ISSN 1213-6972 *Volume 22, Number 2, 2014*

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.

EDITOR – IN – CHIEF

Václav Skala

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

Baranoski,G. (Canada)
Benes,B. (United States)
Biri,V. (France)
Bouatouch,K. (France)
Coquillart,S. (France)
Csebfalvi,B. (Hungary)
Cunningham,S. (United States)
Davis,L. (United States)
Debelov,V. (Russia)
Deussen,O. (Germany)
Ferguson,S. (United Kingdom)
Goebel,M. (Germany)
Groeller,E. (Austria)
Chen,M. (United Kingdom)
Chrysanthou,Y. (Cyprus)
Jansen,F. (The Netherlands)
Jorge,J. (Portugal)
Klosowski,J. (United States)
Lee,T. (Taiwan)
Magnor,M. (Germany)
Myszkowski,K. (Germany)
Oliveira,Manuel M. (Brazil)
Pasko,A. (United Kingdom)
Peroche,B. (France)
Puppo,E. (Italy)
Purgathofer,W. (Austria)
Rokita,P. (Poland)
Rosenhahn,B. (Germany)
Rossignac,J. (United States)
Rudomin,I. (Mexico)
Sbert,M. (Spain)
Shamir,A. (Israel)
Schumann,H. (Germany)
Teschner,M. (Germany)
Theoharis,T. (Greece)
Triantafyllidis,G. (Greece)
Veltkamp,R. (Netherlands)
Weiskopf,D. (Germany)
Weiss,G. (Germany)
Wu,S. (Brazil)
Zara,J. (Czech Republic)
Zemcik,P. (Czech Republic)

Muller,Heinrich
Munoz,Adolfo
Murtagh,Fionn
Okabe,Makoto
Oliveira,Joao
Oliveira,Manuel M.
Oyarzun,Cristina Laura
Pan,Rongjiang
Papaioannou,Georgios
Paquette,Eric
Pasko,Galina
Patane,Giuseppe
Patow,Gustavo
Pedrini,Helio
Pereira,Joao Madeiras
Peters,Jorg
Pina,Jose Luis
Platis,Nikos
Post,Frits,H.
Puig,Anna
Puppo,Enrico
Puppo,Enrico
Rafferty,Karen
Raffin,Romain
Renaud,Christophe
Reshetouski,Ilya
Reshetov,Alexander
Ribardiere,Mickael
Ribeiro,Roberto
Richardson,John

Ritter,Marcel
Rojas-Sola,Jose Ignacio
Rokita,Przemyslaw
Rudomin,Isaac
Sacco,Marco
Sadlo,Filip
Salvetti,Ovidio
Sanna,Andrea
Santos,Luis Paulo
Sapidis,Nickolas,S.
Savchenko,Vladimir
Segura,Rafael
Seipel,Stefan
Sellent,Anita
Semwal,Sudhanshu
Sheng,Bin
Sheng,Yu
Shesh,Amit
Schmidt,Johanna
Sik-Lanyi,Cecilia
Sintorn,Erik
Sirakov,Nikolay Metodiev
Sourin,Alexei
Sousa,A.Augusto
Sramek,Milos
Stroud,Ian
Subsol,Gerard
Sundstedt,Veronica
Svoboda,Tomas
Szecsi,Laszlo

Tang,Min
Teschner,Matthias
Theussl,Thomas
Tian,Feng
Tokuta,Alade
Torrens,Francisco
Trapp,Matthias
Tytkowski,Krzysztof
Umlauf,Georg
Vergeest,Joris
Vitulano,Domenico
Vosinakis,Spyros
Walczak,Krzysztof
Wan,Liang
Wu,Shin-Ting
Wuensche,Burkhard,C.
Wuethrich,Charles
Xin,Shi-Qing
Xu,Fei Dai Dongrong
Yoshizawa,Shin
Yue,Yonghao
Yue,Yonghao
Zalik,Borut
Zemcik,Pavel
Zhang,Xinyu
Zhao,Qiang
Zheng,Youyi
Zitova,Barbara
Zwettler,Gerald

Journal of WSCG Vol. 22, 2014

No.2

Contents

A Visuomotor Coordination Model for Obstacle Recognition

Tomoyori Iwao
Waseda University
3-4-1 Okubo, Shinjuku-ku
169-8555, Tokyo, Japan
sazabi@akane.waseda.jp

Hiroyuki Kubo
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma
630-0192, Nara, Japan
hkubo@is.naist.jp

Akinobu Maejima
Waseda University
3-4-1 Okubo, Shinjuku-ku
169-8555, Tokyo, Japan
a.maejima@kurenai.waseda.jp

Shigeo Morishima
Waseda Research Institute for
Science and Engineering
3-4-1 Okubo, Shinjuku-ku
169-8555, Tokyo, Japan
shigeo@waseda.jp

ABSTRACT

In this paper, we propose a novel method for animating CG characters that while walking or running pay heed to obstacles. Here, our primary contribution is to formulate a generic visuomotor coordination model for obstacle recognition with whole body movements. In addition, our model easily generates gaze shifts, which expresses the individuality of characters. Based on experimental evidence, we also incorporate the coordination of eye movements in response to obstacle recognition behavior via simple parameters related to the target position and individuality of the characters's gaze shifts. Our overall model can generate plausible visuomotor coordinated movements in various scenes by manipulating parameters of our proposed functions.

