completion, high-resolution, object removal, patches coherence, localized search, multi-resolution

1 INTRODUCTION

Both image and video completion are important tasks in many multimedia applications. Their goal is to automatically fill missing regions of an image/video in a visually plausible manner. Two key factors differentiate video completion from image completion. Firstly, for video completion, it is important to maintain temporal consistency since human vision is more sensitive to temporal artifacts than to spatial artifacts. Using an image completion technique individually on each frame produces undesired temporal artifacts. Secondly, it is more important for video completion to be time- and memory-efficient since video contains much more data than image.

In the past years, many new solutions have been proposed for video completion. It has been shown that exemplar-based methods, that use *non-parametric patch sampling*, work well and provide good results. Unfortunately, they work only on relatively low-resolution videos because larger ones require too much memory. Few methods [8, 12] present results for 640×480 or 540×432 resolutions, with most [6, 7, 9–11, 15, 17, 20, 21] presenting results of 320×240 or lower resolutions. Since High Definition (HD) videos with 1920×1080 or higher resolutions are now commonplace, most of these methods cannot be applied directly or they require too long computation times.

To understand the proposed method, we must first look at the non-parametric patch sampling approaches. Those methods are based on an iteration through each of the patches in the missing regions and a search in all of the patches of the existing regions to find the most similar patch. Without optimization, this search can be excessively time consuming: $O(m^3M^2F)$ with M representing the video width and height; m the patch width, height and depth; and F the number of frames. Even with optimization methods, the search time of the non-parametric patch sampling approaches still remains excessive. Furthermore, the structures needed for theses optimization methods require too much memory, making them inappropriate for HD videos.

Rather than focusing on the acceleration of the nearest neighbors search, the proposed method narrows the search space at finer (higher) resolutions using information obtained at coarser (lower) resolutions. First, let us consider two patches at coarser resolutions: patch w_p^l from the missing region and its most similar patch $w_{p'}^l$ from the existing region. The most sim-

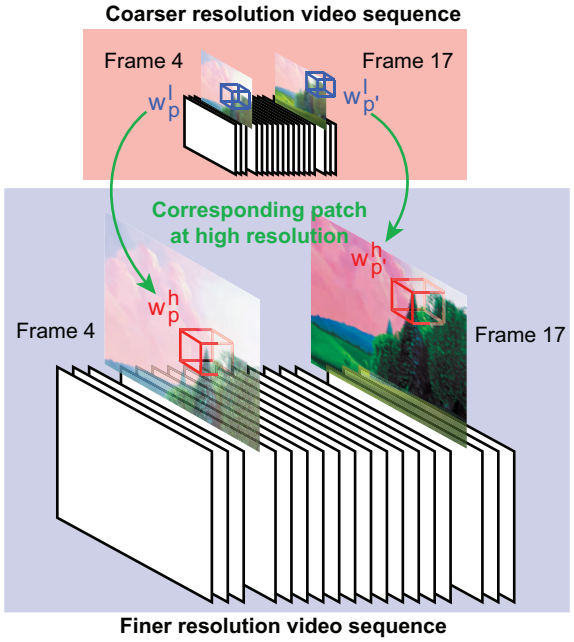


Figure 1: First row: coarser resolution video. Patch (w_p^l) in the missing region and its most similar patch ($w_{p'}^l$) in the existing region. Second row: finer resolution video. The corresponding patch (w_p^h) of w_p^l and its most similar patch ($w_{p'}^h$) in the existing region.

ilar patch of the corresponding patch w_p^h at finer resolutions is likely to be found near $w_{p'}^h$, as illustrated in Figure 1. The proposed approach begins by completing the video at coarser resolutions using a dual inpainting-sampling filling-order completion approach based on Wexler et al. [20]. Since efficient but approximate search approaches are used to find the most similar patches, errors are introduced and several matches are sub-optimal patches. To solve this problem, a coherence-based matches refinement process is used to search for better matches. The technique then stores the space-time location of the most similar patch found for each patch of the missing region in a matches list *ML*. This *ML* is then used by a localized search completion approach to narrow the search space in higher resolution, thus enabling the completion of HD video sequences.

The contributions of the proposed method include a dual inpainting-sampling filling-order completion approach based on Wexler et al. [20]; a new coherence-based matches refinement process that improves the quality of the matches when approximate search approaches are used; and a new localized search completion approach based on an exact search using uncompressed data but restricted to a localized region. We show that the proposed methods enable the completion of HD video sequences and that they produce visually plausible results within reasonable timeframes. Moreover, the approach requires very little memory at the finest resolution except for the input video storage.

2 PREVIOUS WORKS

In past years, many methods have been proposed to replace missing regions of an image. *Image inpainting* techniques propose to fill the missing region by extending the surrounding existing region until the hole vanishes. These techniques generally work only on small and thin holes. *Image completion* techniques use non-parametric patch sampling and are able to fill even larger missing regions of an image. While video completion methods are based on image completion and inpainting methods, video completion poses the additional challenge of maintaining spatio-temporal consistency. Using image completion or image inpainting methods on each frame independently produces temporal artifacts that are easily noticed by the viewer [3].

2.1 Video completion

Extending the image completion methods based on Markov Random Fields (MRF) and non-parametric patch sampling, Wexler et al. [19, 20] address the problem of video completion as a global optimization, and thus obtain good results on relatively large missing regions. Shiratori et al. [17] proposed a similar approach, but find patches based on motion fields instead of color values. Xiao et al. [21] extend these works by formulating video completion as a new global optimization problem defined over a 3D graph defined in the space-time volume of the video. Liu et al. [10] later have proposed an algorithm with two stages: motion fields completion and color completion via global optimization. The major drawback of all these approaches is the amount of information that must be processed when considering HD video sequences. While some methods use per-pixel searches [19–21], other approaches use larger primitives instead of pixels: Shih et al. [16] use fragments, while Cheung et al. [4] use “epitomes”. Approaches using fragments or epitomes can reduce the search time and improve overall coherence, but per-pixel searches are more likely to correctly restore the fine and subtle details found in HD video sequences.

Many methods segment the video sequence into foreground and background parts [6–8, 11, 14] or into layers [22]. These methods create a static background mosaic of the entire sequence, and as a result, these techniques are limited to video sequences with a static background using a fixed camera. Patwardhan et al. [12] later proposed a framework for dealing with videos containing simple camera motions, such as small parallax and hand-held camera motions. The major drawback with all these techniques is that the pixels replaced are static across the video sequence, thus removing details such as video noise, film grain, or slightly moving objects, such as tree leaves, from the background. At an

HD resolution, this lack of detail is quickly noticed by the viewer.

2.2 Coherence techniques

When Ashikhmin [2] introduced the concept of coherence, he observed that the results of synthesis algorithms often contain large contiguous regions of the input texture/image when using non-parametric patch sampling. Consequently, the independent search for every patch in the input texture/image can be accelerated by using information from previously computed searches. Thus it limits the search space of a given patch to the locations of the most similar patches of its neighbors. We based our coherence-based matches refinement on the same coherence observation and developed an novel approach that is efficient with respect to both computation time and memory consumption. Tong et al. [18] also proposed a coherence technique called *k-coherence*. While this technique improved the search time, the pre-processing time and the memory consumption are major drawbacks for high-resolution video completion methods.

This paper presents an approach for the completion of video sequences that requires very low memory usage and reasonable computation time, making it usable for HD video sequences. Further, it presents a new coherence-based match refinement approach that increases the overall quality of the results by eliminating many noticeable artifacts. Unlike most of the previous works, this paper presents results on video sequences with non-stationary camera movements.

3 HIGH DEFINITION VIDEO COMPLETION

In this section, we present a new video completion approach that is able to automatically fill missing regions of HD video sequences. Section 3.1 presents the approach overview, Section 3.2 explains the dual inpainting-sampling filling-order completion approach, Section 3.3 describes the coherence-based matches refinement process, and Section 3.4 details the new localized search completion method.

3.1 Approach overview

Starting with an input video sequence V containing a missing region or hole H ($H \subset V$), our approach fills H in a visually plausible manner by copying similar patches found in the existing region E ($E = V \setminus H$), thus creating a completed video sequence V^*. This process is shown in Figure 2. In order to maintain spatio-temporal consistency, we consider the input video as a space-time volume, and thus a pixel located at (x, y) in frame t can be represented by the space-time point $p = (x, y, t)$. Consenquently, a patch w_p can be seen as

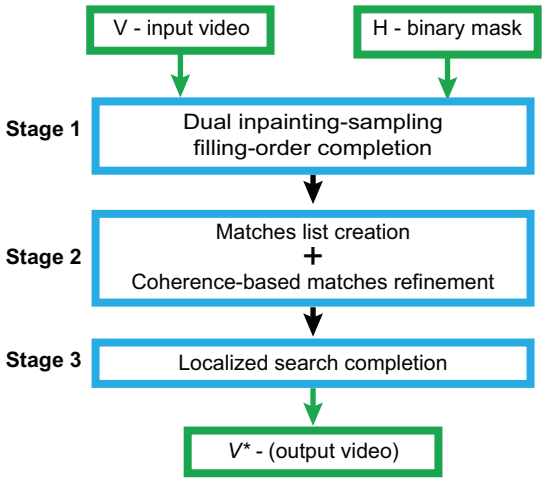


Figure 2: Schematic overview of the proposed approach

V	original video sequence
H	missing region or hole of V, $H \subset V$
E	existing region of V, $E = V \setminus H$
H^*	completed region
V^*	completed video sequence
p_l	point located at (x, y, t) at coarser resolution
w_p^l	patch centered at p_l at coarser resolution
$w_{p'}^l$	patch centered at p'_l, most similar to patch w_p^l
p_h	point located at (x, y, t) at finer resolution
w_p^h	patch centered at p_h at finer resolution
$w_{p'}^h$	patch centered at p'_h, most similar to patch w_p^h
ML	matches list
c	RGB color
S	search region centered at p'_h

Table 1: Symbols definitions

a spatio-temporal cube of pixels centered at p. Table 1 summarizes the symbols used in this paper.

The missing region H is indicated to the system by a binary video sequence in which identified pixels are in H. The binary video sequence can be constructed using object tracking in the video sequence. Many digital motion graphics and compositing softwares already provide accurate and rapid tools to create such binary video sequences. In our experimentations, we relied on such tools to define H.

Figure 2 shows a schematic overview of our approach while Figure 3 presents the detailed steps. First, the input video is downsampled and completed by a dual inpainting-sampling filling-order completion based on the works of Wexler et al. [19, 20] using global optimization and non-parametric patch sampling (see Section 3.2). Completing the video at low-resolution with the proposed approach is efficient and provides good results. When the dual inpainting-sampling filling-order completion is finished, each patch w_p^l (centered at $p_l \in H$) is associated with its best matching patch $w_{p'}^l$ (centered at $p'_l \in E$) by creating a matches list containing

space-time pairs p_l-p'_l for every $p_l \in H$. To complete the search in a reasonable amount of time, the best matching patches $w^l_{p'}$ must be selected using approximate search and data compression methods. Thus, the $w^l_{p'}$ found might not be the best match. The second stage of the proposed method consists of an iterative coherence-based matches refinement process that improves the search results for the worst matching patches $w^l_{p'}$ (see Section 3.3). This stage is efficient, and provides significant quality improvement. Finally, the matches list is used by the localized search completion method to narrow the search space, thus enabling the completion of HD video sequences (see Section 3.4). This final stage of the method is also efficient, and it provides good results at HD resolution.

3.2 Dual inpainting-sampling filling-order completion

A visually plausible completion of a video sequence replaces the missing region H by a completed region H^* where pixels of H^* fit well within the whole video V^*. To achieve this, a video completion approach must satisfy two criteria: first, every local space-time patch of the completed region H^* must be similar to an existing patch of E, and secondly, all patches that fill H^* must be coherent with each other. Consequently, we seek a completed video V^* that minimizes the objective function stated in Equation 1:

$$Coherence(H^*|E) = \prod_{p_l \in H} \min_{p'_l \in E} D(w^l_p, w^l_{p'}), \quad (1)$$

where $D(w^l_p, w^l_{p'})$ is a similarity metric between two patches. The similarity value of two patches is evaluated with the Sum of Squared Differences (SSD) of color information (in the RGB color space) for every pair of space-time points contained in these patches. Wexler et al. [20] added the spatial and temporal derivatives to the RGB components to obtain a five-dimensional representation for each space-time point. In experimentations, RGB alone produced good results for most videos. Problems occured when trying to reconstruct a hidden moving object. While the technique of Wexler et al. [20] can solve these problems, it is however limited to objects with cyclic motion (i.e. like a walking person). Moreover, it requires more memory and computation time. For these reasons, we limited our problem domain to videos without occluded moving objects and chosen to use only RGB components.

The first step of the dual inpainting-sampling filling-order completion approach is to downsample V to a coarse resolution (see Figure 3, Stage 1.1). Then, before starting the completion, the values of each space-time point of H need to be initialized. Unlike Wexler et al. [20] who used random values, the proposed approach fills H using an image inpainting technique [3]

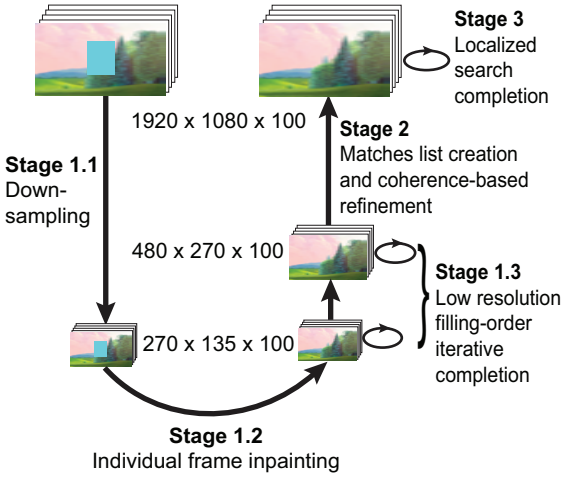


Figure 3: Steps of the proposed video completion approach

(see Figure 3, Stage 1.2). Our aim is to speed up the convergence by using the existing information around H. This initialization is done only once, prior to the first iteration of the low resolution filling-order iterative completion approach.

After the initialization, the approach performs an iterative process, improving the overall coherence of H (see Figure 3, Stage 1.3). During each iteration, the approach seeks a replacement color value for every space-time point in H in order to minimize Equation 1. Unlike previous methods, which used scan-line ordering, our approach fills H using a 3D hole-filling approach, thus ensuring that each patch w^l_p contains information that is more reliable (space-time points in E or space-time points already processed during the current iteration). Consequently, it speeds-up the convergence and reduces discontinuities near the boundaries of H. The patches can have different sizes in the spatial and temporal dimensions. Generally, we used $5 \times 5 \times 5$ patches or $7 \times 7 \times 5$ patches and we based our choice on the element structure size that needs to be completed within the video sequence.

To seek a replacement color c for a space-time point p, the approach uses a single best-matching patch $w^l_{p'}$ that minimizes $D(w^l_p, w^l_{p'})$. When $w^l_{p'}$ is found, the color c' is copied from space-time point p'_l to p_l. Compared to other methods that blend together several matches, using the single best-matching patch does not result in blurring artifacts and preserves film grain and noise from the original video. For these reasons, our approach uses the single best-matching patch.

To enforce spatio-temporal consistency, this iterative process is done on multiple scales using spatial pyramids (see Figure 3, Stage 1.3). Each pyramid level contains $^1/_2 \times {}^1/_2$ of the spatial resolution while maintaining the temporal resolution. The iterative process starts with the coarsest pyramid level and propagates its re-

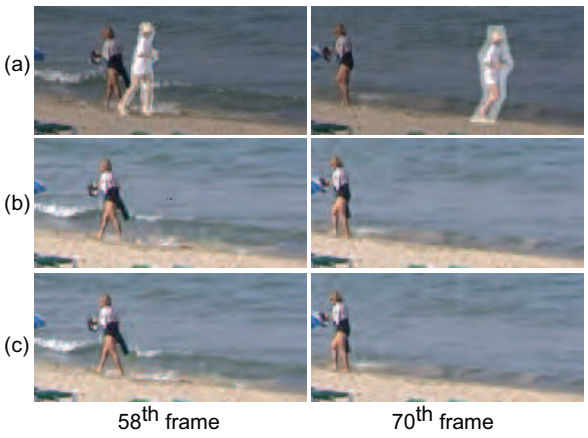


Figure 4: Comparison of the results obtained with the low-resolution video completion approach: (a) Original frames; (b) results from Wexler et al. [20]; (c) results from the proposed method

sults to finer levels. Because it involves long computation times and a lot of memory for the search structure, this iterative process is impractical at finer pyramid levels for HD videos. Therefore, the proposed approach stops the iterative process when it reaches a fixed resolution (typically 480×270).

The proposed dual inpainting-sampling filling-order completion approach produces results with a quality equivalent to the results of Wexler et al. [20], but within much less time. Figure 4 shows the completion results of the "Jogging lady" sequence of Wexler et al. [20] and ours. Wexler's approach took one hour per iteration at the finest resolution level while our approach took less than four minutes per iteration.

3.3 Coherence-based matches refinement

When Stage 1 is over, each patch $w_p^l \in H^*$ has a corresponding patch $w_{p'}^l$. Each space-time point p_l is associated with its corresponding p'_l and the pairs are stored in a matches list *ML*. During the high-resolution completion iterative process, *ML* enables the approach to narrow the search space to only sub-regions of *E*. As a reminder, our key observation is that, for a patch w_p^l at coarser resolution with its most similar patch $w_{p'}^l$, the most similar patch of the corresponding patch w_p^h at finer resolution is likely to be found near p'_h (see Figure 1).

For efficiency reasons, optimization methods such as principal component analysis (PCA) and approximate nearest neighbors search (ANN) [1] are used in Stage 1. While these methods are essential to achieving acceptable search times, they often provide matches $w_{p'}^l$ that do not minimize Equation 1. Consequently, *ML* needs to be refined during Stage 2 (see Figure 3, Stage 2) to have better matching patches $w_{p'}^l$.

Figure 5: Impact of the *ML* refinement iterative process on high-resolution video completion results: (a) Original frame; (b) completed frame without *ML* refinement; (c) completed frame with *ML* refinement. The frames were cropped to better show the missing and completed regions

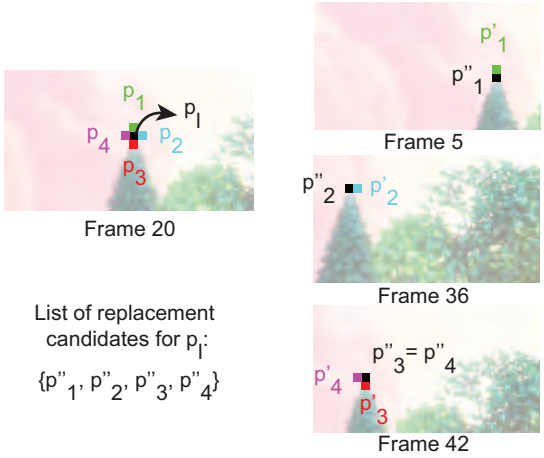


Figure 6: Coherence-based matches refinement process

The information contained in *ML* must be reliable in order for visually plausible results to be possible with the localized search completion iterative process. The patches w_p^l and $w_{p'}^l$ must be highly similar for every pair p_l-p'_l of *ML*, otherwise, the approach will narrow the search space to a region where it is less likely to find the best matching $w_{p'}^h$. Figure 5 (a, b) shows an example where the information contained in *ML* is not reliable. As can be seen, there are many visible artefacts such as the centered window and the left building edge.