Keywords

visuomotor coordination, obstacle recognition, general versatility

1 INTRODUCTION

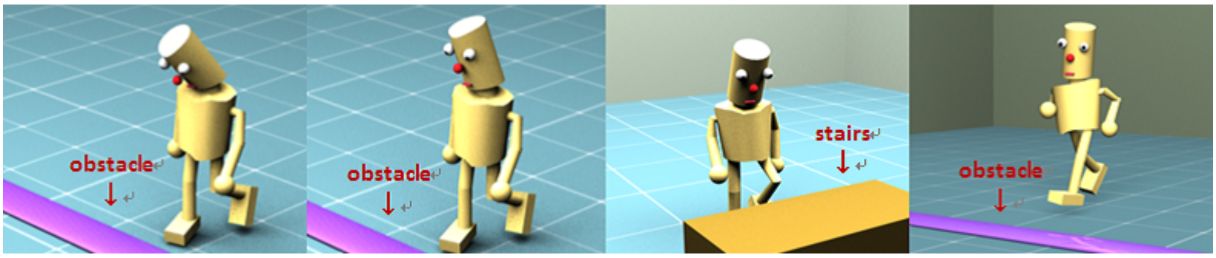
Obstacle recognition is an important aspect of computer-generated (CG) animation, for example, perceiving a puddle of water on the ground while walking or running. In general, this appears to be a simple problem involving characters responses to static or moving obstacles. Such movements are well-researched in the character animation field, however, coordinating eye movements in response to obstacles remains an unaddressed problem. Further, although eye movements have been researched in several areas, such as computer graphics and physiology [Arg76a, Arg65a, Bec89a, Col00a, Den07a, Mat03a], analyzing or generating realistic eye movement is a challenging task. In this paper, we propose a visuomotor coordination model for obstacle recognition that simultaneously measures and analyzes eye and whole body movements. As shown in Figure 1, our model can efficiently generate realistic cognitive movements by designating the obstacle position. Further, our model is artist-friendly in terms of representing characteristic gaze shifts, for example, consistently maintaining a gaze with a human continually looking downward.

Human Walk.

Understanding and modeling human walk is well-researched in several areas, including physiology and robotics. Humans often walk to search for something or to reach their destinations. Similarly, in CG scenes, numerous characters walk for the same purposes. Therefore, modeling actual human walking is crucial to synthesize realistic CG character walking. Particularly, obstacle recognition modeling while walking is significant because CG characters usually observe some object while walking.

To understand how humans recognize obstacles while walking, collecting and studying human data is necessary. Therefore, we observed humans recognizing obstacles while walking and, simultaneously measured the arm, foot, body, and head movements via a motion capture system. In addition, we measured eye move-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.



(a) Walking carefully (b) Walking less carefully (c) Climbing Stairs (d) Running

Figure 1: The character recognizes an obstacle while walking or running; synthesized character animations in various scenes were achieved by manipulating parameters of our model, which can also express the individuality of the character; note that movements for obstacle recognition mainly depends on the obstacle's position.

ments via a head mounted eye tracker. This experimental environment is presented in Figure 2.

To measure how the body and eyes respond to an obstacle, the obstacle should be noticeable and its position must be easily estimated from a human subject's perspective. We placed a black 70cm-squared cloth on the floor as an easily recognizable obstacle. In our experiment, a subject approached the obstacle and passed across it. Next, the subject walked away from the obstacle. The subject observed the obstacle while approaching, on the other hand, did not pay heed to it while passing across it or walking away from it. In particular, we observed that subjects naturally walked away from the obstacle.

Actual measurement results of the arm and foot movements are presented in Figure 3; results indicated that the arm and foot move in a fixed cycle while walking. The arm moves forward and backward, whereas, the foot moves upward and downward. These results are observed in ordinary scenes and are often taken for granted in physiology and robotics. CG artists are aware of these movements and easily synthesize such arm and foot movements into their CG scenes.

As illustrated in Figure 4, the head movement shares a strong relationship with the foot movement. Figure 4 indicates that the head moves in the same cycle as the foot movement cycle. Generally, head movements are divided into two types; translation and rotation. In our experiment, the head translates upward and downward with foot movement. Simultaneously, the head rotates upward and downward to recognize the obstacle on the ground or look ahead carefully.

As illustrated in Figure 5, eye movements also share a strong relationship with foot movements. Figure 5 indicates that the eye moves approximately in the same cycle as the foot. However, unlike head movements, eye movements have a subtle time lag to the foot movement. In general, the eye rotates upward and downward much like the head rotation.

Contributions.

In this paper, we propose a novel visuomotor coordination model for CG characters to recognize an obstacle while walking or running. Our model is based on experimental evidence of actual human behavior. This model has two key features: *visuomotor coordination* based on simultaneous measurements using a motion capture system as well as a head mounted eye tracker and *general versatility* for use in various CG scenes to represent the characteristic behaviors of CG characters. Principally, characteristic behaviors are the eye and body movements of CG characters, including the individuality of such characters; e.g., a character might always look downward while walking or not pay heed to an obstacle while walking. To the best of our knowledge, simultaneous measurements of eye and body movements are rarely conducted in the computer graphics field [Yeo12a]. Further, the visuomotor coordination models proposed before have less versatility, i.e., they can be used only in limited CG scenes with object interception such as ball catch [Yeo12a]. Our model is much easier to manipulate for rigging artists because we model obstacle recognition behavior via a function with a set of simple parameters. Moreover, considering the temporal sequence of foot, head, and eye movements is an effective means to naturally produce obstacle recognition behavior in CG scenes.

The rest of this paper is organized as follows. In Section 2, we introduce related work and compare such work with our approach. In Section 3, we create the practical visuomotor coordination model for obstacle recognition based on experimental evidence. In Section 4, we illustrate the solution to our proposed model equations and the application of our model to various CG scenes; we also present how to manipulate parameters in our model to represent specific characteristic behaviors. In Section 5 and the supplemental video, we demonstrate the performance of our model. We discuss the limitations of our model in Section 6 and conclude our work in Section 7.

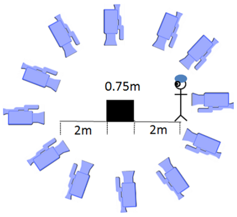
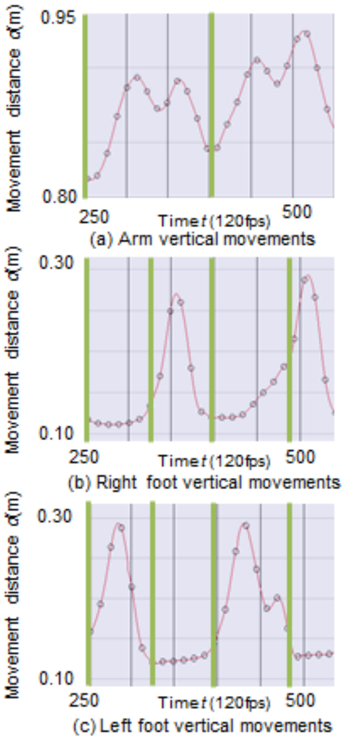


Figure 2: Experimental Environment



(a) Arm vertical movements

(b) Right foot vertical movements

(c) Left foot vertical movements

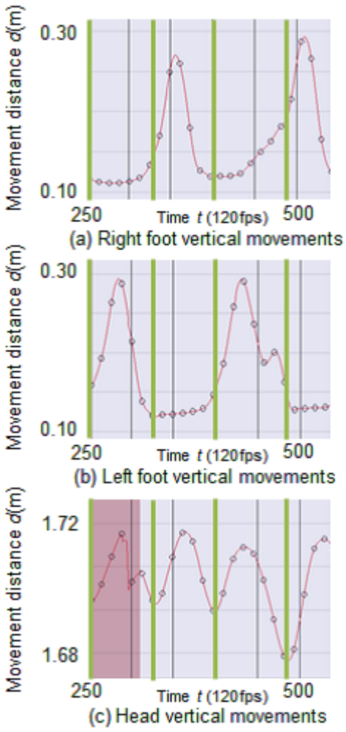
Figure 3: Tracking arm and foot vertical movements

2 RELATED WORK

Computer Graphics.

Numerous papers regarding character animation focus on ways to synthesize a character's body motion based on actual motion capture data or inverse kinematics [Bae04a, Hua11a, Kov02a]. These methods can synthesize various motions, including walking and running, however, such approaches cannot simultaneously generate body and eye movements.

A virtual character's gaze is considered an important part of realistic character animation and crowd simulation [Den07a, Fuk02, Gu07a, Iwa12a, Lan10a, Lee02a, Man11a, Pel03a, Wan02a, Sha05a]. Eye movement is sometimes expressed independently because it has complex facilities or features. However, eye movements are synchronized with head movements and, in general, several researchers consider eye movements to be related to head movements [Itt03a, Ma09a, Mas07a].

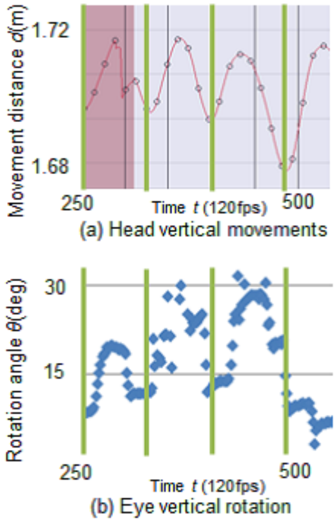


(a) Right foot vertical movements

(b) Left foot vertical movements

(c) Head vertical movements

Figure 4: Tracking head and foot vertical movements



(a) Head vertical movements

(b) Eye vertical rotation

Figure 5: Tracking eye and head vertical movements

Owing to this, in computer graphics, eye movements are typically modeled only according to head movements. However, head and eye movements should be further related to body movements.