To find better matching patches $w_{p'}^l$, we take advantage of the concept of coherence. First, the approach calculates the distance (L_2 norm of uncompressed data) of patches w_p^l and $w_{p'}^l$ for each pair p_l-p'_l from *ML*. Then an iterative process refines pairs with distances higher than a given threshold. During the first iteration, this threshold is set such that 15% of the pairs are refined. After each iteration, this threshold is reduced by 20% of its initial value. For each pair p_l-p'_l above the threshold, the approach seeks for a replacement p'_l that decreases $D(w_p^l, w_{p'}^l)$. Instead of using a brute force approach that searches the entire video sequence, the search is restricted around the best matching patches of

p_l neighbors. An example is shown in Figure 6. The four neighbors of p_l are considered: top, right, bottom, and left; respectively p_1, p_2, p_3, and p_4. For the top neighbor (p_1) the approach considers its previously calculated best matching point p'_1, then from p'_1, its bottom neighbor p''_1 is considered. The L_2 norm is computed between patches p_l and p''_1, and if the norm is lower than the current value, p'_l is replaced by p''_1 and the color from p''_1 is copied to p_l. This process is repeated for p_2, p_3, and p_4. If there is no good replacement, the pair p_l-p'_l is left unchanged, and is considered in the next iteration.

When considering the top neighbor p_1, instead of searching anywhere around its best matching point p'_1, only its bottom neighbor (p''_1) is considered. The rationale behind this is that several successful approaches use large primitives such as fragments or epitomes. When considering larger primitives, the bottom neighbor (p''_1) is the one that would be copied on top of p_l. This effectively reduces the search to only four points (p''_1 to p''_4). To even further reduce the number of points to test, each neighbor p_1 to p_4 is considered only if the L_2 norm of a pair, for example, p_1-p'_1, is below the current threshold. This is a very rapid test since the value is already computed and stored in the *ML*. Since there is a maximum of only four potential points to consider as opposed to the millions from the whole video sequence, this process is extremely fast. Figure 7 shows an example of the *ML* coherence-based matches refinement process that minimizes the distance of w_p and $w_{p'}$ for each pair p_l-p'_l in *ML*. The *ML* refinement provides a significant quality improvement (as shown in Figure 5) within a few seconds.

3.4 Localized search completion

This section presents the proposed approach for completing missing regions of video sequences at HD resolutions. As stated earlier, current exemplar-based methods are unpratical to complete HD video sequences because best match searches require excessive amount of memory and computation time. Many attempts have been made to accelerate this search with optimization methods such as ANN and dimensionality reduction methods such as PCA, but the structures needed for these optimization methods require too much memory for HD video sequences. Instead of accelerating the best match search, the proposed method narrows the search space at HD resolution using information from coarser resolutions.

Before the localized search completion process starts, the information contained in *ML* must to be scaled up to the finer resolution (see Figure 8). For each space-time point $p_h \in H$ at a finer resolution, its corresponding low-resolution p_l is found as well as the space-time location p'_l associated with it. The space-time location

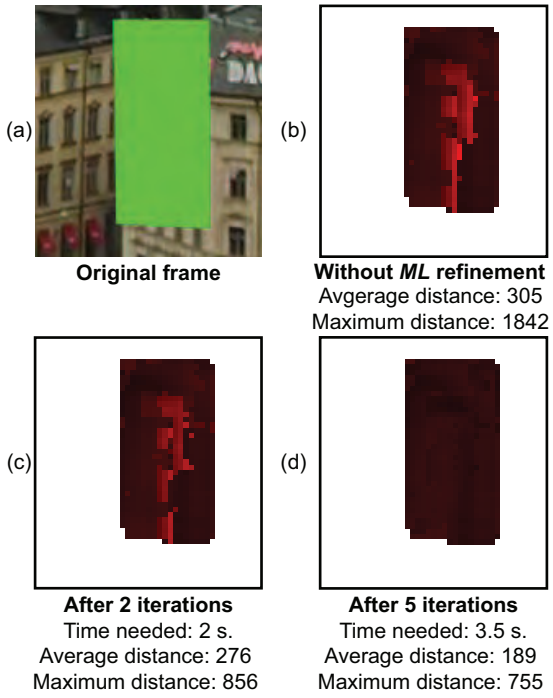


Figure 7: Impact of the *ML* coherence-based matches refinement process: (a) original frame; (b) distance of w_p and $w_{p'}$ for each pair p_l-p'_l after *ML* creation; (c) distances after two iterations of the *ML* refinement process; (d) distances after four iterations of the *ML* refinement process. The frames were cropped to better show the missing and completed regions

p'_l is then scaled up to a finer resolution resulting in p'_h. The pair p_h-p'_h is then added in a new matches list *MLH* which will be used by the localized search completion process to narrow the search space.

The main steps of the localized search completion process are similar to those of the low-resolution process: using a 3D hole-filling approach, the method seeks a

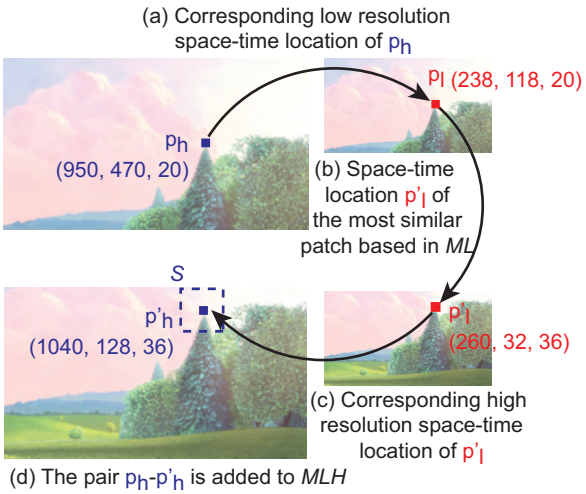


Figure 8: Creation of *MLH* based on *ML*

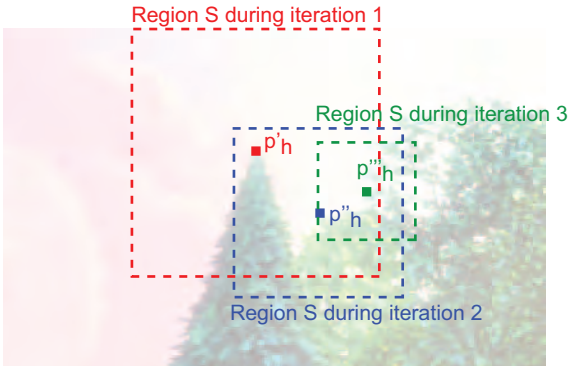


Figure 9: Locations and sizes of search regions S for the first three iterations of the localized search completion process

replacement color c' for every $p_h \in H$ using a single best-matching patch $w^h_{p'}$. However, instead of searching through the entire video sequence using a brute force algorithm or expensive search structures, the approach only searches in a small sub-region S, based on the information from *MLH*. For each space-time point $p_h \in H$, the approach first looks in *MLH* and seeks for its associated p'_h. Then, a small region S centered at p'_h is selected. Next, the approach searches only in S for the best-matching patch $w^h_{p''}$ (located at $p''_h \in S$) and the color c_h is replaced by c''_h. If p'_h and p''_h are different, the pair p_h-p'_h from *MLH* is replaced with the pair p_h-p''_h. In the next iteration of the localized search completion process, the sub-region S will be recentered around this updated space-time location. During the first iteration of the localized search completion process, the window size of sub-region S is 17×17 pixels. This window size is then decreased after each iteration (13×13, 9×9, 5×5). Figure 9 shows an example of the location changes and size decreases of a search region S for three iterations.

Obviously, the search time is dramatically reduced when using *MLH* to narrow the search space, as compared to using methods such as ANN and PCA. When using the proposed *MLH* technique, less than a thousand patches are searched for each p_h compared to the tens of millions of patches from the whole video. Moreover, the computation time for the creation and the refinement of *ML* and *MLH* is shorter than the time needed for the creation of the structures used by ANN and PCA. Another important advantage of the proposed method is that *MLH* requires much less memory than typical search structures, such as ANN. Finally, the proposed *MLH* search does not rely on compressed data, and thus can provide better matches.

4 RESULTS AND DISCUSSION

Figure 10 shows the completion of the "Station" sequence and Figure 11 shows the completion of the "Race to Mars" sequence. The main challenge of these sequences is the constant motion of the camera. The "Station" sequence contains a constant zooming motion while the "Race to Mars" sequence contains complex rotating and panning motions. Video sequences with such motions cannot be handled by video completion techniques using a static background mosaic because the size and orientation of the objects contained in the background are not constant during the entire video sequence.

It can be seen in Figure 10 that the proposed method works well with large missing regions. Figures 10 and 11 demonstrate that the proposed methods produce good results for missing regions containing stochastic texture as well as salient structure. Since state of the art papers introduced in Section 2 show results with resolutions ranging from 320×240 to 640×480, it is not possible to compare the quality of our results with other techniques. Therefore, we used a structural similarity method (SSIM) [13], a full reference metric, to measure the quality of our results at high-resolution. Even though SSIM is generally used to evaluate video compression methods, it can also be used to measure the similarity between a reference sequence and a completed sequence. Figure 12 shows the completion of the "Old town cross" sequence. Considering only the pixels in the missing region instead of all the pixels from the full frames of the sequence, the average SSIM index is 90.63. Since the completed region does not need to be exactly like the reference region, as long as the region is completed in a visually plausible manner, this SSIM index is good.

Table 2 shows a comparison of the proposed approach with earlier works based on different criteria (some were taken from Shih et al. [15]):

- **Missing region specification**: how the user interacts with the method to specify the missing region;
- **Exemplar-based approach**: what type of completion method is used;
- **Camera motion**: video sequences with stationary or nonstationary camera;
- **Maximum resolution**: the highest resolution of the video sequences presented in the paper.

All completion methods use an exemplar-based technique with different variations. Most of the completion methods only use video sequences taken with a stationary camera to test their algorithm. Patwardhan et al. [12] present results with a non-static camera, but the camera motion is always parallel to the projection plane. Thus, Patwardhan et al. [12] do not deal with changes in size, perspective, nor zooming. Only Shih et al. [15] and the proposed method present results with

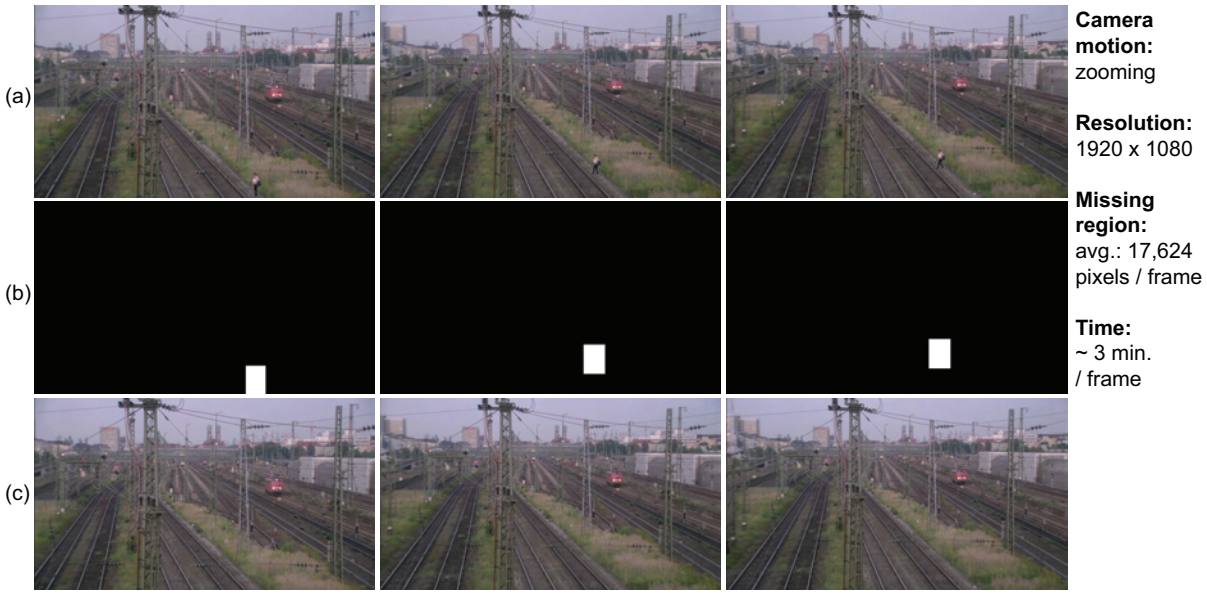


Figure 10: Results for the "Station" sequence : (a) Original frames; (b) missing regions; and (c) completed frames. Frames from `https://cs-nsl-wiki.cs.surrey.sfu.ca/wiki/Video_Library_and_Tools`

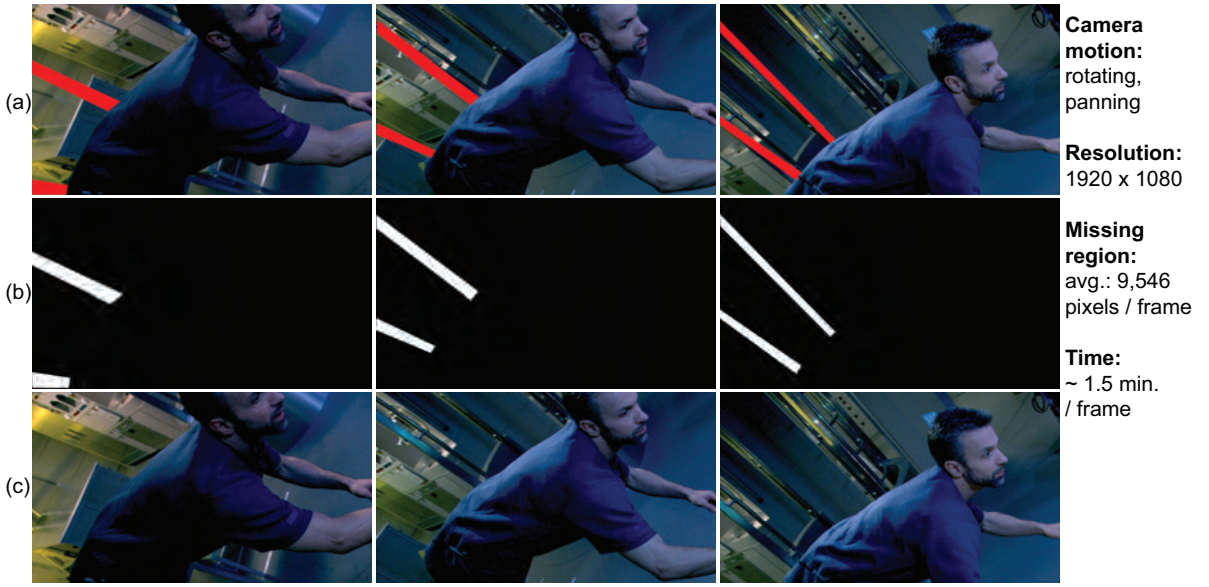


Figure 11: Results for the "Race to Mars" sequence (frames were cropped to better see the regions): (a) Original frames (with unwanted wires highlighted in red); (b) missing regions; and (c) completed frames. Frames from "Race to Mars", a courtesy of Galafilm and Discovery Channel Canada

different camera motions such as zooming, rotating, and panning. Finally, the main advantage of the proposed method over previous works is the maximum resolution it can handle. The proposed method handles HD video sequences while the highest resolution of all previous works from Section 2 is only 640×480, which is more than a six-fold improvement over state-of-the-art exemplar-based methods.

5 CONCLUSION

We have presented a video completion method that can handle much higher resolutions than previous work. The proposed method is based on three new approaches: a dual inpainting-sampling filling-order completion, a new coherence-based matches refinement, and a new localized search completion approach. Together, these three approaches solve the memory consumption and computation time problems for the completion of HD video sequences. Furthermore, the quality of the results generated by our method compares fa-

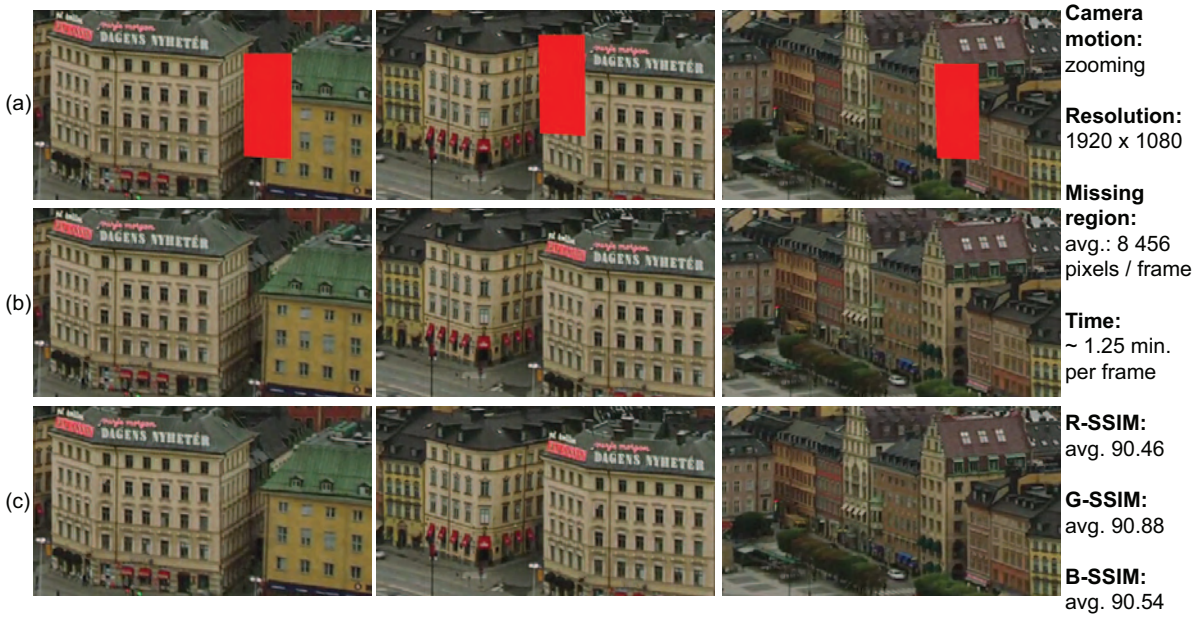


Figure 12: Results for the "Old town cross" sequence (frames were cropped to better see the regions): (a) Frames with a synthetic object; (b) completed frames; and (c) clean frames. Frames from `https://cs-nsl-wiki.cs.surrey.sfu.ca/wiki/Video_Library_and_Tools`

	Criteria			
Related works	**Missing region(s) specification**	**Exemplar-based approach**	**Camera motions**	**Maximum resolution**
Kamel et al. [7]	User provided mask	Standard	Static	80×110
Shih et al. [15]	Bounding box given by user, missing region is tracked	Improved patch-matching	Static, non-static	320×240
Liu et al. [10]	User provided mask	Motion fields and colors	Static	320×240
Xiao et al. [21]	User provided mask	Motion similarity and colors	Static	384×192
Shiratori et al. [17]	User provided mask	Motion fields	Static	352×240
Wexler et al. [20]	User provided mask	Motion similarity and colors	Static	360×288
Koochari and Soryani [8]	User provided mask	Standard	Static	540×432
Patwardhan et al. [12]	User provided mask	Motion inpainting and priority based texture synthesis	Static, non-static	640×480
Herling and Broll [5]	Rough selection by user, missing region is tracked	Combined pixel-based approach	Static, non-static	640×480
The proposed approach	User provided mask	dual inpainting-sampling filling-order completion, coherent and localized search	Static, non-static	1920×1080

Table 2: Comparison of the proposed method with previous works

vorably to previous works and allows for a significant increase of the resolutions that can be completed.