To the best of our knowledge, modeling the eye and body movements simultaneously is a challenging task that is rarely performed [Yeo12a]. Yeo et al. proposed the visuomotor coordination model for object interception. However, their model is only applicable to limited situations such as catching a ball. Ball catch scene is appeared less frequently than walking scene we focus on in this paper.

Physiology.

Gaze is well researched in physiology because it plays an important part in human behavior and communication [Evi91a, Ike88a, Ike07a, Ken67a, Kur12a, Mar04a, Mat03a]. Gaze shifts while walking is an interesting topic in physiology in which head and eye movements vary according to foot movements while walking [Blo92a, Gro88a, Gro89a]. For example, the head rotates downward when we bend our knees to walk forward. In spite of these sophisticated works, modeling head and eye movements while walking is still not fully adequate. Much of the work to date cannot be directly applied to three-dimensional character animation.

Robotics.

Visuomotor coordination models for walking or for obstacle recognition have been proposed in the robotics field [Bor95a, Hua01a, Zhe90a]. These models are useful and optimized for robots to recognize and avoid an obstacle rather than for generating realistic human-like motion. Therefore, such models are not used for improving the quality of character animation.

3 MODELING

To establish our model, we measured body movement and eye movements simultaneously and constructed our visuomotor coordination model for obstacle recognition. The relationships between the foot, body, head, and eye movements were crucial in constructing our model. The head and eye movements required to be carefully modeled because these movements were highly related to the obstacle. In this chapter, we first independently describe the head and eye movement models. Next, we illustrate the revised model that combines these movements with corresponding body movements and with the influence of the view angle.

Head Rotation.

We divide head movements into two distinct types of movements; translation and rotation. Because head translation is only provided by foot movement while ordinary walking, head rotation is a significant component in expressing movements involved in obstacle recognition. Figure 6 depicts actual head rotation measurements from our experiment. Note that there are some discontinuities in Figure 6 because we remove outliers which have very high value and can be regarded as error data from the experimental result. The dynamic range is set by measuring maximum and minimum of the head rotation angle after removing error data . Figure 7 $\sim$ 9 are represented in the same manner as Figure 6.

We identified a few key features of head rotation from these measurements. An important feature was the rotating cycle; the head rotated in the same cycle as foot movements. Specifically, the head rotated upward when the foot moved upward and vice versa. The trajectory of the rotation angle is also important; the peak of rotation angle was observed in the middle of a cycle. In other words, the time for upward head rotation was equal to the time for downward head rotation. Moreover, the head was directed toward the front for a longer period of time than downward. Considering the above features and the actual trajectory of the rotation angle, we approximated the trajectory in a cycle by a Gaussian function as follows.

$$\theta_H(t) = \theta_{Hmax} \exp\left\{-\frac{(t-\mu)^2}{2\sigma_H^2}\right\} \tag{1}$$

Here, θ_{Hmax} is a value on the peak of a Gaussian function and, σ_H^2 is the variance of the head rotation representing the initial head rotation angle. Cyclic functions are usually used for approximating periodical movement like walking. However, we found that obstacle direction from subjects varies according to the time course. Thus we need to control the obstacle direction in each cycle. Although cyclic functions have an advantage of manipulating whole movement with a few parameters, we use Gaussian function in this paper to manipulate parameters in each cycle.

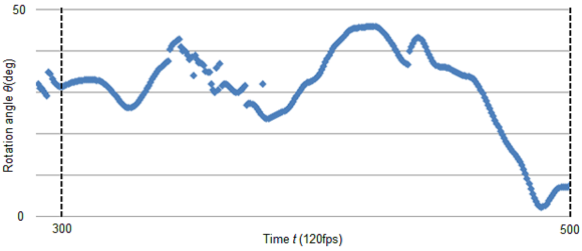


Figure 6: Actual head rotation measurements during obstacle recognition

Eye Rotation.

Eye translation is easily described and followed by head translation, therefore, we only considered the eye rotation. Actual eye rotation measurements from our experiments are presented in Figure 7.

We identified a few features of eye rotation that were similar to that of head rotation. Compared with foot movements, eyes rotated in nearly the same cycle. However, unlike the head rotation, eye rotation included a subtle time lag. As to the trajectory of the eye rotation angle, the shape of the trajectory was similar to that of head rotation. Much like the head rotation, we approximated the trajectory in a cycle via a Gaussian function as follows.

$$\theta_E(t) = \theta_{Emax} \exp\left\{-\frac{(t-(\mu+t_E))^2}{2\sigma_E^2}\right\} \tag{2}$$

Here, θ_{Emax} is a value on the peak of a Gaussian function and σ_E^2 is the variance of the eye rotation representing the initial eye rotation angle.

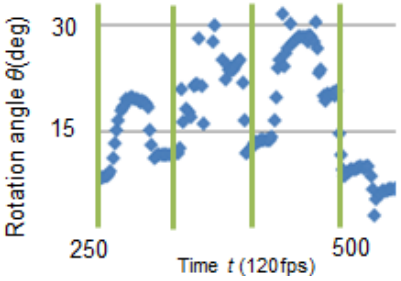


Figure 7: Actual eye rotation measurements during obstacle recognition

Model for Obstacle Recognition.