The proposed coherence-based match refinement is promising as it could be applied at various steps of several video completion approachs. Future work will involve an investigation of when the coherence approach provides the best improvements: between each iterations; between each resolution levels; at coarser or finer resolutions; etc. As they are used in the proposed method, the coherence-based matches refinement and localized search completion consider a fairly limited number of patches. Therefore, the search could stop in a local minimum while there are better matches elsewhere in the video. Future work should look at appropriate techniques to expand the search to other locations that are likely to contain good matches.

6 ACKNOWLEDGMENTS

We would like to thank the Fonds québécois de la recherche sur la nature et les technologies (FQRNT),

Mokko Studio inc., and the École de technologie supérieure for funding this project. We would also like to thank all members of the Multimedia Lab for their reviews. Finally, we would also want to thank Galafilm and Discovery Channel Canada for the "Race to Mars" video sequence.

REFERENCES

[1] Arya, S., Mount, D.M.: A library for approximate nearest neighbor searching (2010). URL `http://www.cs.umd.edu/~mount/ANN/`

[2] Ashikhmin, M.: Synthesizing natural textures. In: 2001 ACM symposium on Interactive 3D graphics, pp. 217–226. ACM (2001)

[3] Bertalmio, M., Bertozzi, A., Sapiro, G.: Navier-stokes, fluid dynamics, and image and video inpainting. In: Proc. Conf. Comp. Vision Pattern Rec., pp. 355–362 (2001)

[4] Cheung, V., Frey, B.J., Jojic, N.: Video epitomes. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 42–49 (2005)

[5] Herling, J., Broll, W.: High-quality real-time video inpaintingwith pixmix. IEEE Transactions on Visualization and Computer Graphics **20**(6), 866–879 (2014)

[6] Jia, J., Tai, Y.W., Wu, T.P., Tang, C.K.: Video repairing under variable illumination using cyclic motions. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(5), 832 –839 (2006)

[7] Kamel, S., Ebrahimnezhad, H., Ebrahimi, A.: Moving object removal in video sequence and background restoration using kalman filter. In: Int. Symp. on Telecommunications 08, pp. 580–585 (2008)

[8] Koochari, A., Soryani, M.: Exemplar-based video inpainting with large patches. Journal of Zhejiang University - Science C **11**(4), 270–277 (2010)

[9] Ling, C., Lin, C., Su, C., Liao, H.Y., Chen, Y.: Video object inpainting using posture mapping. In: Proc. IEEE Int. Conf. on Image Processing, pp. 2785–2788 (2009)

[10] Liu, M., Chen, S., Liu, J., Tang, X.: Video completion via motion guided spatial-temporal global optimization. In: Proc. ACM Multimedia, pp. 537–540 (2009)

[11] Patwardhan, K., Sapiro, G., Bertalmio, M.: Video inpainting of occluding and occluded objects. In: Proc. IEEE Int. Conf. on Image Processing, vol. 2, pp. 69–72 (2005)

[12] Patwardhan, K.A., Sapiro, G., Bertalmio, M.: Video inpainting under constrained camera motion. IEEE Transactions on Image Processing **16**(2), 545 –553 (2007)

[13] Sheikh, H., Sabir, M., Bovik, A.: A statistical evaluation of recent full reference image quality assessment algorithms. IEEE Transactions on Image Processing, **15**(11), 3440 –3451 (2006)

[14] Shen, Y., Lu, F., Cao, X., Foroosh, H.: Video completion for perspective camera under constrained motion. In: Proc. of the 18th Int. Conf. on Pattern Recognition, vol. 3, pp. 63–66 (2006)

[15] Shih, T., Tang, N., Hwang, J.N.: Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity. IEEE Transactions on Circuits and Systems for Video Technology **19**(3), 347–360 (2009)

[16] Shih, T.K., Tang, N.C., Yeh, W.S., Chen, T.J.: Video inpainting and implant via diversified temporal continuations. In: Proc. of the 14th annual ACM int. conf. on Multimedia, MULTIMEDIA '06, pp. 965–966. ACM (2006)

[17] Shiratori, T., Matsushita, Y., Tang, X., Kang, S.B.: Video completion by motion field transfer. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 411–418 (2006)

[18] Tong, X., Zhang, J., Liu, L., Wang, X., Guo, B., Shum, H.Y.: Synthesis of bidirectional texture functions on arbitrary surfaces. ACM Trans. Graph. **21**, 665–672 (2002)

[19] Wexler, Y., Shechtman, E., Irani, M.: Space-time video completion. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 120–127 (2004)

[20] Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. IEEE Trans. on Pattern Analysis and Machine Intelligence, **29**(3), 463–476 (2007)

[21] Xiao, C., Liu, S., Fu, H., Lin, C., Song, C., Huang, Z., He, F., Peng, Q.: Video completion and synthesis. Computer Anim. Virt. Worlds **19**, 341–353 (2008)

[22] Zhang, Y., Xiao, J., Shah, M.: Motion layer based object removal in videos. In: 17th IEEE Workshops on Application of Computer Vision, vol. 1, pp. 516 –521 (2005)

Evaluating Stereoscopic Visualization for Predictive Rendering.

Fernando da Graça
Center for Robotics
MINES ParisTech,
PSL-Research University
60 Boulevard Saint Michel
75006, Paris, France
fernando.graca@mines-paristech.fr

Alexis Paljic
Centre for Robotics
MINES ParisTech,
PSL-Research University
60 Boulevard Saint Michel
75006, Paris, France
alexis.paljic@mines-paristech.fr

Emmanuelle Diaz
R&D Department,
Cognitive Science and Human Factors
PSA Peugeot Citroën
2, route de Gisy
78943,
Vélizy-Villacoublay Cedex, France
emmanuelle.diaz@mpsa.com

ABSTRACT

The context of this work is predictive rendering; our objective is to previsualize materials based on physical models within computer graphics simulations. In this work we focus on paints constituted of metallic flakes within a dielectric binder. We want to validate a "virtual material workshop" approach, where a user could change the composition and the microstructure of a virtual material, visualize its predicted appearance, and be able to compare it to an actual sample. To do so, our methodology is to start from Scanning Electron Microscopy (SEM) imaging measures on an actual sample that allowed us to characterize two metrics: flake size and flake density. A statistical model based on those measures was then integrated in our spectral rendering engine using raytracing and photon mapping, with an off axis-frustum method to generate stereoscopic images for binocular visualization. Our objective is twofold: 1) perceptually validate our physical model, we evaluate if the virtual metric perceptually corresponds to the real metric of the real samples; 2) evaluate the contribution of virtual reality techniques in the visualization of materials. To do so, we designed a user study comparing photographs of car paint samples and their virtual counterpart based on a design of experiments. The observers evaluated the visual correspondence of different virtual materials generated from microstructures with varying metric values. The results show a perceptual correspondence between real and virtual metrics. This result has a strong impact: it means that for a desired appearance the proposed models correctly predict the microstructure. The second result is that stereoscopy improves the metric correspondence, and the overall appearance score.

Keywords

virtual-reality, predictive rendering, visual perception, complex materials, metallic paints, microstructure, statistical model

1 INTRODUCTION

The study of visualization quality is a crucial step to accurately represent digital prototypes. Manufacturing departments understand this problematic, since they rely on digital representations of the end product to make critical choices about its final appearance. Furthermore, it gives them the freedom to comprehensively explore (simulate) a large variety of materials without the need to manufacture the actual object. However, a correct representation of real materials still remains a challenging task. To this end, researchers focused on developing Predictive Rendering (PR) techniques to reduce de gap between the observations of a physical object and its virtual replica.

According to Wilkie et al. [1] PR, as opposed to believable rendering, is a field of research that aims at creating physically correct computer images [2, 3]. The objective is to predict the true visual appearance from a virtual reflectance model, which takes into consideration the physical parameters of the actual material. If such a tool was to be mastered, it would allow to design a virtual material and simulate its visual appearance iteratively. Then, when the desired appearance is obtained for the digital prototype, the actual equivalent object could be produced with the same set of param-

eters that were used as input for the virtual reflectance model. This process is meant to lead to better design decisions at a lower cost. Predictive rendering has a very strong potential in various application domains, ranging from manufacturing industries (automotive, aeronautics), cosmetics to architecture (material design).

On the other hand, complex materials such as automotive paints are very challenging to simulate, due to their spatially varying reflectance. To this end, we argue that VR tools are indispensable to fully explore the visual aspect of such materials. Virtual Reality is used in the industry for product design or industrial process validations. Through the use of stereoscopic displays, user motion capture, motion parallax, VR allows for a user to feel immersed in a 1:1 scale virtual environment, and to observe objects and interact with them. In particular, binocular vision and motion parallax are necessary to provide the human visual system with valuable depth and shape cues. These are key characteristics of VR that are necessary to ensure "human in the loop" simulations. In order to combine the physical and visual validity of Predictive Rendering and perception cues provided by Virtual Reality, the need for stereoscopy and motion parallax for predictive rendering simulations is being expressed by the industry.

Predictive rendering approaches require perceptual validations that take into account the human visual system in order to be valid. The field of research for such validations is large and we are only starting to draw the boundaries within which virtual material samples are representative of real material samples. Indeed, image quality perception can depend on a lot of parameters that appear at several stages of the process. One should, at least, consider the following: the human user (visual acuity, individual color perception, visual fatigue), the technical setup (display calibration, display resolution, luminance), the sensory motor inputs and outputs (use of stereoscopy, motion parallax, user's ability to manipulate the virtual material sample), and the rendering engine itself (light-matter interaction models, material models). In this context, this work is part of an iterative validation process in a research project of a predictive rendering engine in which the objective is to link microstructure and appearance. In this work, our objective is twofold: 1) evaluate the pertinence of a "virtual material workshop" approach where a user could change the composition and the microstructure of a virtual material, visualize its predicted appearance, and be able to compare it to an actual sample. To do so, we propose to use a microstructure model [4] based on measures of actual material samples presented in section 2.2; 2) evaluate the role of stereoscopy on perception of materials that depict binocular differences such as automotive paints with metallic flakes. For this purpose, our methodology evaluates, through a user study, the visual agreement between the observation of the computer generated object and the actual object. We introduce a novel approach in Computer Graphics (CG) domain to render virtual materials by using the microstructure formulation of the real material, see figure 2.

We begin the paper with a survey of related work in section 1.1. In section 2 we describe the simulation of virtual automotive paints using a microstructure model. Then, in section 3 we describe the experimental setup of the virtual scene, and section 4 the design of experiment to evaluate the response of the observers. In the section 5 we present the results of the measured data. The experimental results are analyzed in the section 6. Finally, in section 7 we present our conclusions, and we address some aspects for future work to complement our findings.

1.1 Related Work

In this section, we first propose an overview of the existing CG (Computer Graphics) methods for computer generated images of materials with nano/macro inclusions such as car paints with flakes. We then explore the existing literature on the role of stereoscopy on the perception of surface aspect.

In the CG domain, several researches have proposed different methods to simulate car paint models. These models are based on Bidirectional Reflectance Distribution Function [5], which represents how the surface reflects the incident light at different angles. The distribution of the reflectance of the light can be captured by optical measurement devices [6]. Then, the obtained data is used to derive reflectance models that represent the appearance of the physical material [7]. In addition, we can also consider the Bidirectional surface scattering distribution function (BSSRDF) models [8] that takes into account the scattering of the incident beam of light in the interior of the material.

Generally speaking, we can distinguish two groups: analytical, and data-driven models. In the first, the user tweaks several parameters until he achieves a visual aspect that is similar to the real paint. Durikovic et al. [9] model the geometry of the flakes inside the paint film. Their system is capable of generating stereoscopic images, and it allows to define the parameters for the random distribution of the position of flakes and their orientation. The approach of Ershov et al. [10] is based on reverse engineering, the appearance attributes such gloss are added to the physical model by adjusting parameters. The inconvenient of these models is the amount of parameters that are necessary to represent the car paint appearance.

In the second approach, Günther et al. [11] developed an image based acquisition setup to measure the Bidirectional Reflectance Distribution Function (BRDF) of

a car paint. To this BRDF they add the sparkle simulation. The distribution of the the flakes is stored as a texture. Rump et al. [12] used a similar approach as the previous one. The difference is that, instead of using a sparkle simulation with the BRDF, they capture the spatially varying appearance (sparkle effect) using Bidirectional Texture Function (BTF) measurements. In addition, they store and simulate directional visual effects. Sung et al. [13] determine individual flake orientations in the car paint by using confocal laser scanning microscope. They capture the angular dependent reflectance using a goniometer. Then, they use these measurements to build a reflectance model. In the context of this work, we need to evaluate the perception limits in a VR environment, when the observer perceives the aspect of the reflectance models using binocular vision. A human observer perceives the binocular summation of the left and right image. In the combination of the two images, we can identify two cases binocular fusion and rivalry. In the first case, if two retinal points are sufficiently similar, a binocular combination occurs. In contrast, when two retinal points are very distinct, the observer perceives a fluctuation between the left and right images, that is, a failed fusion occurs. This phenomenon is known as binocular rivalry [14].

Most of the work was done on the role of stereoscopy for the perception of gloss. In the literature the gloss is defined to be a global property of the surface aspect [15, 16]. Glossy surfaces reflect the incident light to a particular outgoing direction, characterized by a specific angular interval. The perception of specular reflections is an important cue to evaluate the glossiness of the materials. In addition, there is an influence of binocular cues such as highlight disparities on the perception of gloss [17]. The experimental results of Wendt et al. [18] show an improvement of gloss perception when highlights cues disparities are taken into account. The experimental results of the work of G. Obein et al. [19] suggest also that the binocular vision helps for judgment of high gloss samples. They found that with binocular vision the sensitivity to gloss is higher than the monocular vision, for high gloss levels. Sakano et al. [20] examined the effects of the combination of self-motion and contingent retinal-image motion (motion parallax) on perceived glossiness. When the observer was able to move his head, a stronger glossiness was perceived than when both the observer and the stimulus were static. From their experimental results, they found that the glossiness under the monoscopic condition was underestimated compared to stereoscopic condition. The glossiness under static (head not moving) condition was underestimated compared to dynamic condition. Knill et al. [21] study the combination of different cues for slant perception. At low slants, observers use more the binocular cues than the texture. At slants of 50 and 70, the subjects do better slant judgments using the texture information of the image. As the slant increases the observers give more attention to texture information.

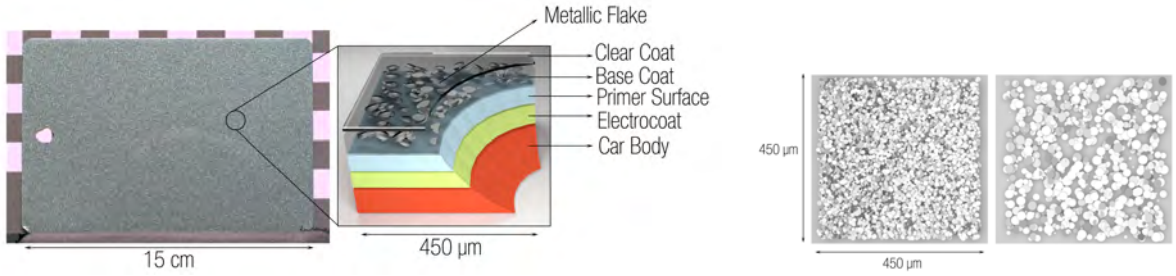
2 MATERIAL SIMULATION

2.1 Physical Plate

This study focus on grey automotive paints with metallic flakes up to 30μm. Figure1(a) shows a cross section of the studied actual sample of a car paint. Typically an automotive paint is made of four layers: clear coat, base coat, primer surface, and electrocoat. The metallic flakes are made of aluminum, and are distributed with different orientations on the primer surface of the plate at different depths, and the clear coat is transparent. The amount, distribution, and the orientation of the flakes are controlled by a milling machine. Due to their size, and orientation, these metallic pigments convey distinctive visual appearances to the object such as sparkle, and directional visual effects. At the macroscopic scale of visualization, the appearance of these nonuniform paints depend on the lighting conditions (directional/diffuse), distance and angle of observation, orientation, diameter, and density of the flakes [22, 23, 24]. The nanoscopic effects such as the chemical composition, rugosity and clustering effects of the flakes can also influence the visual appearance. In this work we only consider the macroscopic effects for which we use geometric optics models.

2.2 Stack Model

As for the models of the microstructure, we can distinguish two cases: a quasistatic, where the particles are smaller than the wavelength, consequently the human eye cannot perceive each particle. We can obtain the macroscopic visual aspect of the microstructure, which is visible by the human eye, by using homogenization methods on agglomerations of nanoscopic particles. In this work, we are interested in the second case, geometric optics, where the particle size is larger than electromagnetic visible wavelength, which is suitable to be directly used in our render engine. In this case the microstructure simulates the distribution of microscopic particles, such as metallic flakes, according to the analyses of the actual object. Morphology and statistical analysis of Scanning Electron Microscopy (SEM) images of real objects is used to create virtual microstructures that geometrically corresponds to the one of the real material. These microstructures are then used to simulate the optical behavior between the different particles. Secondly, it allows the user to modify the microstructure in order to tune the visual appearance keeping the physical feasibility, and therefore the manufacture of the virtual material, see figure 2. The stack model [4] used in our render engine distributes the metallic flakes on the surface. Though the complex



(a) (b)

Figure 1: On the left of the image (a), an example of a real car paint plate with a diagram of a typical coating paint. The flakes have typically the shape of a disk with a diameter of 5μm to 50μm, in the case of a car paint. The surface can contain rugosity and an interference film. The flakes are imbedded in the base coat with different orientations and at different depths. The figure (b) depicts two microstructures generated by the stack model. On the left a microstructure with metallic flakes of size 15μm, and on the right with metallic flakes of size 30μm.

geometry of a real metallic flake, the virtual metallic flakes were modelized with a flat cylinder shapes whose height and radius is parameterized, see figure 1(b). The statistical variation of dispersion, size, orientation is measured on the SEM images of the real plate using different morphological analyses. Then, the model simulates the clusters of flakes by using a 2D Poisson point process. Due to the large number of flakes on the surface. The stack model generates continuous microstructures of size $450\mu\text{m} \times 450\mu\text{m} \times 20.24\mu\text{m}$. The generation of plates with different flake densities and radius sizes is controlled by two parameters of the stack model. For each microstructure we have information about the radius size, the orientation, position, and amount of flakes. To generate a virtual plate with $15\text{cm} \times 10\text{cm}$, a set of 332×240 microstructures is necessary to fill the surface of the virtual plate. To avoid the large amount of geometry in our simulations, we could consider the usage of a set of textures encoding different types of information about the flakes, for instance the normal vector, and depth.