We approximated each rotation of the head and eye in a cycle via a Gaussian function. Obstacle recognition was highly related to the head and eye rotation. However, body rotation angle and view angle should also be considered for three-dimensional character animation.

Body rotation while walking is easily provided by rigging artists or approximately constant in a short period. It influences a character's direction of gaze. Further, view angle is important to control gaze shifts, especially when humans recognize an object quickly, in which case they tend to look at an object indirectly in the range of their view angle. Because subjects in our measurement paid attention to what was in front of them, i.e., the obstacle, they looked at it indirectly. We combined body rotation, head rotation, eye rotation, and view angle to form our final work; thus, the model for obstacle recognition angle θ_G is described as follows.

$$\theta_G(t) = \theta_H(t) + \theta_E(t) + \theta_B(t) + \theta_V(t) \qquad (3)$$

Therefore, we have the following.

$$\theta_G(t) = \theta_{Hmax} \exp\left\{-\frac{(t-\mu)^2}{2\sigma_H^2}\right\} + \theta_{Emax} \exp\left\{-\frac{(t-(\mu+t_E))^2}{2\sigma_E^2}\right\} + \theta_B(t) + \theta_{Vmax} \qquad (4)$$

We incorporated additional features from our observations into the above equation. First, the head rotated with constant rate α against the eye rotation. Second, the energy and burden for head and eye movements increased while humans approached the obstacle because θ_G increased. We assumed that humans use a broad view angle to reduce the burden when they look at a nearby obstacle. By the same reasoning, a broad view angle is used when the cycle is short. Therefore, the final model is described as follows.

$$\theta_G(t) = \theta_{Hmax} \exp\left\{-\frac{(t-\mu)^2}{2\sigma_H^2}\right\} + \theta_{Emax} \exp\left\{-\frac{(t-(\mu+t_E))^2}{2\sigma_E^2}\right\} + \theta_B(t) + (\theta_{Vmax} - \frac{l(t)}{\beta}) \qquad (5)$$

$$\theta_{Hmax} = \alpha \theta_{Emax} \qquad (6)$$

4 GENERATING CHARACTER ANIMATIONS

Based on an obstacle's position, our proposed model can generate character animations in various scenes. Our model can also express characteristic behaviors, such as looking downward while walking. In this chapter, we describe how to use our model to generate character animations.

4.1 Solving the model equation

Our proposed equation can be solved in a simple manner. Described above, parameter θ_B is provided by artists or set as a constant. To generate character animation, we require to obtain the trajectory of θ_H and θ_E. From Equation (5) and Equation (6), we have the following.

$$\theta_{Hmax} \exp\left\{-\frac{(t-\mu)^2}{2\sigma_H^2}\right\} + \alpha \theta_{Hmax} \exp\left\{-\frac{(t-(\mu+t_E))^2}{2\sigma_E^2}\right\} = \theta_G(t) - \theta_B(t) - (\theta_{Vmax} - \frac{l(t)}{\beta}) \qquad (7)$$

Because θ_{Hmax} is a value on the peak of a Gaussian function, we substitute $t = \mu$ $(= \frac{T_F}{2})$. Further, because time lag t_E is very short, we opt to ignore it; $t_E \fallingdotseq 0$; therefore, we have the following.

$$\theta_{Hmax} = \frac{\theta_G(\frac{T_F}{2}) - \theta_B(\frac{T_F}{2}) - (\theta_{Vmax} - \frac{l(\frac{T_F}{2})}{\beta})}{1 + \frac{1}{\alpha}} \qquad (8)$$

Finally, we obtain trajectory θ_H, θ_E as follows.

$$\theta_H(t) = \frac{\theta_G(\frac{T_F}{2}) - \theta_B(\frac{T_F}{2}) - (\theta_{Vmax} - \frac{l(\frac{T_F}{2})}{\beta})}{1 + \frac{1}{\alpha}} \exp\left\{-\frac{(t-\mu)^2}{2\sigma_H^2}\right\} \qquad (9)$$

$$\theta_E(t) = \frac{\theta_G(\frac{T_F}{2}) - \theta_B(\frac{T_F}{2}) - (\theta_{Vmax} - \frac{l(\frac{T_F}{2})}{\beta})}{1 + \alpha} \exp\left\{-\frac{(t-(\mu+t_E))^2}{2\sigma_E^2}\right\} \qquad (10)$$

4.2 Obstacle recognition in various scenes

4.2.1 *Walking*

Obstacle recognition while walking is the same scene as our measurement environment. Generating the animation is straightforward in this case, because obstacle recognition angle θ_G is represented by the following simple manner. In this scene, θ_G is represented as

$arctan(\frac{h}{l})$ and the parameter h represents the height of the character. We substitute θ_G and can obtain the trajectory θ_H and θ_E via Equation (9) and Equation (10).

4.2.2 Climbing stairs

A character watches his steps when climbing stairs. This situation seems complicated, however, our model simply generates the animation according to obstacle recognition direction θ_G. In this scene, when a character is climbing a given step, we assume that the character is watching the next step and treat it as an obstacle position in a cycle. The character animation for climbing stairs is generated because θ_G in each cycle is determined.