2.3 Rendering of the Virtual Model

For the optical simulation of the virtual object we used our spectral render engine raytracing with photon mapping. The scene was rendered within the visible spectrum interval $[380\text{nm} - 780\text{nm}]$ with a wavelength step of 5nm. This interval corresponds to the range of wavelengths that the human eye is sensitive. Then, we used a virtual scene with the same light conditions that were used in the real room. We associate for each plate the optical constants n and k of aluminium. In the virtual scene we used the spectrum of SOLUX $4700K$ light. The index of absorption was found in a prior experience with paired comparison. The observers were asked to evaluate which of the two virtual images was the closest to the photograph. The found absorption coefficient is $k = 6$. Furthermore, we noticed that the thickness and the absorption coefficient have an important role in the brightness of the metallic flakes. We used a virtual plate with three layers. The base layer is a black surface to minimize the back-surface reflection. The second layer, the binder, contains the microstructure of the metallic flakes, which was generated by the stack model. Finally, the clear coat layer imparts a glossy appearance to the plate. The virtual samples were defined thanks to a design of experiments. We have two factors the flake size and the flake density. A surface response design was chosen in order to evaluate the influence of the main effects but also the potential non linear effects and the interaction between factors. 13 virtual samples were built following a central composite design with replicated centered points.

3 SETUP

In this section we describe the configuration of the virtual scene. Figure 3(a) represents the virtual scene used in the experiments. We used a directional isotropic light, which is emitted from the top. The virtual cameras are placed at a distance of 77cm from the plate. The stereo cameras have an interocular distance of 6.5cm and the cyclopean camera is placed in the middle of the left and right cameras. Figure 3(b) shows the distance of observation, and the display area to visualize the virtual plates. The field of view, fov, of the camera was calculated using the observation distance, a, and the width of the area of projection, b, $fov = 2 \times \arctan\left({}^{2a}/_{b}\right)$.

4 USER STUDY

We designed a user study with 26 subjects using experimental design theory to optimize the number of trials run in order to obtain valid results using a minimum parameters variation of flake density and flake radius, see figure 5 (c). During the experiment, the subjects sat in a dark room, facing a stereoscopic screen. They observed two series of plates (stereoscopic/monoscopic),

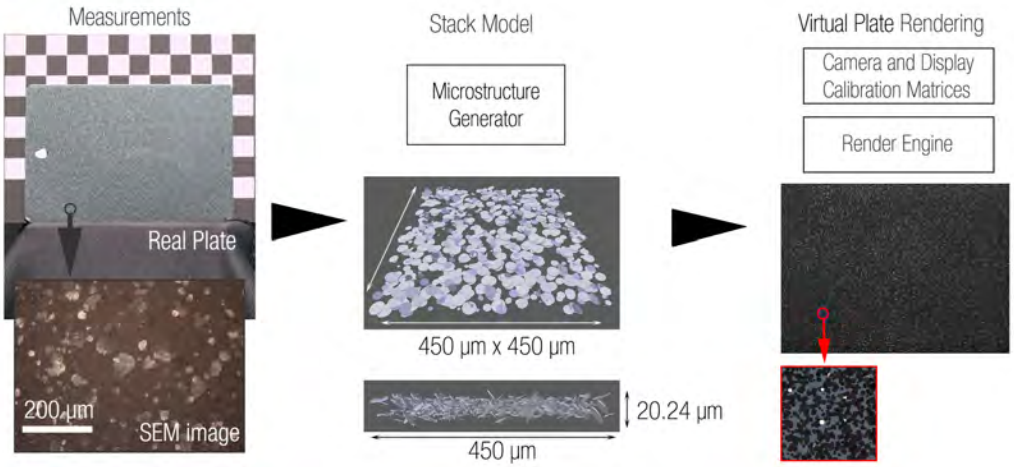
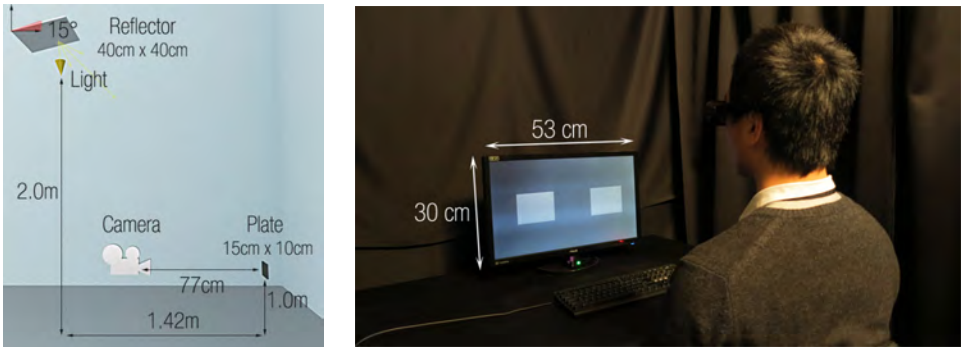


Figure 2: Through morphological and statistical analysis of SEM images, a set of microstructures are generated in order to fill the surface of the virtual plate. Then, the render engine computes the appearance of the virtual plate using the information of the microstructures, and the colorimetric calibration matrices of the photographic camera and the screen display used during the experiment.



(a) (b)

Figure 3: The image (a) is a schematic diagram used for the virtual scene. (b)The physical observation conditions to visualize the virtual plates.

Factor	Low Level	High Level	Physical Value
Flake Density	3.5	9.5	1.0
Metallic Flake Radius	15μm	45μm	30μm

Table 1: Based on the morphological and statistical analysis of the actual automotive paint, the flake radius is 30μm, and flake density is 1.0 (this value is unitless). The minimum and maximum range for flake density, and flake radius were found empirically. Within each interval, 13 values were chosen through Central Composite Design.

each one with different flake density and flake radius, see table 1. Subjects evaluated the similarity of the virtual plate to the photograph of the actual plate by using a scale from 0 (equal to the reference plate) to 10 (different from the reference plate). After an answer was given, the subject changed to the next plate with different radius and density of metallic flakes. For each trial, the presentation order of the stimuli follows a Williams Latin Square Design. The main experiment was preceded by a practice trial of two stimuli to gain familiarity with the experience.

The subjects used stereoscopic active shutter glasses during both monoscopic and stereoscopic conditions. For each trial, the subjects observed two sequences of 13 stereoscopic images. The first sequence corresponds to the monoscopic-condition, while the second corresponds to the stereoscopic case. In all experiments, the subjects used stereoscopic active shutter glasses NVIDIA 3D Vision 2 during both monoscopic and stereoscopic conditions. For the monoscopic conditions the same image was displayed for both eyes. The images were displayed using an active stereo ASUS VG248QE display. The stimulus was displayed with

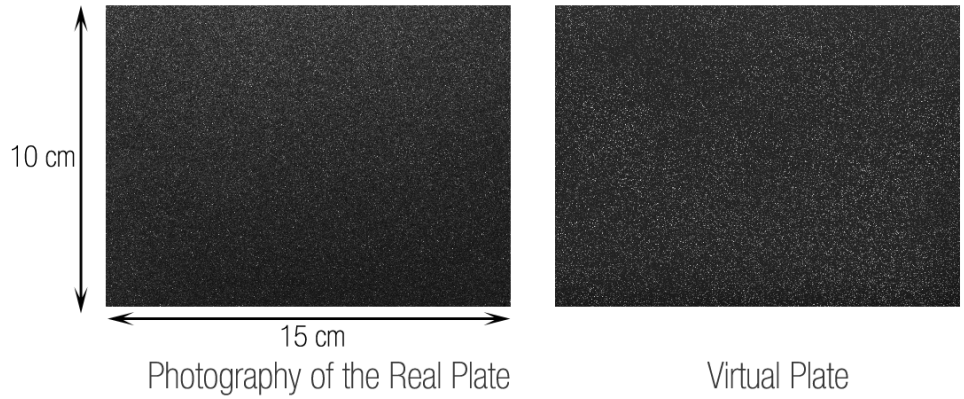


Figure 4: The stimulus displayed to the subjects. The "Reference" material (left plate) is static through the experiment, while the material of the plate "Sample" (right plate) changes according to 13 different values of flake densities, and flake radius size.

a resolution of 1920×1080 pixels at 72Hz. The stimulus consisted of two plates designated as reference, and variable plate, see figure 4. The reference is the photography of the actual automotive paint plate, which was taken with a camera *NikonD*800. The initial high dynamic range (HDR) image was made by taking 6 images with film sensitivity ISO 100, and the following shutter speeds: 1.3, 1.6, 2.0, 2.5, 3.0, and 4.0 seconds. Then, the HDR image image was converted to the RGB colour space using the calibration matrices of the photographic camera, and the screen display. The plates dimensions are $10\text{cm} \times 15\text{cm}$. The samples are placed inside of a dark room with one source of light SOLUX $4700K$ with known spectrum. The virtual stereo and cyclopean cameras are located at 77cm from the plate, which is the same distance of observation that was used to take the photography of the real plate.

5 RESULTS

5.1 Data Exploration

From the box plots, we noticed that there are some extreme outliers and extreme values. This is due to the fact that there is an important agreement within the population to give the same evaluation to a given plate.This is translated into a positive Kurtosis. A principal component analysis (PCA) was applied in order to verify if the observers generally agree in their evaluations. The first axis represents more than 80% in the two studies that is to say that there is an extraordinary agreement between the subjects. Thanks to these preliminary analyses, the average of the panel represents well the raw data so this indicator can be used in Data Modeling.

5.2 Data Modeling

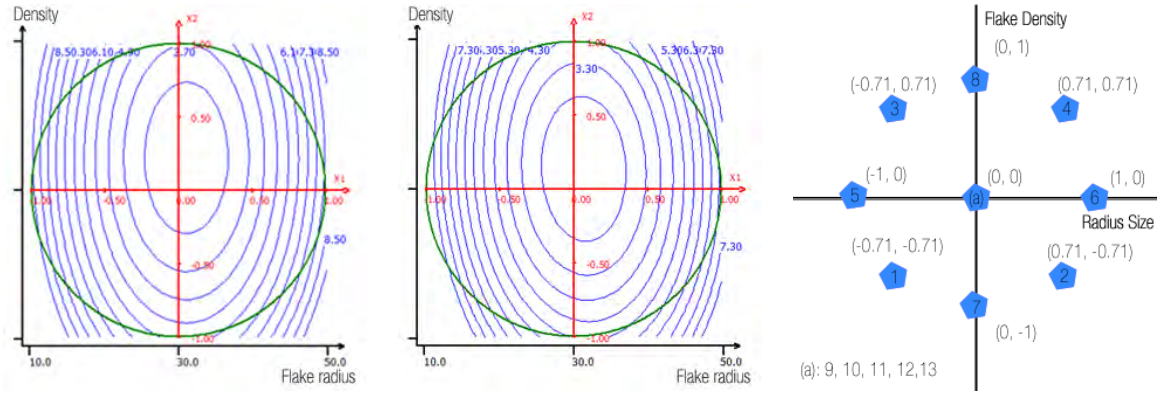
The experimental design allows to evaluate the interaction between the two parameters, non linear effects, and to find the optimum values for each parameter. As we said in the previous section, we can consider that the average of response is representative of the population. Therefore, we can model the response according to the parameters of the experimental design: linear and non linear effects of density and radius size, and their interaction. The surface of response depicted in figures 5 (a) and (b) have a bell shape surface. Hence, there is no interaction between the two factors: flake density and flake radius size. For the stereoscopic case we obtain a better R2 pred (based on cross validation) and a smaller PRESS (Predicted Residual Sum of Squares), see table 3, therefore the stereoscopic model is more robust. The average degree of proximity to the photography of the actual sample is 5.1, and 4.6, for monoscopic and stereoscopic. The analysis of the quadratic effects show that for low or high density of flakes, the virtual plate is not considered similar to the real plate. The same result was found for small and large flakes radius. Finally, the Least Square Difference Test (LSD) show that with the stereoscopic condition the observers were able to discriminate better the plates.

6 DISCUSSION

There is a difference between the the monoscopic and stereoscopic observations of the plate. In average, the stereoscopic images of the virtual plate were better evaluated. The ANOVA repeated measure analysis on the global response indicates that the stereoscopic visualization of the virtual plate is closer to the actual plate. In other words, there is a better visual agreement to the photography of the physical plate with stereoscopic vision. Furthermore, the second ANOVA analysis on stereoscopic and monoscopic condition, confirms that the stereoscopic vision allows a better differentiation of the virtual plates. With monoscopic vision it is more perceptible the individual white reflections of the metallic flakes, and also the visual patterns resulted from the clustering of metallic flakes. While with the

	Coefficient		Signif. %	
	Monoscopic	Stereoscopic	Monoscopic	Stereoscopic
$b0$	2.723	2.423	$< 0.01^{***}$	$< 0.01^{***}$
Radius Size $b1$	-0.590	-0.730	$< 0.0927^{***}$	$< 0.0857^{***}$
Radius Size $b1-1$	6.003	5.323	$< 0.001^{***}$	$< 0.001^{***}$
Flake Density $b2$	-0.794	-0.516	$< 0.0294^{***}$	0.318^{**}
Flake Density $b2-2$	1.810	1.822	$< 0.001^{***}$	0.0123^{***}

Table 2: The table of coefficients.



(a) (b) (c)

Figure 5: The graphs (a) and (b) depict the response surface of the experiment. (c) is the composite design used in the experiment that shows how many different plates to use and the variation of density and size of the metallic flakes. The values depicted in the graph of surface of responses are reduced and centered to the interval -1 to 1 so the influence of the factors can be compared. In this way, we can represent the domain of each factor in the same reference.

	Monoscopic	Stereoscopic
R2	0.961	0.976
R2 pred	0.726	0.838
PRESS	19.494	9.116

Table 3: The adequacy measures for the model.

stereoscopic vision it is more difficult to identify these patterns effects due to the binocular rivalry. Instead we can observe a glittering effect, which according to the comments of the observers, make the virtual plates to look more realistic.

In our experiments, we found that the radius of the flake has a great impact in the judgment of similarity. In figures 5 (a) and (b), the ellipsoidal shape of the isometric curve is oriented in the $x2$ axis. This particular orientation shows that the quadratic effects are stronger in the $x1$ axis, i.e., the flakes radius size axis. This dissymmetry is an indicator that the similarity evaluation note is more sensible to the radius size than to the flake density factor. In addition, according to the table of coefficients 2, the flake radius, $b1-1$, is three times more important than the flake density, $b2-2$. In the stereoscopic case, the shape of the isometric curve is tilted to the right. However, the radius size factor remains more sensible or influent than the flake density parameter.

From the obtained results, there are not particular tuples of factors that makes the subject to judge that the virtual plate is similar to the photography, for instance a tuple with a large flake radius, and a strong flake density. For the monoscopic condition, the optimum values calculated from the surface response, show that in order the virtual plate to be considered similar to the photography, the plate must contain flakes with smaller radius size and to have a higher flake density. While in the stereoscopic condition, the optimum values show the inverse, higher flake radius size with lower flake density.

7 CONCLUSIONS

In this work, we evaluated the pertinence of a "virtual material workshop" approach, and the role of stereoscopy on perception of materials that depict binocular differences such as automotive paints with metallic flakes. For this purpose, we developed a user study based on a design of experiments, to evaluate the visual agreement between the observation of a computer generated object and the actual object. The results show that there is a match between the real and virtual metrics. This means that for a desired appearance our methodology can predict the microstructure. Secondly, the stereoscopic vision improves the visual representation of the virtual plates with metallic flakes. Finally, the size of flake radius has a great influence in the judgment of the observers.

We are currently working on the visualisation of virtual materials with a High Dynamic Range display, to study the influence of the high dynamic luminance on the perception of materials such as car paints. The next step for evaluating VR is to assess the pertinence of using head tracking to generate correct dynamic perspectives.

8 ACKNOWLEDGMENTS

This work is part of the LIMA project (Light Interaction Material Aspect), which is a collaborative research project between academic researchers and industrial manufacturers. The project is funded by the French National Research Agency, ANR, with the Grant Number ANR-11-RMNP-0014. The authors wish to thank F. Willot, E. Couka, D. Jeulin, and P. Callet for the microstructure models, A. Thorel, A. Chesnaud, and M. Ben Achour for the SEM measures, P. Porral for providing PSA Peugeot Citroën paint samples, and the persons who participated in the user study.

9 REFERENCES

[1] Wilkie, A., Weidlich, A., Magnor, M., and Chalmers, A., "Predictive rendering," in [*ACM SIGGRAPH ASIA 2009 Courses*], *SIGGRAPH ASIA '09*, 12:1–12:428, ACM, New York, NY, USA (2009).

[2] Ulbricht, C., Wilkie, A., and Purgathofer, W., "Verification of physically based rendering algorithms," *Computer Graphics Forum* **25**(2), 237–255 (2006).

[3] Rushmeier, H., [*Computer Graphics Techniques for Capturing and Rendering the Appearance of Aging Materials*], 283–292, Springer US (2009).

[4] Enguerrand Couka, François Willot, D. J., "A mixed boolean and deposit model for modeling of metal flakes in paint layers," *Image Anal Stereol* **32**(2), 8–13 (2012).

[5] Nicodemus, F. E., Richmond, J. C., Hsia, J. J., Ginsberg, I. W., and Limperis, T., "Radiometry," ch. Geometrical Considerations and Nomenclature for Reflectance, 94–145, Jones and Bartlett Publishers, Inc., USA (1992).

[6] Boher, P., Leroux, T., and Bignon, T. *SID Conference Record of the International Display Research Conference* .

[7] Kook Seo, M., Yeon Kim, K., Bong Kim, D., and Lee, K. H., "Efficient representation of bidirectional reflectance distribution functions for metallic paints considering manufacturing parameters," *Optical Engineering* **50**(1), 013603–013603–12 (2011).

[8] Jensen, H. W., Marschner, S. R., Levoy, M., and Hanrahan, P., "A practical model for subsurface light transport," in [*Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*], *SIGGRAPH '01*, 511–518, ACM, New York, NY, USA (2001).

[9] Ďurikovič, R. and Martens, W. L., "Simulation of sparkling and depth effect in paints," in [*Proceedings of the 19th spring conference on Computer graphics*], *SCCG '03*, 193–198, ACM, New York, NY, USA (2003).

[10] Ershov, S., Ďurikovič;, R., Kolchin, K., and Myszkowski, K., "Reverse engineering approach to appearance-based design of metallic and pearlescent paints," *Vis. Comput.* **20**, 586–600 (Nov. 2004).

[11] Günther, J., Chen, T., Goesele, M., Wald, I., and Seidel, H.-P., "Efficient acquisition and realistic rendering of car paint," in [*Vision, Modelling, and Visualization 2005 (VMV), Proceedings, November 16-18, Erlangen, Germany*], Greiner, G., Hornegger, J., Niemann, H., and Stamminger, M., eds., 487–494, Akademische Verlagsgesellschaft Aka GmbH, Berlin (November 2005).

[12] Rump, M., Müller, G., Sarlette, R., Koch, D., and Klein, R., "Photo-realistic rendering of metallic car paint from image-based measurements," *Computer Graphics Forum* **27**(2), 527–536 (2008).

[13] Sung, L., Nadal, M. E., McKnight, M. E., Marx, E., and Dutruc, R., "Effect of aluminum flake orientation on coating appearance.," 1–15 (2001).

[14] LEVELT, W. J. M., "Binocular brightness averaging and contour information," *British Journal of Psychology* **56**(1), 1–13 (1965).

[15] Ged, G., Obein, G., Silvestri, Z., Le Rohellec, J., and ViÃ©not, F., "Recognizing real materials from their glossy appearance," *Journal of Vision* **10**(9) (2010).

[16] Obein, G., Knoblauch, K., and Viénot, F., "A framework for the measurement of visual appearance.," *Commission Internationale de l'Eclairage (CIE)* **4**, 711–720 (Aug. 2006).

[17] Templin, K., Didyk, P., Ritschel, T., Myszkowski, K., and Seidel, H.-P., "Highlight microdisparity for improved gloss depiction," *ACM Trans. Graph.* **31**, 92:1–92:5 (July 2012).

[18] Wendt, G., Faul, F., and Mausfeld, R., "Highlight disparity contributes to the authenticity and strength of perceived glossiness," *Journal of Vision* **8**(1) (2008).

[19] Obein, G., Knoblauch, K., and Viénot, F., "Difference scaling of gloss: nonlinearity, binocularity, and constancy.," *Journal of vision* **4**, 711–720 (Aug. 2004).

[20] Sakano, Y. and Ando, H., "Effects of head mo-

tion and stereo viewing on perceived glossiness," *Journal of Vision* **10**(9), 1–14 (2010).

[21] David C. Knill, J. A. S., "Do humans optimally integrate stereo and texture information for judgments of surface slant?," (2003).

[22] McCamy, C. S., "Observation and measurement of the appearance of metallic materials. part i. macro appearance," *Color Research and Application* **21**(4), 292–304 (1996).

[23] McCamy, C. S., "Observation and measurement of the appearance of metallic materials. part ii. micro appearance," *Color Research and Application* **23**(6), 362–373 (1998).

[24] Fettis, G., [*Automotive Paints and Coatings*], Wiley (2008).

Adaptive Importance Caching for Many-Light Rendering

Hiroshi Yoshida
Wakayama University
Wakayama, Japan
s141064@center.wakayama-u.ac.jp

Kosuke Nabata
Wakayama University
Wakayama, Japan
s141044@center.wakayama-u.ac.jp

Kei Iwasaki
Wakayama University,
UEI Research
Wakayama, Japan
iwasaki@sys.wakayama-u.ac.jp

Yoshinori Dobashi
Hokkaido University,
JST CREST
Sapporo, Japan
doba@ime.ist.hokudai.ac.jp

Tomoyuki Nishita
UEI Research, Hiroshima
Shudo University
Hiroshima, Japan
nishita@shudo-u.ac.jp

ABSTRACT

Importance sampling of virtual point lights (VPLs) is an efficient method for computing global illumination. The key to importance sampling is to construct the probability function, which is used to sample the VPLs, such that it is proportional to the distribution of contributions from all the VPLs. Importance caching records the contributions of all the VPLs at sparsely distributed cache points on the surfaces and the probability function is calculated by interpolating the cached data. Importance caching, however, distributes cache points randomly, which makes it difficult to obtain probability functions proportional to the contributions of VPLs where the variation in the VPL contribution at nearby cache points is large. This paper proposes an adaptive cache insertion method for VPL sampling. Our method exploits the spatial and directional correlations of shading points and surface normals to enhance the proportionality. The method detects cache points that have large variations in their contribution from VPLs and inserts additional cache points with a small overhead. In equal-time comparisons including cache point generation and rendering, we demonstrate that the images rendered with our method are less noisy compared to importance caching.

Keywords

Global Illumination, Many-Light Rendering, Importance Sampling, Importance Caching

1 INTRODUCTION

Photorealistic rendering has, for many years, been an interesting and challenging topic in the field of computer graphics. It has been widely used in many applications such as movies, games, architectural design, and so on. Indirect illumination plays an important role in enhancing realism. However, efficient rendering with indirect illumination is still a challenging problem due to the high computational cost.

To compute indirect illumination efficiently, Keller introduced an instant radiosity, which approximates the indirect illumination with virtual point lights (VPLs) [KELLER97]. Many-light rendering [DACHSBACHER14], which extends the instant radiosity, has been extensively researched. Many-light rendering approximates both the direct and indirect illumination incident onto each point to be shaded (referred to as *shading points*) with VPLs. Increasing the number of VPLs increases the accuracy of many-light rendering, but at the cost of computational time.

To handle a large number of VPLs efficiently, importance sampling methods [WANG09, GEORGIEV12, WU13] for VPLs that estimate the outgoing radiance of shading points have been proposed. The key component for the importance sampling method is to construct a probability function that is as proportional as possible to the distribution of contributions from all the VPLs. However, constructing a probability function perfectly proportional to the distribution at each shading point is computationally expensive since it requires a large number of visibility tests between the shading point and all VPLs. Importance caching [GEORGIEV12] constructs a probability function by sparsely distributing the cache points on the surfaces of the scene, and recording the contributions of all VPLs. The probability function at each shading point is calculated by in-

terpolating those at nearby cache points. This method, however, distributes cache points randomly and does not consider the variation in contributions from each VPL. This makes it difficult to construct a probability function that is proportional to the distribution of the VPL contributions, leading to an increase in variance.

To address this problem, we propose an adaptive cache insertion method for a many-light rendering framework. Our method detects regions where the distribution of the VPL contributions varies drastically due to the spatial variations of the shading points, the directional variations of the normals to the shading points, and due to the occlusions between the VPLs and the shading points. Additional cache points are inserted into such regions. In addition, while importance caching calculates the interpolated probability function by simple averaging, our method takes into account the spatial correlation between the shading points and the cache points, and the directional correlation between the normals to the shading points and cache points, and uses these to weight the interpolation. Our results demonstrate that, in an equal time comparison, our method provides better performance (i.e. less variance) than importance caching.

2 PREVIOUS WORK

Many-light rendering, which is based on the instant radiosity as proposed by Keller [KELLER97], distributes a large number of VPLs in the scene and approximates the incident radiance from the direct and indirect illumination from the VPLs. Increasing the number of VPLs increases the rendering accuracy at the cost of computational time. Since several thousand VPLs are required to obtain plausible results, several methods have been proposed that can handle many VPLs efficiently.

To handle a large number of VPLs efficiently, several methods that cluster VPLs have been proposed. Walter et al. proposed a hierarchical representation of the VPLs called Lightcuts [WALTER05]. Hasan et al. proposed the matrix row-column sampling (MRCS) method that samples a small number of VPLs that give a good approximation to the contributions from all the VPLs [HASAN07]. Ou et al. proposed the Lightslice method, which extends the MRCS method by clustering the shading points and applying the MRCS method to each cluster to improve the accuracy [OU11]. Although these clustering methods can efficiently render realistic images by approximating the contributions of the VPLs in each cluster by a representative VPL from each cluster, the rendered images suffer from errors due to VPL clustering.

Importance sampling methods for VPLs have also been proposed. The contribution of a VPL is the product of the incident radiance, the bidirectional reflectance distribution function (BRDF), a geometry term, and the visibility function. By constructing probability functions proportional to the contribution from VPLs and sampling the VPL according to this probability function, the outgoing radiance can be estimated with high accuracy and small variance. Wang and Akerlund proposed a bidirectional importance sampling method for many-light rendering [WANG09]. This method, however, does not take into account the visibility function, resulting in high variance where the incident light is occluded. Wu et al. proposed the VisibilityCluster algorithm, which clusters shading points and the VPLs [WU13]. The visibility function is approximated by the average values of the estimated visibilities between the clusters of shading points and the VPLs. Although this method can render realistic images efficiently, it can fail to sample VPLs with large contributions since estimates of the average values of the visibilities are done by random sampling.

Cache-based methods that exploit correlation to increase the rendering efficiency have been proposed. Ward et al. proposed irradiance caching, which accelerates the indirect illumination calculation by interpolating the incident illumination stored at cache points [WARD88]. Radiance caching methods [KRIVANEK05, KRIVANEK06] store radiance instead of irradiance to efficiently render glossy materials. Visibility caching stores visibility information to accelerate the direct illumination computation [CLARBERG08] . The work that is most relevant to our method is importance caching [GEORGIEV12]. This method randomly distributes cache points, called importance records, in the scene and records the contributions of all the VPLs as shown in Fig. 1(a). Then a probability function perfectly proportional to the contributions of the VPLs at each cache points is calculated. At each shading point, the probability function is calculated by interpolating the contributions stored at nearby cache points, and a small number of VPLs are sampled to estimate the outgoing radiance. Although it can render plausible images efficiently, this method has several drawbacks. Firstly, cache points are distributed randomly. If the VPL contributions stored at the cache points vary drastically, the interpolated probability function may not be proportional to the contributions of the VPLs. Secondly, the interpolated probability function is simply an average of those recorded at nearby cache points, which does not account for the correlation between the shading and cache points. To address this problem, we propose an adaptive cache insertion method for many-light rendering. Our method distributes cache points taking into account variations in the VPL contributions. In addition, our method interpolates the probability function by weighted averages of the probability functions stored at nearby cache points taking into account the correlations between the shading and cache points.

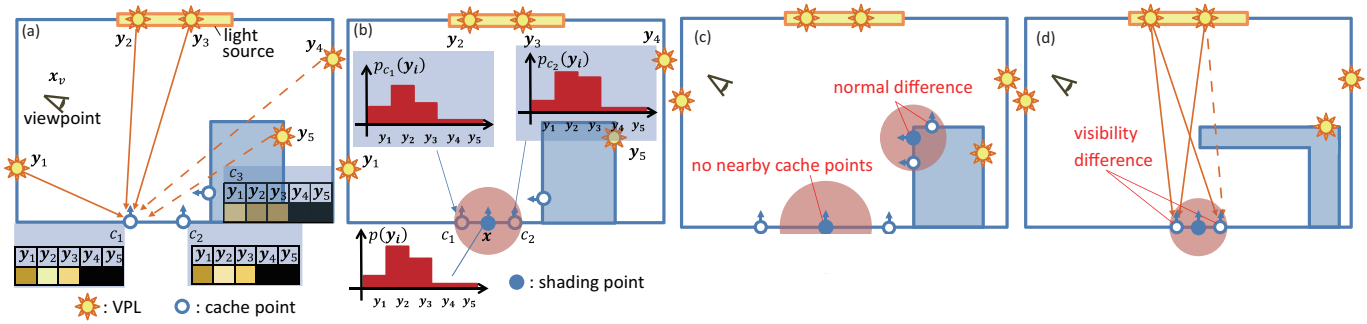


Figure 1: Importance caching (a) records contributions of all VPLs at cache points c_j. Tables under cache points represent VPL contributions. (b) Probability function p at shading point is calculated by averaging those at nearby cache points. Each graph shows the probability function, and the probability functions at cache points are calculated by normalizing VPL contributions. (c) Inefficient cases of importance caching, lack of nearby cache points and lack of cache points with similar normals. (d) Contributions of VPLs at nearby cache points differ due to occlusions.

3 IMPORTANCE CACHING

Importance caching [GEORGIEV12] samples VPLs based on a probability function calculated by interpolation between those stored at cache points. The outgoing radiance $L_o(\mathbf{x},\mathbf{x}_v)$ at shading point $\mathbf{x}$ towards the viewpoint $\mathbf{x}_v$ is estimated by the following equation:

$$L_o(\mathbf{x},\mathbf{x}_v) = \frac{1}{N}\sum_{n=1}^{N}\frac{L(\mathbf{y}_n,\mathbf{x})f_r(\mathbf{y}_n,\mathbf{x},\mathbf{x}_v)G(\mathbf{x},\mathbf{y}_n)V(\mathbf{x},\mathbf{y}_n)}{p(\mathbf{y}_n)}, \quad (1)$$

where N is the number of sampled VPLs, $\mathbf{y}_n$ is the n-th VPL, and L, f_r, G, and V are the radiance, BRDF, the geometry, and the visibility terms, respectively (please refer to the many-light rendering survey paper [DACHSBACHER14] for more details). The contribution of the VPL is the product of L, f_r, G, and V. The probability function p for sampling the VPLs is expected to be proportional to the distributions of the VPL contributions. However, constructing a probability function perfectly proportional to the distribution of the VPL contributions is computationally expensive since it requires evaluation of all the VPL contributions.

To address this problem, importance caching randomly distributes a small number of cache points in the scene. At each cache point, the contributions from all the VPLs are calculated. The probability function that is perfectly proportional to the distribution of the VPL contributions is calculated by normalizing the distribution. The contribution from the VPLs to the shading point seems to be correlated with those stored at nearby cache points. By exploiting the correlation of the contributions, the probability function p at each shading point is obtained by interpolating those at nearby cache points. However, when geometrical information (e.g. the normal) or the VPL contribution between a shading point and a cache point has a small correlation as shown in Figs. 1(c)(d), the proportionality of the interpolated probability function decreases.

4 PROPOSED METHOD

Instead of random sampling, our method distributes cache points taking into account the geometrical information of the shading points and the distribution of the VPL contributions. Fig. 2 shows an overview of our method.

4.1 Generating Initial Cache Points

The contributions from a VPL to two shading points $\mathbf{x}_i,\mathbf{x}_j$ have large correlation when the positions $\mathbf{x}_i,\mathbf{x}_j$ and the normals $\mathbf{n}_i,\mathbf{n}_j$ to the shading points are similar. By exploiting this, our method first clusters the shading points based on the positions and the normals, employing the clustering method described in [OU11]. The shading points are represented by 6-dimensional points consisting of the positions and the normals. Firstly, the positions of the shading points are normalized into $[-1,1]^3$, which is equal to the range of the normals. The bounding box of the 6-dimensional points is calculated, and then recursively subdivided until the number of 6-dimensional points or the size of bounding box is smaller than the thresholds. The bounding box is split along its longest axis. After the subdivision is terminated, one shading point is randomly sampled from each cluster and is used as the cache point. At each cache point, the contributions from all the VPLs are calculated and a cumulative distribution function is constructed.

4.2 Adaptive Insertion of Cache Points

The initial cache points are distributed according to the similarity of the shading points, but not considering the contributions of VPLs. For example, as shown in Fig. 1(d), the contributions of VPLs can differ at nearby cache points due to occlusions, leading to the interpolated probability function having reduced proportionality and increased variance.

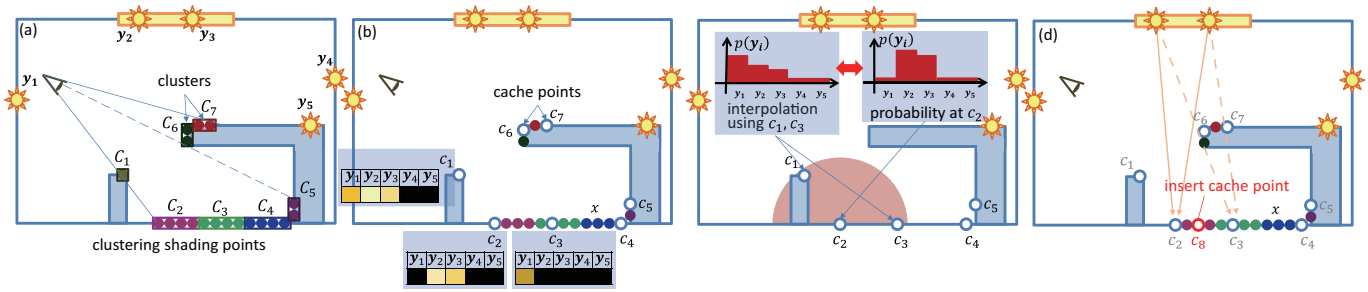


Figure 2: Overview of our method. (a) Clustering shading points based on their positions and normals. (b) Calculate contributions of VPLs at each cache point c_j. (c) Calculate sum of differences of p at c_2 and interpolated probability function using c_1 and c_3. (d) Insert new cache point c_8 from cluster C_2.

To address this problem, our method exploits the fact that each cache point records the contributions of all VPLs and therefore, an ideal probability function perfectly proportional to the contributions of all the VPLs is easily obtained. Our method detects those nearby cache points whose recorded contributions differ due to occlusions, and inserts additional cache points for such regions. If the VPL contributions recorded at cache points near to c_j differ drastically from those at c_j due to occlusions, the interpolated probability function differs from the probability function of c_j. Therefore, our method calculates the sum of the differences between the probability function recorded at c_j and interpolated probability function from nearby cache points of c_j. If the sum of differences exceeds the threshold δ, an additional cache point is inserted. The threshold δ is set experimentally in the current implementation.

The VPL contributions at the additional cache point need to have large correlation with those recorded at c_j. To correlate VPL contributions between the additional cache point and c_j, small variations in the geometrical information and the occlusions are required. However, computing the visibilities between all VPLs and a cache point is computationally expensive, it is difficult to detect the variations in the occlusions with a small overhead. Our method calculates the positions of the additional cache points using the geometrical information of c_j. Since the shading points in the cluster C_j corresponding to c_j have similar geometry information, our method samples one shading point randomly from C_j. The insertion process for all the cache points is repeated until the number of inserted cache points is smaller than a threshold. The cache points are stored in a kd-tree for fast search of cache points near to each shading point.

Fig. 3 shows the initial cache points (left) and the adaptively inserted cache points (right) of a Cornell box scene. The initial cache points are distributed uniformly on the surfaces of the scene, while the inserted cache points are distributed near the boundaries of shadows, where the visibilities between the VPLs and the cache points change.