4.2.3 Running

Like walking, obstacle recognition while running is similar to our measurement environment. The character runs to the obstacle, passes over it, and then runs away from it. The scene is generated by merely manipulating θ_{Vmax} depending on the head or eye rotation cycle times. In general, the vertical view angle reaches 60～80 ° . This angle includes both the upward angle and downward angles. In this paper, because the obstacle is on the ground, we only consider the downward view angle and set θ_{Vmax} to 20° while walking and 30° while running.

4.2.4 Characteristic Behavior

Characteristic behavior is expressed by manipulating parameters α, σ_H and σ_E. Specifically α represents the degree of carefulness; the character walks more carefully when α is increased. Further, σ represents the individuality of the character; for example, when σ is increased, the character is more apt to look downward while walking or running.

5 RESULTS

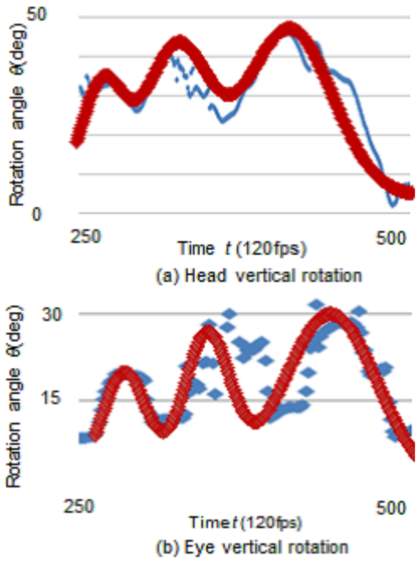
Using our proposed model, we generated character animations as shown in Figure 1 and the supplemental video. By merely designating the obstacle recognition position, we were able to express character animations in various situations, including walking (Figure 1(a) and Figure 1 (b)), climbing stairs (Figure 1(c)), and running (Figure 1(d)). In addition, we generated characteristic behaviors, e.g., looking downward while walking by changing parameter values of our model.

Validation.

To verify whether our model was able to express human-like movements, we compared the results of our simulation with actual measurement. We recorded the movements of the body by using a 13-camera Vicon motion capture system; further, we recorded eye movements via a head mounted EMR-9 eye-tracker. The motion capture system recorded whole body movements at 120 Hz, and the eye-tracker recorded eye movements at 60 Hz, which was sufficient for our purpose. In our measurements, subjects first approached the obstacle, passed over it, then walked away from it, as described in Chapter 1 above. Figure 9 compares actual trajectories of head and eye rotation angles with trajectories created via our model. Qualitatively, they look similar and we therefore conclude that our model can express a variety of characteristic behaviors in the experimental environment.

Generalization.

Our model can synthesize character animations for obstacle recognition in various CG scenes, including stair-climbing and walking on the uneven terrain. By changing obstacle recognition positions or parameter values in our model, various character animations can be successfully generated. We captured a subject climbing two steps to evaluate the different movements. Figure 9 compares actual trajectories of head and eye rotation angles with trajectories created via our model. Qualitatively, these results look similar to one another. Therefore, we conclude that our model can be applied to various movements and generate realistic corresponding character animations.



(a) Head vertical rotation

(b) Eye vertical rotation

Figure 8: Comparing trajectories of eye and head rotation angles while walking; actual trajectories (blue), trajectories created via our model(red).

6 LIMITATIONS

Our model has a few limitations. We primarily focus on the relationships between the foot, body, head and eye movements, and model head and eye rotations for rigging artists. Therefore, our model is indeed practical for generating character animations. However, our

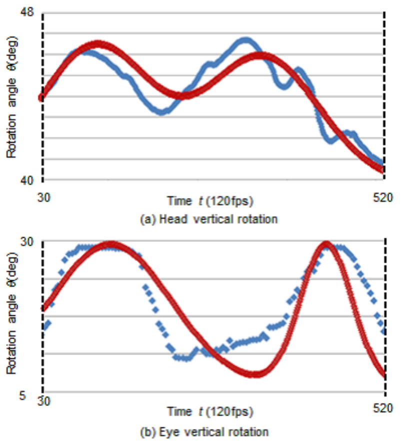


Figure 9: Comparing trajectories of eye and head rotation angles while stair-climbing; actual trajectories (blue), trajectories created via our model(red).

model is rather simplified. We modeled head and eye rotations in a cycle by using a Gaussian function. Other functions, such as a quadratic function or a trigonometric function, may better approximate head and eye rotations. Further, the frequency at which the head and eyes rotate should be probabilistically controlled. Probabilistic control would determine whether the person looked at an obstacle in a cycle and it will generate more sophisticated human-like movements for obstacle recognition. Finally, human and CG character responses to multiple obstacles is an interesting topic. To express character animations that can respond to multiple obstacles, a saliency map method would be a representative approach. Saliency maps are often used in image recognition and gaze control for CG characters because saliency is also useful in computer graphics. Combining our model with the saliency map would enable us to express more complex movements.