4.3 Rendering

The outgoing radiance $L_o(\mathbf{x}, \mathbf{x}_v)$ at shading point $\mathbf{x}$ is calculated by sampling VPLs according to the probability function p interpolated from those recorded at a number, M, of nearby cache points. In contrast to the simple average as in importance caching [GEORGIEV12], our method calculates the probability function p using a weighted average that considers the spatial and directional correlations between the shading and cache points. The probability function p is calculated by the following equation:

$$p(\mathbf{y}_n) = \sum_{k=1}^{M} w_k p_k(\mathbf{y}_n), \tag{2}$$

where M is the number of cache points. $M = 3$ works well for our method as proposed in [GEORGIEV12]. w_k and p_k are the weight and probability function for the k-th nearest cache point, respectively. The weight w_k is calculated using the formula proposed in [CLARBERG08].

$$w_k = \sqrt{1 - |\mathbf{n} \cdot \mathbf{v}|} \cdot \hat{w}(d, \theta), \tag{3}$$

where $\mathbf{n}$ is the normal to shading point $\mathbf{x}$, and $\mathbf{v}$ is the normalized vector from $\mathbf{x}$ to the k-th nearest cache point c_k. d is the distance between the shading point and the cache point, and θ is the angle between the normals to the shading and cache points. The weight function $\hat{w}$ is calculated from the unnormalized weight function w:

$$w(d, \theta) = \left(1 - \frac{\theta}{\pi}\right)\left(1 - \frac{d'}{1 + \lambda d'}\right), \tag{4}$$

where $d' = d/d_{max}$, d_{max} is the maximum search range, and λ is a parameter. The weight function $\hat{w}$ is calculated from $\hat{w} = (w(d, \theta) - w(d_{max}, \theta_{max}))/(1 - w(d_{max}, \theta_{max}))$, where θ_{max} is the maximum angle between the normals. $\theta_{max} = \pi/6$ and $\lambda = 5$ are used as proposed in [CLARBERG08].

5 RESULTS

Figs. 4, 5, and 6 show equal-time comparisons between our method and importance caching [GEORGIEV12].

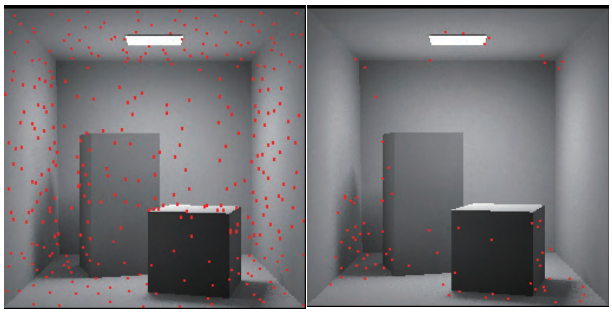


Figure 3: Example of cache points (red points) in the Cornell box scene. Left: initial cache points generated by sampling each cluster of shading points. Right: inserted cache points generated by considering VPL contributions.

The computational times are measured on a desktop PC with an Intel Xeon CPU E3 1270 3.4GHz. All the computations were performed in parallel using multi-threading. The image resolutions are 1024×768. All the computations used in our method, except for cache generation, adaptive insertion, and weight calculation, are the same as those used in importance caching. Cache points are generated randomly in importance caching, and the same numbers of cache points are used in our method as in importance caching. As described in [GEORGIEV12], four sampling strategies are used. Bilateral multiple importance sampling using the α-max heuristic is performed for both methods. Our method distributes VPLs in the same way as described in [DACHSBACHER14]. The reference images are rendered by accumulating the contributions from all the VPLs. Table 1 shows the statistics of our results.

Fig. 4 shows an equal-time comparison of a Sibenik scene. The computational times (cache generation/rendering) for our method and importance caching were 36.6s (17.8s for cache generation/18.8s for rendering) and 33.7s (15.2s/18.5s), respectively. Fig. 4(a) shows the result rendered using our method. Figs. 4(b), (c), and (d) show close-ups of the area outlined in red in Fig. 4(a) for the reference image, and the results rendered by our method, and importance caching, respectively. Figs. 4(e), (f), and (g) show close-ups of the area outlined in blue in Fig. 4(a). As shown in these images, our method can render images with less noise compared to importance caching. Importance caching [GEORGIEV12] tends to distribute cache points near the viewpoint. Therefore regions far from the viewpoint (e.g. the windows in Fig. 4(d)) have less cache points, resulting in noisy images. In addition, since importance caching does not take into account occlusions in distributing cache points, the regions where occlusions vary drastically (e.g. Fig. 4(g)) suffer from noise, whereas our method can distribute cache points for such regions, resulting in less noise as shown in Fig. 4(f).

Table 1: Statistics of results. N_T, N, and N_c are the number of triangles, VPLs, and cache points, respectively.

Scene	N_T	N	N_c
Sibenik (Fig. 4)	75,284	7,785	2,950
Sponza (Fig, 5)	66,450	6,479	1,728
Conference (Fig. 6)	331,179	5,133	2,478

Fig. 5 shows an equal-time comparison of a Sponza scene. As shown in Figs. 5(b) to (g), our method can render less noisy images especially for regions (e.g. arches and pillars) where the visibilities between the VPLs and the shading points change. The computational times (cache generation/rendering) for our method and importance caching were 24.7s (7.5s/17.2s) and 23.3s (5.9s/17.3s), respectively.

Fig. 6 shows an equal-time comparison of a Conference scene. Figs. 6(b)(d), (c)(f), and (d)(g) show close-ups of the reference image, the results rendered by our method, and importance caching, respectively. In Fig. 6(d), a large variance due to the occlusion due to the table appears in the chair, whereas our method (Fig. 6(c)) renders an image comparable to the reference image shown in Fig. 6(b). The computational times for our method and importance caching were 18.7s (6.5s/12.2s) and 18.2s (6.4s/11.9s), respectively.

Figs. 7, 8, and 9 show visualizations of the root-mean-square-error (RMSE) between each method and reference images rendered by summing all the VPL contributions. The color bar shows the false color. For the Sibenik scene, the RMSE for our method is 0.0486773 while that for importance caching is 0.0662813. As shown in Fig. 7, our method can render less noisy images especially near windows and pillars. In the Sponza scene, the RMSE for our method is 0.146164 while that of importance caching is 0.207015. Since it is difficult for importance caching to distribute cache points inside the scene, large variance can appear as shown in Fig. 8(b), while our method can lessen this as shown in Fig. 8(a). In the Conference scene (Fig. 9), the RMSE for our method and importance caching are 0.0294903 and 0.032524, respectively. As shown in Fig. 9, by inserting additional cache points, our method reduces the variance near chairs occluded by the table.

Since, with our method, new cache points are added in regions where the variations in the VPL contributions are large, for the same number of cache points, the cache points are distributed more sparsely in other regions compared to importance caching, resulting in a slightly increased variance (e.g. near the floor in Fig. 8). However, our method can reduce the RMSE for the overall scene as shown in Figs. 7, 8, and 9.

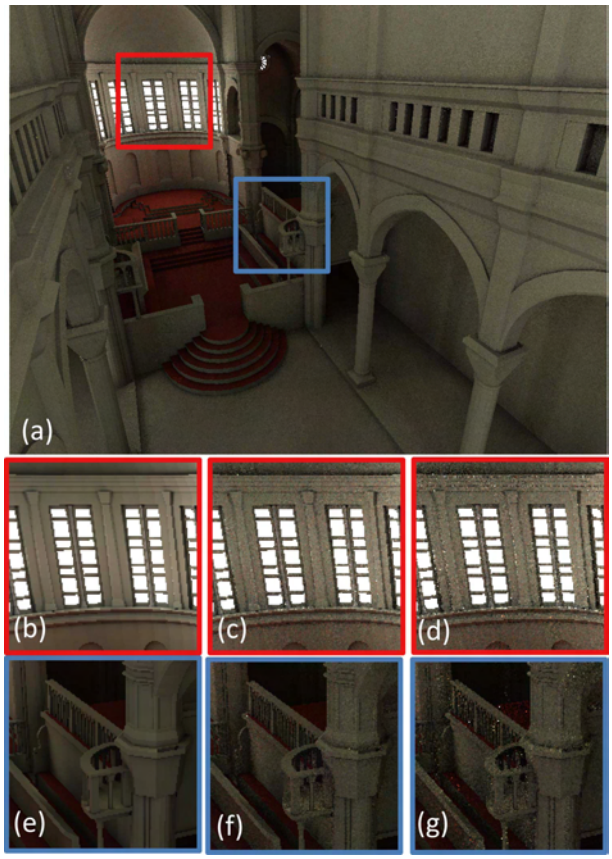


Figure 4: Sibenik scene. (a) rendering result of our method. (b)(c)(d) close-up images of reference, our method, and importance caching, respectively. (e)(f)(g) close-up images of reference, our method, and importance caching. Our method can render less noisy image in equal time rendering compared to importance caching.

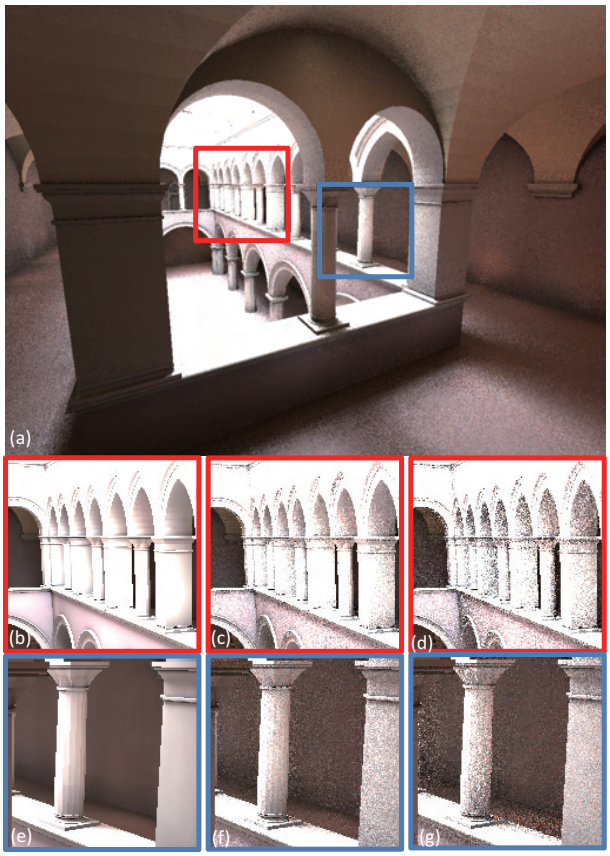


Figure 5: Sponza scene. (a) rendering result of our method. (b)(c)(d) close-up images of reference, our method, and importance caching, respectively. (e)(f)(g) close-up images of reference, our method, and importance caching. Our method can render less noisy images, especially near arches and pillars in equal time rendering.

To inspect the effectiveness of the adaptive insertion of cache points and the weighting function that considers the spatial and directional correlations, our method renders the Sibenik scene using adaptively inserted cache points and uniform weights used in [GEORGIEV12]. The RMSE in this case is 0.0516844, while that with random cache points and uniform weights is 0.0662813. As shown in this experiment, adaptive cache insertion contributes to the improvements most.

6 CONCLUSIONS AND FUTURE WORK

We have proposed an adaptive cache insertion method for importance caching. Our method clusters the shading points and selects cache points from clusters and exploits the spatial and directional correlations between shading points and cache points. Our method detects cache points whose VPL contributions differ from those of nearby cache points and inserts further cache points, resulting in reduced variance compared to that obtained in equal-time rendering using the original importance caching method.

For future work, we plan to accelerate our method using VPL clustering. Moreover, we propose to distribute cache points taking into account the scene saliency.

ACKNOWLEDGEMENTS

This research was partially supported by JST, CREST.

7 REFERENCES

[KELLER97] A. Keller, Instant Radiosity, Proc. of SIGGRAPH '97, pp. 49-56, 1997.

[DACHSBACHER14] C. Dachsbacher, J. Krivanek, M. Hasan, A. Arbree, B. Walter, J. Novak, Scalable Realistic Rendering with Many-Light Methods, Computer Graphics Forum, Vol. 33, No. 1, pp. 88-104, 2014.

[WANG09] R. Wang, O. Akerlund, Bidirectional Importance Sampling for Unstructured Direct Illumination, Computer Graphics Forum, Vol. 28, No. 2, pp. 269-278, 2009.

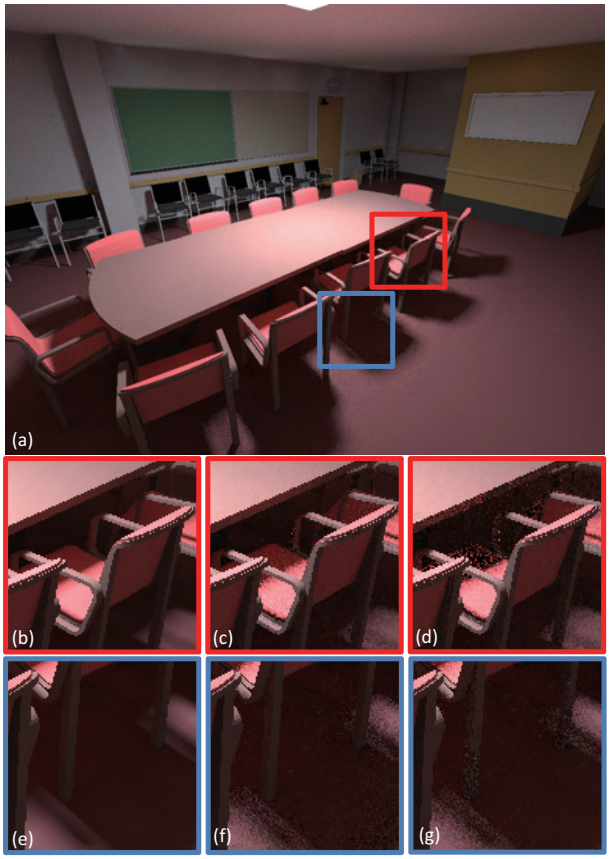


Figure 6: Conference scene. (a) rendering result of our method. (b)(c)(d) close-up images of reference, our method, and importance caching, respectively. (e)(f)(g) close-up images of reference, our method, and importance caching. Our method can render less noise images, especially for chairs in equal time rendering.

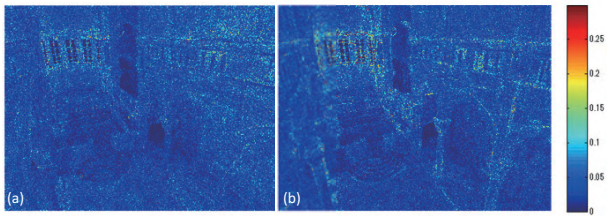


Figure 7: Error images of Sibenik scene. (a) our method and (b) importance caching. RMSEs of our method and importance caching are 0.0486773 and 0.0662813, respectively.

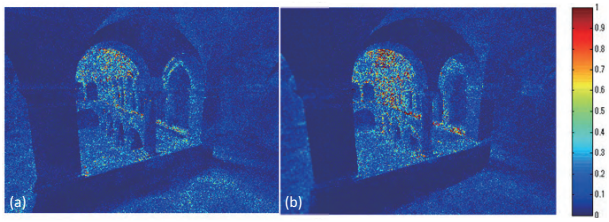


Figure 8: Error images of Sponza scene. (a) our method and (b) importance caching. RMSEs of our method and importance caching are 0.146164 and 0.207015, respectively.

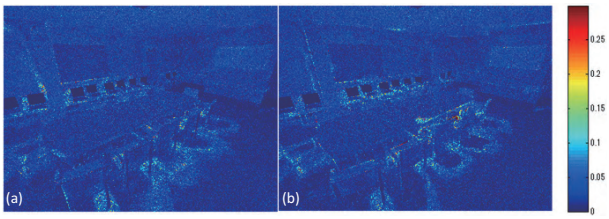


Figure 9: Error images of Conference scene. (a) our method and (b) importance caching. RMSEs of our method and importance caching are 0.0294903 and 0.032524, respectively.

[WU13] Y. T. Wu, Y. Y. Chuang, VisibilityCluster: Average Directional Visibility for Many-Light Rendering, IEEE Transactions on Visualization and Computer Graphics, Vol. 19, No. 9, pp. 1566-1578, 2013.

[GEORGIEV12] I. Georgiev, J. Krivanek, S. Popov, P. Slusallek, Importance Caching for Complex Illumination, Computer Graphics Forum, Vol. 31, No. 2, pp. 701-710, 2012.

[WALTER05] B. Walter, S. Fernandez, A. Arbree, K. Bala, M. Donikian, D. Greenberg, Lightcuts: A Scalable Approach to Illumination, ACM Transactions on Graphics, Vol. 24, No. 3, pp. 1098-1107, 2005.

[HASAN07] M. Hasan, F. Pellacini, K. Bala, Matrix Row-Column Sampling for the Many-Light Problem, ACM Transactions on Graphics, Vol. 26, No. 3, pp. 26:1-26:10, 2007.

[OU11] J. Ou, F. Pellacini, LightSlice: Matrix Slice Sampling for the Many-Lights Problem, ACM Transactions on Graphics, Vol. 30, No. 6, pp. 179:1-179:8, 2011.

[CLARBERG08] P. Clarberg, T. Akenine-Moller, Exploiting Visibility Correlation in Direct Illumination, Computer Graphics Forum, Vol. 27, No. 4, pp. 1125-1136, 2008.

[WARD88] G.J. Ward, F.M. Rubinstein, R.D. Clear, A Ray Tracing Solution for Diffuse Interreflection, ACM SIGGRAPH Computer Graphics, Vol. 22, No. 4, pp. 85-92, 1988.

[KRIVANEK05] J. Krivanek, P. Gautron, S. Pattanaik, K. Bouatouch, Radiance Caching for Efficient Global Illumination Calculation, IEEE Transactions on Visualization and Computer Graphics, Vol. 11, No. 5, pp. 550-561, 2005.

[KRIVANEK06] J. Krivanek, K. Bouatouch, S. Pattanaik, J. Zara, Making Radiance and Irradiance Caching Practical:Adaptive Caching and Neighbor Clamping, Proc. of Eurographics Symposium on Rendering, pp. 127-138, 2006.

Isosurface Orientation Estimation in Sparse Grids Using Tetrahedral Splines

Brian A. Wood
University of Alabama in Huntsville
Huntsville, Alabama 35899
bwood@cs.uah.edu

Timothy S. Newman
University of Alabama in Huntsville
Huntsville, Alabama 35899
tnewman@cs.uah.edu

ABSTRACT

One challenge in applying standard marching isosurfacing methods to sparse rectilinear grid data is addressed. This challenge, the problem of finding approximating gradients adjacent to locations with data dropouts, is addressed here by a new approach that utilizes a tetrahedral spline fitting-based strategy for gradient approximation. The new approach offers improved robustness in certain scenarios (compared to the current state-of-the-art approach for sparse grid isosurfacing). Comparative studies of the new approach's accuracy and computational performance are also presented.

Keywords

Orientation Estimation, Volume Visualization, Isosurfaces

1 INTRODUCTION

One common means for visualizing scalar volumetric data is isosurfacing, which involves finding the set of locations in space where the phenomenon recorded in the dataset achieves a particular value, called the isovalue, denoted herein as α. Isosurface visualization is a powerful approach for observing and studying the behavior of volumetric data. Isosurfacing can promote discovery in disparate applications areas, such as medical diagnosis, fluid flow studies, etc.

Well-known isosurfacing methods exist for volumetric data organized on a number of grid types [10]. Focus here is on scalar data organized on rectilinear grids, which is very common, and on isosurfacing methods applied to such grids that produce triangle meshes approximating the isosurface and assume data values are available at each grid point. However, in some applications, the data is sparse; there is not a data value available at every grid point. (Here, we will use the term sparse grid to mean a 3D rectilinear grid dataset with some missing values.) For example, data collected from sensor arrays may have missing data values when data cannot be collected at every grid point due to physical limitations. Popular isosurfacing methods for rectilinear grid data, such as the standard, marching methods of Marching Cubes and Marching Tetrahedra, require determination of a local gradient at each mesh vertex to estimate the isosurface orientation, and then use that in rendering to produce a shading that is harmonious with local data trends. When data is sparse, the schemes these methods use for estimating orientation can fail at certain locations. Thus, sparseness can make well-known isosurfacing rendering methods unable to be applied. Here, we introduce a new solution to the challenge of isosurfacing on sparse grids.

Sparse grids may be produced from a variety of sensing modalities and volume data generation methods. Data from sensor arrays, particularly ones that measure physical phenomena, has the potential to have missing data values due to sensor faults. For example, wireless 3D sensor arrays, such as those used to capture data underground [1] and underwater [17], operate under harsh conditions and can be particularly vulnerable to sensor faults. Low batteries, bad calibration, high noise, or environmental hazards can all contribute to faults in sensor arrays [11]. Conversion of 3D mesh geometry to volume data via voxelization algorithms [16] can produce datasets with data values only at grid points necessary to reproduce the original mesh. Additionally, volume data derived from point clouds or signed distance functions may not contain sufficient data to estimate data gradients at all isosurface locations, in particular the mesh vertices [12].

One prior work has proposed a work-around to the gradient (orientation) determination challenge in Marching Cubes on sparse grids. The new approach we describe here offers improved results in certain scenarios.

The paper is organized as follows. Section 2 discusses background material and related work. Section 3 de-

scribes the new approach for estimating isosurface orientation on sparse grid datasets. Section 4 provides details on rendering isosurfaces extracted from sparse grids. Section 5 provides results from experiments and comparisons to prior orientation (or normal) estimation approaches. Section 6 contains the paper's conclusion.

2 BACKGROUND AND RELATED WORK

The most common method [10] for isosurfacing on scalar data on rectilinear grids is the Marching Cubes (MC) algorithm. MC has been adapted by Nielson et al. [12] to allow application to rectilinear grids with missing data values (i.e., sparse grids). We describe that adaptation in Section 3.2. First, though, we describe the basic steps of MC and illustrate its failings for sparse grids.

Marching Cubes isosurfacing produces a triangle mesh representation of the isosurface by advancing cell-by-cell through the volume. In each cell, it follows three major steps. In the first step, the general topological arrangement of the isosurface mesh in the cell is determined. (Each general topological arrangement is called a "case" in this paper, reflecting the typical nomenclature of the MC literature.) Second, for topologies containing isosurface mesh facets, the mesh vertex locations in the cell are found. Third, the triangle mesh is formed by connecting vertex locations into the determined topology. An orientation vector is also determined for each vertex location.

In MC, mesh vertices are located on grid lines, with positions there found via linear interpolation. At each vertex, an orientation vector is ultimately used in rendering the produced mesh. These vectors are determined by linearly interpolating the gradients of the grid point locations bounding the grid segment containing each mesh vertex. These gradients are computed using central differencing; for grid point (x_i, y_i, z_i), MC finds the gradient ∇f as:

$$\nabla f(x_i,y_i,z_i) = \left\{ \begin{array}{c} \frac{f(x_{i+1},y_i,z_i) - f(x_{i-1},y_i,z_i)}{2} \\ \frac{f(x_i,y_{i+1},z_i) - f(x_i,y_{i-1},z_i)}{2} \\ \frac{f(x_i,y_i,z_{i+1}) - f(x_i,y_i,z_{i-1})}{2} \end{array} \right\}, \quad (1)$$

where $f(x_i, y_i, z_i)$ is the scalar value at (x_i, y_i, z_i).

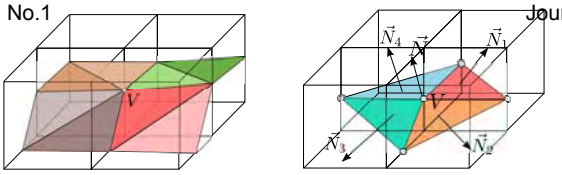
Since the central-difference gradient uses the values of adjacent grid points, if there is a missing data value preceding or following a grid point in any axial direction, central-differencing will be undefined. As a result, MC is unable to estimate the orientation vector for any mesh vertex on a grid segment whose endpoint has an undefined gradient value. Data sets with missing or undefined data thus require an alternative orientation estimator. One option could be use of ad-hoc alternatives for those grid points where central differencing is undefined. For example, a mix of methods could be used (e.g., forward-differencing and reverse-differencing, as suitable) at a cost of consistency.

Other works have considered the issue of estimating orientation in volume data without relying on differencing techniques. For example, Möller et al. [15] have used a two-step approach for shading raytraced isosurface renderings. Hossain et al. [8] have proposed reconstruction filters for gradient estimation derived from methods using Taylor series and Hilbert spaces. They evaluated the accuracy of their filters on both Cartesian and Body-Centered Cubic lattices. Correa et al. [4] have studied averaging-based and regression-based orientation estimation approaches for use in volume raycasting on unstructured grids. Their study recommended the use of a hybrid approach that selects the gradient estimator to use based on local properties of the unstructured grid. Neumann et al. [9] have estimated orientation by fitting a hyperplane on points nearby to a grid point and then taking a linear regression result on data points on the hyperplane. However, while these orientation estimation approaches do not rely on differencing, they assume that data is available at all grid points and thus cannot be used with sparse grids.

Other methods for producing visualizations of sparse grid volume data have also been described. For example, Djurcilov and Pang [6] have described some techniques for visualizing weather data when sample points are missing due to sensor failures. Their techniques require resampling data to produce a fully populated grid prior to isosurface extraction.

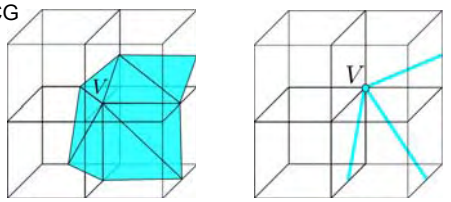
2.1 Quadratic and Quintic Splines

Rössl et al. [14] have proposed a technique for volume reconstruction by fitting a spline model to regular, rectilinear volumetric data. Their technique first partitions the volume's grid into uniform tetrahedra and then fits super splines on each partition. Super splines are a class of splines in which smoothness is preserved on vertices between adjacent tetrahedra. Each fitting uses Bézier splines with constants drawn from the values at the vertices of each tetrahedron, ensuring that the super spline condition is not violated. Details of their process are described later, in Section 3. Awanou and Lai [2] have presented an approach using quintic splines to interpolate a volume. Their approach is similar to that of Rössl et al., but it does not require a regular grid and uses a higher order spline to model the volume data.



(a) Marching Cubes output (b) LSN virtual local mesh

Figure 1: LSN perp vector estimation



(a) Non-degenerate mesh (b) Degenerate mesh used by LSN

Figure 2: LSN summed average estimation

2.2 Locally Supported Normals (LSN)

One methodology for determining isosurface orientation in sparse rectilinear grid data is the Locally Supported Normal (LSN) approach described by Nielson et al. [12]. It considers isosurfacing in a Marching Cubes context, resolving the undefined orientation problem by an estimation process that uses a virtual mesh constructed on the vertices of the MC isosurface. Orientation vectors computed in this manner tend to exhibit sharper shading color transitions at triangle edges than if central-differencing could be used, resulting in a surface with a more faceted appearance. However, central-differencing cannot be applied where grid values are missing or undefined.

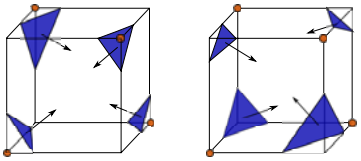
The estimation used in LSN is integrated into MC-style isosurfacing; it produces orientation estimates as vertex locations are calculated. The LSN approach relies on a temporary virtual local mesh that it defines about the point for which an orientation vector is needed. This virtual mesh is *not* the Marching Cubes output mesh; Figure 1 demonstrates the difference between a mesh produced by MC and the virtual mesh used by LSN for a point V. The LSN approach first computes perpendicular (perp) vectors for each face in the virtual mesh; these vertex perp operations are done independent of the MC topology determination. Each perp vector is found as the cross-product of edge vectors of the virtual mesh face. For each of the MC internal vertices shared by multiple triangles, all perp vectors of faces incident to it are averaged to form a *master perp vector* at the vertex. The master perp vector becomes the LSN's estimate of the isosurface orientation at that vertex. Figure 1(b) shows the LSN's estimation of the orientation for a location V in a volume. Four perp vectors, $\vec{N}_1, \vec{N}_2, \vec{N}_3$, and $\vec{N}_4$, are shown. The average of these is the master perp vector $\vec{N}$; here, $\vec{N}$ is $\frac{1}{4}\sum_{i=1}^{4}\vec{N}_i$.

The LSN's estimation can produce erroneous results when certain data characteristics are encountered. The first, and most pronounced, of these errors occurs when degenerate triangles are encountered during orientation estimation. A degenerate triangle with two or more coincident vertices will yield a cross-product of zero, resulting in a zero vector (because the triangle does not lie on a unique plane in space). MC produces degenerate triangles when the isovalue is identical to a grid point value [13]. If a vertex is associated with only degenerate triangles, the orientation vector computed using LSN at that vertex has length zero. The rendered isosurface can contain artifacts at pixel locations affected by the zero length orientation vector.

In Figure 2(a), a mesh containing no degenerate triangles is shown. In contrast, Figure 2(b) displays an LSN mesh corresponding to the same topology, but with vertex V located at a cube corner, resulting in four degenerate triangles (one triangle degenerating to a point and three to a line). The result from LSN is a zero length orientation vector.

Additionally, the LSN approach makes assumptions about what have been called ambiguous faces [10] of cells. These assumptions can lead to inaccurate orientation vectors. One example cube where this incorrect assumption is a problem is shown in Figure 3. The cube has the Case 13 base topology of the MC [12], shown in Figure 3(a). However, the LSN estimation uses the virtual mesh shown in Figure 3(b) to compute orientation vectors in corners of the cube opposite to those defined by MC. We refer to triangles used in the LSN virtual mesh that do not appear in the MC mesh as *illusory triangles*. The normals (i.e., perp vectors) associated with these triangles may differ greatly from the orientation vectors that would result if the actual MC isosurface facets had been used. In particular, each vector found using an illusory triangle will contribute errors to the orientation vector estimation at vertices of illusory triangles. For such situations, the orientations can be estimated incorrectly and yield incorrectly shaded renderings.

The illusory triangle problem in LSN is not just limited to cubes with ambiguous faces. For example, in Marching Cubes Case 5 LSN uses an illusory triangle to compute a perp vector. The topology used by MC for the Case 5 topology is shown at the top of Figure 4 (labeled "C5"). The five virtual mesh triangles used by LSN are shown in the rest of the figure. While most of the virtual mesh triangles should produce reasonable results, the one used for V_4 is illusory and its orientation is not consistent with the actual mesh properties at V_4. Other MC cases also exhibit illusory triangles yielding orientations that differ markedly from that of the MC isosurface mesh. An example of the incorrect orientation from illusory triangles is provided later in this work.



(a) Topology of cell as defined by Marching Cubes

(b) Topology of cell assumed by the LSN approach

Figure 3: Comparison of cell topologies used by MC and LSN

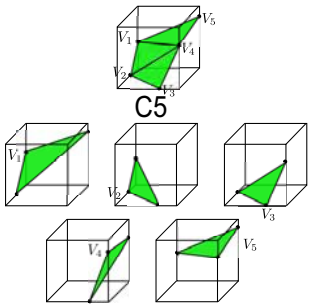


Figure 4: Case 5 MC and LSN topologies

3 NEW GRADIENT ESTIMATION APPROACH FOR SPARSE GRIDS

Next, we describe our new approach for determining orientation vectors in MC for sparse grids. The approach is guaranteed to produce orientation vectors at any location for which it is possible to find a Marching Cubes isosurface vertex. That is, the scheme introduced here can handle any rectilinear sparse grid configuration satisfying the condition that the isosurface vertices can be computed. (I.e., like LSN, our approach assumes there is local support for the isosurface.) For some scenarios, it also offers improved performance over prior approaches for computing MC isosurface orientation vectors on sparse grids.

3.1 Using Quadratic Splines

Our work is motivated by Rössl et al.'s modeling of volumetric data variation using quadratic Bézier-Bernstein super splines (2BBSS) in tetrahedral regions. A tetrahedron allows for the use of an interpolating volumetric spline using a barycentric coordinate system given a sufficient number of data points on the tetrahedron. Specifically, given four points v_0, v_1, v_2, v_3 defining the four vertices of a tetrahedron, a quadratic trivariate spline p is composed in the Bézier-Bernstein form:

$$p(\lambda) = \sum_{i+j+k+l=2} a_{ijkl} B_{ijkl}(\lambda), \qquad (2)$$

where the parameter λ is the location within the spline (in barycentric coordinates with $\lambda = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)$), the coefficients a_{ijkl} are the control points of the spline,

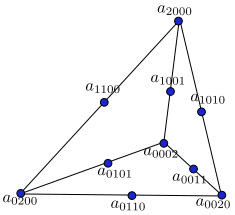


Figure 5: Spline control points

and the B_{ijkl}'s are Bernstein polynomials. The control points are calculated as linear combinations of the vertices of the tetrahedron:

$$a_{ijkl} = \frac{i}{2}v_0 + \frac{j}{2}v_1 + \frac{k}{2}v_2 + \frac{l}{2}v_3, \qquad (3)$$

as depicted in Figure 5. The Bernstein polynomials B_{ijkl} are defined as

$$B_{ijkl}(\lambda) = \frac{2!}{i!j!k!l!}\lambda_0^i\lambda_1^j\lambda_2^k\lambda_3^l,\ i+j+k+l=2, \qquad (4)$$

where each $\lambda = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ is a barycentric coordinate with respect to the tetrahedron.

Numerous schemes exist for partitioning rectilinear grids into collections of tetrahedra. We employ one such scheme here to enable the use of tetrahedral splines in the estimation of orientation vectors. Tetrahedral partitions also alleviate the problem of missing data because only 4 grid values are needed to model isosurface behavior within a tetrahedral partition, as opposed to the 6 necessary for a central differencing. By partitioning rectilinear dataset cells into tetrahedra, we can calculate an orientation in any cell intersected by the isosurface. In the 2BBSS model, each tetrahedron must have associated data values at each tetrahedral vertex. Given such, a spline is formulated that approximates the surface within the tetrahedron.

Our approach finds the approximating spline in cells intersected by the isosurface by partitioning the cell into tetrahedra and then evaluating the spline constructed on those tetrahedra to determine orientation vectors (i.e., spline normals) at any barycentric coordinate $(\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ within each tetrahedron of interest. For each of them, our approach uses de Casteljau's algorithm [3] [5] to determine the spline's partial derivative [14] by applying the algorithm in the direction of tetrahedron edges. The usage of de Casteljau's algorithm to compute the derivative of a curve is well understood [7].

For any point on a spline, the formulation of de Casteljau's algorithm enables finding the directional derivatives at q as follows. First, given a spline with control points of the form a^0_{ijkl}, for a point q having the

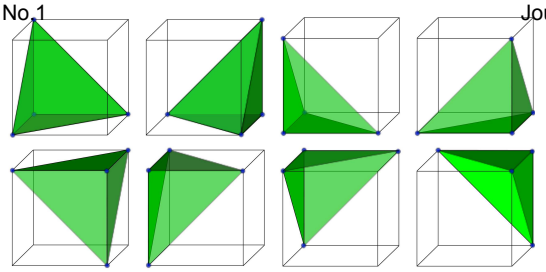
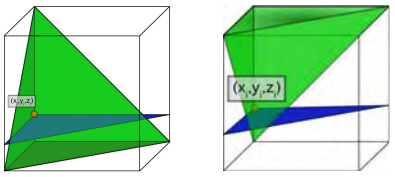


Figure 6: Tetrahedral partitions



(a) Tetrahedral partition 1

(b) Tetrahedral partition 2

Figure 7: Two choices of tetrahedral partition of the cell

barycentric coordinate $\lambda_{\mathbf{q}} = (\lambda_{0_q}, \lambda_{1_q}, \lambda_{2_q}, \lambda_{3_q})$, new control points denoted by a^1_{ijkl} are computed as:

$$a^1_{ijkl} = \lambda_{0_q} a^0_{i+1,j,k,l} + \lambda_{1_q} a^0_{i,j+1,k,l} + \lambda_{2_q} a^0_{i,j,k+1,l} + \lambda_{3_q} a^0_{i,j,k,l+1}, \quad (5)$$

which define a subdivision of the original spline. (Another application of the formula would produce the value at q, however we need just the control points a^1_{ijkl} of the spline subdivision because they define partial derivatives for the spline.) Since the normal at any point on a surface $s(x,y,z)$ can be defined as

$$\nabla s(x,y,z) = (\frac{\partial s}{\partial x}, \frac{\partial s}{\partial y}, \frac{\partial s}{\partial z}), \quad (6)$$

we compute the orientation by finding the partial derivatives in the directions parallel to the coordinate system axes. The formulation of this partial derivative is given in Section 4.

4 ISOSURFACE RENDERING WITH SPARSE GRIDS

Our approach defines 2BBSS splines for tetrahedral subregions of each active cell. We consider eight candidate tetrahedral partitions of each cell (shown in Figure 6) and choose from these the one that enables the most accurate estimate of the orientation vector. The choice is described shortly. This orientation estimation is based on a 2BBSS approximation of the volume within that tetrahedron. The eight tetrahedra were chosen because they share the property that three tetrehdron faces are coplanar with faces of the cell which helped simplify the construction of the spline.

For each isosurface mesh vertex, there are two candidate tetrahedra from which the orientation at that vertex could be computed. Next, how our approach decides on the one to use is described. An example situation is shown in Figure 7. In it, the vertex shown in red is located on the rear edge of the cell. One candidate tetrahedron is shown in Figure 7(a) and the other is shown in Figure 7(b). For the case where the vertex lies on an isosurface mesh triangle completely located within a tetrahedron, that tetrahedron is chosen. However, a triangle's surface may span both possible choices of tetrahedra. For such cases, tetrahedron selection is done instead by considering the total number of isosurface mesh triangle edges; we select the tetrahedron containing the greatest number of triangle edges. We have found that selection using this criterion provides more accurate orientation vectors than using a static tetrahedral partition that is ignorant of the triangles's location in the cell. Our approach uses an adaptation of the MC topological case lookup table to record the tetrahedral selections, allowing fast determination of the tetrahedron as well as supporting orientation vector determination coincident with mesh determination (i.e., within an extended MC context).