7 CONCLUSIONS

In this paper, we proposed a novel visuomotor coordination model for animated CG characters walking or running while paying heed to obstacles. Our model proved the capability to generate realistic human-like motion for obstacle recognition based on simultaneous measurements of the entire body and eye movements, an approach that has rarely been undertaken, especially in the field of computer graphics. Movements in various CG scenes are represented via our model. Characteristic behaviors can be expressed by changing parameter values of our model. Further, our model is easily used by rigging artists to manipulate obstacle recognition points and other parameters.

Although we identified a few limitations presented in Section 6, our model is a novel approach to character animation. Briefly, our proposed method generated plausible character animations and is applicable to various CG scenes. Our proposed method further improves the quality of current character animation and, will be useful for further research regarding visuomotor coordination in the future.

8 REFERENCES

[Arg76a] Argyle, M., and Cook, M., ”Gaze and Mutual Gaze.” Cambridge University Press, London, 1976

[Arg65a] Argyle, M., and Dean, J., "Eye-Contact, Distance and Affiliation", Sociometry, vol 28, pp.289-304, 1965

[Bae04a] Baerlocher P., and Boulic R., “An inverse kinematics architecture enforcing an arbitrary number of strict priority levels”, The Visual Computer, Springer Verlag, 20(6), pp.402-417, 2004

[Bec89a] Becker, W., "The Neurobiology of Saccadeic Eye Movements. Metrics", In:RH Wurtz and ME Goldberg(eds), Elsevier Science Publishers BV (Biomedical Division), New York, NY ch.2, pp 13-67, 1989

[Ber07a] Bernhardt, S., and Robindon, P., "Detecting affect from non-stylised body motions.", Affective Computing and Intelligent Interaction, Second International Conference, AC2 2007, Lisbon, Portugal, September 12-14, Proceedings, volume 4738 of LNCS, pp 59-70, 2007

[Blo92a] Bloomberg, J. J., Reschke, M. F., Huebner, W. P., and Peters, B. T., "The effects of target distance on eye and head movement during locomotion.", Annals of the New York Academy of Sciences, vol.656 issue 1, pp.699-702, 1992

[Bor95a] Borenstein, Johann., and Koren, Y., "Error Eliminating Rapid Ultrasonic Firing for Mobile Robot Obstacle Avoidence", IEEE Transactions on Robotics and Automation, vol.11, no.1, pp.132-138, 1995

[Col00a] Colburn, A., Cohen, M. F., and Drucker, S. M., “The role of eye gaze in avatar mediated conversational interfaces.” Microsoft Tech Report 2000-81

[Den07a] Deng, Z., Lewis, J. P., and Neumann, U., ”Realistic Eye Motion Synthesis by Texture Synthesis”, Data-Driven 3D Facial animation, pp.98-112, December 11, 2007

[Evi91a] Evinger, C., Manning, K. A., and Sibony P. A., ”Eyelid Movements Mechanisms and Normal Data”, Investigate Ophthalmology and Visual Science, vol.32, no.2, February 1991

[Fuk02] Fukayama, A., Ohno, T., Mukayama, N., Sawaki, M., and Hagita, N., "Messages Embedded in Gaze of Interface Agents -Impression

Management with Agent's Gaze-", CHI'02 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp.41-48, 2002

[Gro88a] Grossman, G. E., Leigh, R. J., Abel, L. A., Lanska, D. J., and Thurston, S. E., "Frequency and velocity of rotational head perturbations during locomotion.", Experimental brain research, vol.70, pp.470-476, 1998

[Gro89a] Grossman, G. E., Peters, B. T., Smith, S. L., Huebner, W. P., and Lanska, D. J., "Performance of the human vestibuloocular reflex during locomotion.", Journal of Neurophysiology, vo.62, pp 264-272, 1989

[Gu07a] Gu, E., Lee, S. P., Badler, J. B., and Badler, N. I., ”Eye Movements, Saccades, and Multiparty conversations”, Data-Driven 3D Facial animation, pp.79-97, December 11, 2007

[Hua11a] Huang, H., Chai, J., Tong, X., and Wu, H. T., "Leveraging motion capture and 3D scanning for high-fidelity facial performance acquisition", ACM Transactions on Graphics, vol.30 issue 4, no.74, July 2011

[Hua01a] Huang, Q., Yokoi, K., Kajita, S., and Kaneko, K., "Planning Walking Patterns for a Biped Robot", IEEE Transactions on Robotics and Automation, vol.17, no.3, June 2001

[Ike88a] Ikeda, M., “*Meha Naniwo Miteiruka?* [What do eyes look at?]”,Heibonsha,1988

[Ike07a] Ikegami, K., and Kita, Y., ”Interpersonal distance as a function of sex, age, attraction and intimacy”,Bulletin of the Faculty of Education, Kanazawa University. Educational science 56, pp.1-12, 2007

[Itt03a] Itti, L., "Realistic Avatar Eye and Head Animation Using a Neurobiological Model of Visual Attention", Tech.rep., DTI Document, 2003

[Iwa12a] Iwao, T., Mima, D., Kubo, H., Maejima, A., and Morishima S., ”Analysis and Synthesis of Realistic Eye Movement in Face-to-face Communication”, SIGGRAPH'12 ACM SIGGRAPH 2012 Posters Article No. 87, 2012