Next, we describe the orientation determination procedure. The partial derivative of the spline $p(\lambda)$ in the direction ξ_ϕ of a tetrahedron edge $v_\phi - v$ is given by

$$\frac{\partial p}{\partial \xi_\phi} = 2 \sum_{i+j+k+l=1} (a_{i,j+b,k+c,l+d} - a_{i+1,j,k,l}) \lambda_0^i \lambda_1^j \lambda_2^k \lambda_3^l, \quad (7)$$

where $(\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ are the barycentric coordinate variables of the spline equation and (b,c,d) is used to define an offset to a tetrahedral vertex in direction ξ_ϕ.

Figure 8 illustrates the vector calculations when finding the partial derivative in the x direction. The arrows on tetrahedron edges indicate a forward difference calculation using the tetrahedron vertices of that edge. The partial derivative is a linear combination of the differences, with weights for each component dependent on the particular tetrahedral partition being used within the cell. Similar vectors are computed for partial derivatives in the y and z directions.

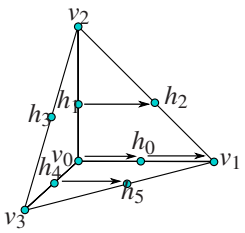


Figure 8: Computing the orientation from sample points

Since cell vertices are located on grid edges, each mesh vertex is guaranteed to have at least two zero-valued components of its barycentric coordinate. By determining which edge type the vertex is located on, we can choose the most appropriate equation to minimize the number of calculations required. For instance, for the tetrahedron shown in Figure 8, the gradient for a vertex on any edge parallel to the base is given by:

$$\nabla F = \left\{ \begin{array}{l} \gamma_0 \sum_{i+j=1} (a_{i,j+1,0,0} - a_{i+1,j,0,0}) \lambda_0^i \lambda_1^j, \\ \gamma_1 \sum_{i+k=1} (a_{i,0,k+1,0} - a_{i+1,0,k,0}) \lambda_0^i \lambda_2^k, \\ \gamma_2 \sum_{i+l=1} (a_{i,0,0,l+1} - a_{i+1,0,0,l}) \lambda_0^i \lambda_3^l \end{array} \right\}, \tag{8}$$

where γ_ν, $\nu = 0,1,2, \gamma = \pm 1$, is an orienting coefficient. For the example in Figure 8, formulation of the spline assumes a tetrahedron oriented as in Figure 8, however the tetrahedral partition used may be a reflection or rotation (or combination of both) of this orientation. The γ coefficient, which corrects for reflected or rotated instances, allows correcting the directions the components of the orientation vector.

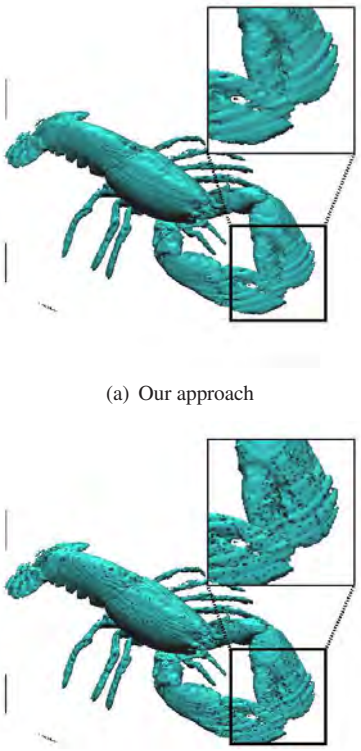
For each tetrahedron the mesh vertex is located in, a gradient vector is produced by evaluating Equation 9. Vertices will be shared among up to four tetrahedra, resulting in as many as four separate vectors per vertex. The orientation vector ultimately assigned to the vertex is the mean of these four gradient vectors.

5 RESULTS

In this section we present results of experiments to evaluate our approach versus LSN. These experiments consider accuracy of orientation vectors and the run times to compute them. We also report a qualitative evaluation of rendered images to determine the impact of degenerate triangles on each approach.

Accuracy was tested by comparing orientation vectors computed using our spline-derived orientations against the orientations using the LSN approach, then comparing these against orientations computed using central-differencing. Eight well-known real (sensed) volume datasets and five mathematically-defined datasets were used in testing. Additionally, we performed visual comparisons of the rendered images to determine if there was a difference between renderings made using the two orientation estimation approaches.

The datasets were converted to sparse grid representations by removing all grid values that were not required by MC to extract the isosurface with marker values. Specifically, grid points that were not on grid edges containing a mesh vertex were set to marker values. By removing all data points that do not contribute to the isosurface extraction, we could operate on volumes with the least possible number of defined values and thus the least favorable datasets for the classic central difference orientation estimation approach used in MC.



(a) Our approach

(b) LSN approach

Figure 9: Renderings performed using both orientation estimation approaches.

5.1 Measurement of Orientation Estimation Accuracy

Isosurfaces were extracted using Marching Cubes for ranges of isovalues on the eight sensed datasets. The range was made large so that results would not be biased against a particular sub-range of isovalues. A root mean square (RMS) error for each isosurface was calculated by comparing the angular difference (in radians) of all orientation vectors produced by both estimation approaches against the central-difference estimate. The central-difference is the baseline in this error comparison because it is equivalent to computing the gradient of a second-order data fitting at each grid point. The mean RMS error of each dataset at all tested isovalues is shown in Table 1. Inspection of individual isovalues on some datasets showed that LSN was sometimes more accurate than our approach, but on average ours appears to be the superior approach. The spline orientation estimation produced more accurate orientation estimates (on average) than the LSN approach in all datasets except for the Engine dataset.

Table 2 shows the RMS errors for 9 isosurfaces extracted on the sensed datasets. LSN does occasionally produce more accurate results, however our orientation

Dataset	Ours	LSN
Foot	0.531	0.547
Frog	0.569	0.589
Lobster	0.369	0.375
MRA	0.639	0.653
Piggy bank	0.876	0.898
Backpack	0.561	0.568
Sheep heart	0.313	0.315
Engine	0.204	0.187

Table 1: Average RMS error of approaches vs. central-difference

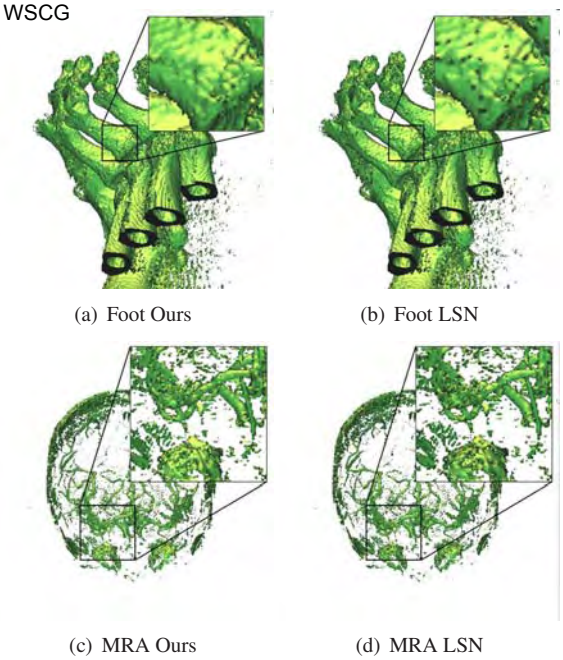
Dataset	Isovalue	Ours	LSN
MRA	65	0.740	0.764
	75	0.684	0.714
	80	0.775	0.783
Foot	80	0.555	0.572
	90	0.502	0.518
	100	0.472	0.322
Frog	40	0.512	0.524
	45	0.513	0.523
	80	0.545	0.640
Lobster	50	0.318	0.316
	65	0.329	0.332
	80	0.336	0.338

Table 2: RMS error of approaches vs. central-difference

estimation produces more accurate results in the majority of cases we tested. Figure 10 shows MRA and Foot isosurfaces (for $\alpha = 65$ and 90, respectively). The magnified callouts show subtle differences in the two renderings, but both are very similar to the baseline images produced using central-difference gradient estimates.

5.2 Accuracy using Mathematically Defined Data

Experiments were also performed to measure the accuracy of the orientation estimation approaches versus exact orientation vector values. These experiments tested scalar fields generated using five mathematically defined fields. The isosurfaces were generated corresponding to level sets (i.e., implicit surfaces) of these fields. Orientation vectors were estimated using our spline-based estimation, the LSN estimation, and the standard MC central-difference approaches. Orientation vectors at each location were compared against the exact orientation vector values computed at the isosurface intersection locations. Table 3 reports the RMS error with respect to the exact orientation vectors for isosurfaces of the zero level set. Excepting the Marschner-Lobb dataset, the central-difference estimates are superior to both LSN and our orientation estimations. But LSN estimates are sometimes better than ours. Thus, empirical evidence suggests that, for mathematically



(a) Foot Ours (b) Foot LSN

(c) MRA Ours (d) MRA LSN

Figure 10: Zoomed comparison of isosurface images

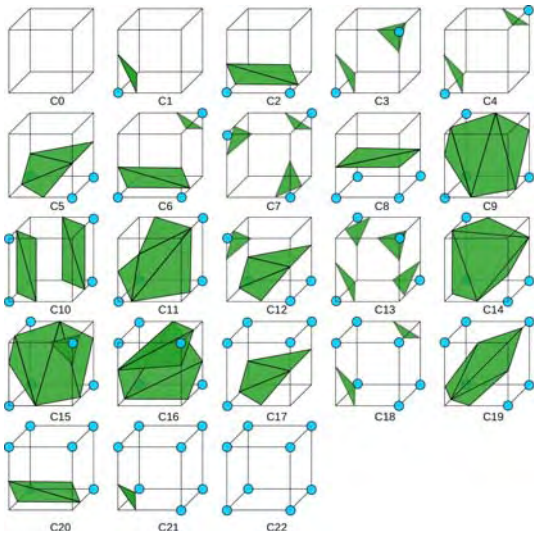


Figure 11: MC lookup table base topologies

defined, noise-free data, LSN estimation may be quite suitable; LSN estimation may provide more accurate normal estimation than our approach for many mathematically defined scalar fields.

5.3 Individual MC Topologies

Since results for the mathematically defined datasets were incongruous with those observed for sensed data (where our approach appears to be better than LSN), we performed an analysis of occurrences of MC base topologies defined in the MC lookup table [12] to determine if one estimation approach produced more accurate orientation vectors for particular base topologies.

Dataset	Ours	LSN	Central Diff
Mar-Lobb	0.0545	0.0616	0.0718
Six Peaks	0.0139	0.0534	0.0179
Genus_3	0.00622	0.00506	0.000265
Flower	0.0261	0.0261	0.0183
Peaks	0.0372	0.0235	0.0218

Table 3: RMS error calculated versus exact orientations

The base topologies are shown in Figure 11 with labels C_i (C_i means "Case i", as used throughout this section). The isosurfaces extracted from mathematically defined datasets showed no occurrences of the Case 4, 7, 12, and 15 topologies. Additionally, very low occurences were observed for Cases 6, 10, 11, 12, 14, 15, 18 and 19. Many of these topologies consist of disconnected triangles within a cell. Due to the nature of the level sets MC produced for these datasets it is not unexpected that occurrences of these topologies would be rare. The sensed data contained far more examples of these topologies. While for some isovalues, there were no instances of a few topological cases, such situations were observed less frequently than for the synthesized datasets. For one dataset (MRA), some isovalues did not give rise to any cells of the type Case 15 or 18. For one dataset (the Engine dataset), the majority of isovalues did not give rise to any of Case 4, 7, 13, or 15 cells. This may be a result of the engine structure in the dataset being manufactured from a CAD model that had a limited number of basic surface types.

To determine the effect that particular base topologies had on orientation estimation accuracy, we considered RMS error of orientation vectors on a topological basis for sensed data isosurfaces. The Case 7, 10 12, 13, 15, and 19 topologies demonstrated much lower RMS errors for our estimation than for LSN estimation. LSN estimation produced consistently more accurate orientation vectors for the Case 8 topology. These results suggest that LSN estimation be considered for isosurfaces likely having low occurrences of topologies beneficial to our approach; mathematically defined datasets similar to ones tested here may be good for LSN.

The LSN's orientation vectors can differ substantially from true orientation vectors and from orientation vectors calculated using central-differencing, as demonstrated in Figure 3. We also analyzed the degree each case should be considered "at-risk" of exhibiting errors due to the incorrect topology assumption, focusing on error-prone vertices. Our criterion for this analysis was if angular divergence in the vector was 90 degrees or more from the central-difference orientation vector. We considered only vertices at the midpoint of cell edges. The analysis showed that 146 of the 256 possible MC cases were potentially problematic. One to five vertices demonstrated angular divergence greater than 90 degrees in these cases. Error comparisons of orientation vectors for just the problematic cases are reported in Table 4 over an average of 100 isovalues for each dataset. Our approach produces orientations that are closer to the central-difference than LSN when these cases are encountered. Figure 9 shows isosurface renderings for the Lobster dataset using both approaches. However, the incidence of orientations that meet the angular divergence criterion in sensed and simulation data is likely much smaller since the triangle vertex locations in the analysis were chosen to highlight the problematic cases and the severity in angular difference is lessened when vertices are located closer to grid point locations.

Dataset	Ours	LSN
MRA	0.639	0.651
Foot	0.922	0.933
Frog	0.652	0.689
Lobster	0.479	0.478

Table 4: RMS error for problematic cases

Dataset	Isoval.	# undef.	Total	%
Foot	40	12204	278894	4.36
Frog	40	1263	101841	1.24
Lobster	40	2946	149250	1.97
Engine	40	4704	637854	0.74
Mar-Lobb	0	0	603343	0
Six Peaks	0	8	2004650	0

Table 5: Undefined orientations using LSN approach

Dataset	Ours (secs)	LSN (secs)
Flower	1.077	2.873
Six Peaks	0.926	2.428
Mar-Lobb	2.959	8.018

Table 6: Orientation estimation times

Rendering artifacts at degenerate triangles in the isosurface mesh can be observed in Figure 9(b). They manifest here as dark spots and are a result of using a vector cross product to compute orientation vectors on degenerate triangles in the virtual mesh. (MC produces a triangle with three coincident vertices when a grid value is identical to the isovalue.) Here, the orientation vector computed for this triangle has length equal to zero. The zero-length vector leads to a zero vector for the Phong illumination diffuse and specular components . Our method does not exhibit this phenomenon, as is illustrated in Figure 9(a), since our orientation vector relies on the result of a fitting to four data values within the cell rather than on any mesh triangles.

In Table 5, we show the number of undefined orientation vectors recorded using the LSN estimation. Volumes with 8 bit integers had more undefined orientations than did those with floating point values. Far fewer undefined orientations were present in the syn-

thetic datasets, which all used 32 bit floating point numbers to store the volume's sample values. The number of undefined orientation vectors using LSN estimation appeared to correspond to the data type used to store the volume's data values.

Finally, in Table 6 execution times for calculating orientations for three of the larger datasets are shown. The LSN estimation requires over twice the computation of our approach. The LSN approach is not as fast as ours.

6 CONCLUSION

We have presented a new approach for estimating isosurface orientation vectors on sparse grid datasets. The typical approach for orientation estimations, central-differencing, cannot be used universally in sparse grids due to undefined data at some grid locations. Our approach can produce isosurface orientations anywhere that MC can produce triangles. Further, the approach is not affected by the presence of degenerate triangles, which produce shading errors in other approaches as a result of undefined orientations. Thus, the new approach has certain advantages even over MC's orientation estimation. Our approach has, on average, a smaller RMS error than a competing approach (using the baseline of central-difference estimations) on real world data. For synthetic data, advantages were less clear. Computation times for our approach were markedly faster. Further, the new approach guarantees orientation vectors to be defined at all vertex locations, making it applicable to a wider variety of data.

An area for further investigation is using spline fittings that observe the continuity properties of super splines in producing more accurate orientation estimations. Also, other isosurfacing algorithms could be investigated with our approach to estimate orientations to determine what increases in accuracy and error tolerance occur. Another area of further investigation is removing random data grid values to simulate random sensor failures. Lastly, we will evaluate the impact of increasing noise levels on the new approach's accuracy.

7 REFERENCES

[1] T. Abdoun, V. Bennett, L. Danisch, T. Shantz, and D. Jang. Field installation details of a wireless shape-acceleration array system for geotechnical applications. In *14th Int'l Symp. on: Smart Structures and Materials & Nondestructive Evaluation and Health Monitoring*. Int'l Society for Optics and Photonics, 2007.

[2] G. Awanou and M.-J. Lai. C^1 quintic spline interpolation over tetrahedral partitions. *Approximation Theory X: Wavelets, Splines, and Apps.*, pages 1–16, 2002.

[3] W. Boehm and A. Müller. On de Casteljau's algorithm. *Computer Aided Geometric Design*, 16:587–605, 1999.

[4] C. D. Correa, R. Hero, and K.-L. Ma. A comparison of gradient estimation methods for volume rendering on unstructured meshes. *IEEE Trans. on Visualization and Computer Graphics*, 17(3):305–319, 2011.

[5] P. de Casteljau. Courbes et surfaces á poles. In *Andres Citröen, Automobiles SA*, Paris, 1963.

[6] S. Djurcilov and A. Pang. Visualizing sparse gridded data sets. *IEEE Computer Graphics and Apps.*, 20:52–57, 2000.

[7] G. Farin. *Curves and Surfaces for CAGD*. Morgan Kaufmann, 5 edition, 2001.

[8] Z. Hossain, U. R. Alim, and T. Möller. Toward high-quality gradient estimation on regular lattices. *IEEE Trans. on Visualization and Computer Graphics*, 17(4):426–439, 2011.

[9] L. Neumann, B. Csébfalvi, A. König, and E. Gröller. Gradient estimation in volume data using 4d linear regression. *Computer Graphics Forum*, 19(3):351–358, 2000.

[10] T. Newman and H. Yi. A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5):854–879, October 2006.

[11] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava. Sensor network data fault types. *ACM Trans. on Sensor Networks (TOSN)*, 5(3):25, 2009.

[12] G. M. Nielson, A. Huang, and S. Sylvester. Approximating normals for marching cubes applied to locally supported isosurfaces. In *Proc., Visualization '02*, pages 459–466, 2002.

[13] S. Raman and R. Wenger. Quality isosurface mesh generation using an extended marching cubes lookup table. *Computer Graphics Forum*, 27(3):791–798, May 2008.

[14] C. Rössl, F. Zielfelder, G. Nurnberger, and H.-P. Siedel. Spline approximation of general volumetric data. In *Proc., Symp. on Solid Modeling and Apps.*, pages 71–82, 2004.

[15] K. M. T. Möller, R. Machiraju and R. Yagel. A comparison of normal estimation schemes. In *Proc., Visualization '97*, pages 19–26, 1997.

[16] S. W. Wang and A. E. Kaufman. Volume sampled voxelization of geometric primitives. In *Proc., Visualization '93*, pages 78–84, 1993.

[17] Y. Wang, Y. Liu, and Z. Guo. Three-dimensional ocean sensor networks: a survey. *Journal of Ocean University of China*, 11(4):436–450, 2012.