[Ken67a] Kendon, A., “Some functions of gaze direction in social interaction” Acta Psychologica,26: pp.22-63, 1967

[Kov02a] Kovar, L., Gleicher, M., and Pighin, F., “Motion graphs”, In Computer Graphics, SIGGRAPH Proceedings, 2002

[Kur12a] Kurihara, D., Honma, M., and Osada, Y., ”Relation between the range of eye contact and individual characteristics in a face-to-face situation”, Rikkyo University, Psychology vol.54,pp.51-58, 2012

[Lan10a] Lance, B. J. and Marsella, S. C., "The Expressive Gaze Model: Using Gaze to Express Emotion", Computer Graphics and Applications, IEEE, vol.30 issue 4, pp.62-73, 2010

[Lee02a] Lee, S. P., Badler, J. B., and Badler, N. I., "Eyes Alive", ACM Transactions on Graphics(TOG)- Proceedings of ACM SIGGRAPH, vol.21 issue 3, pp637-644, July 2002

[Ma09a] Ma, X., and Deng, Z., "natural Eye Motion Synthesis by Modeling Gaze-Head Coupling", VR'09 Proceedings of the 2009 IEEE Virtual Reality Conference, pp.143-150, 2009

[Man11a] Mancini, M., Castellano, G., and Peters, C. , "Evaluating the Communication of Emotion via Expressive Gesture Copying Behaviour in an Embodied Humanoid Agent", Affective Computing and Intelligent Interaction Lecture Notes in Computer ScienceVolume 6974, pp.215-224, 2011

[Mar04a] Martinez-Conde, S., Stephen, L., and Hubel, D. H., "The role of fixational eye movements in visual perception", Nature Reviews Neuroscience 5, pp.229-240, 2004

[Mas07a] Masuko, S., and Hoshino, J., "Head-eye animation corresponding to a conversation for CG characters", Computer Graphics Forum, Journal of the European Association for Computer Graphics, vol.26, no.3, pp.303-311, 2007

[Mat03a] Mathews, A., Fox, E., Yield, J. and Calder, A., "The face of fear:Effects of Eye Gaze and Emotion on Visual Attention", Visual Cognition, 10:7, pp.823-835, 2003

[Sha05a] Shao, W., and Terzopoulos, D., "Autonomous pedestrians.", In Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp.19-28, 2005

[Pel03a] Pelachaud, C., and Bilvi, M., "Modeling Gaze Behavior for Conversational Agents", Intelligent Virtual Agents, Springer, pp.93-100, 2003

[Wan02a] Wang, J. G., and Sung, E., "Study on Eye Gaze Estimation", IEEE Transaction on Systems,Man and Cybernetics-Part B: Cybernetics, vol.32 issue 3, pp.332-350, 2002

[Yeo12a] Yeo, S. H., Lesmana, M., Neog, D. R., and Pai, D. K., "Eyecatch: Simulating Visuomotor Coordination for Object Interception", ACM Transactions on Graphics, vol.31 issue 4, no.42, July 2012

[Zhe90a] Zheng, Y., F. and Shen, J.,“Gait synthesis for the SD-2 biped robot to climb sloping surface,” IEEE Trans. Robot. Automat., vol. 6, pp.86-96, 1990

Improved Particle-based Ice Melting Simulation with SPH Air Model

Jakub Domaradzki
Institute of Computer Science
Warsaw University of Technology
ul. Nowowiejska 15/19
00-661 Warsaw, Poland
J.Domaradzki@stud.elka.pw.edu.pl

Tomasz Martyn
Institute of Computer Science
Warsaw University of Technology
ul. Nowowiejska 15/19
00-661 Warsaw, Poland
T. Martyn@ii.pw.edu.pl

ABSTRACT

This paper presents an improved method for simulating melting of ice. The melting process is implemented as a result of the heat transfer between ice objects and fluids (water and air). Both the solids and the fluids, including air, are modeled as a set of particles with specified temperatures, which can vary locally during simulation. The proposed new particle-based air model allows one to consider in simulation the influence of the natural air convection on the ice melting process. Moreover, the model makes it possible to melt the ice object in a controllable way by means of external heat sources. The motion of air and water, originally described by the Navier-Stokes equations for incompressible fluids, is computed using the Smoothed Particle Hydrodynamics (SPH) algorithm, which we modify to properly handle our particle-based air and its interactions with ice and water. Thanks to a GPU-based implementation, the proposed method allows us to run the simulation of ice melting at interactive speed on an average PC.

Keywords

SPH, ice melting, natural air convection, interactive techniques

1. INTRODUCTION

Simulations of natural phenomena are widely used both in science (physical simulation) and entertainment industry (special effects in movies and computer games). Therefore there is a need for simulation techniques resulting in physically correct and, at the same time, visually attractive outcomes. How to combine the two aspects within efficient and robust simulation is still an active research area in computer graphics. It seems that over the last few years, ice melting is one of the natural phenomena that caught the special attention of computer graphics community.

In this paper we focus on simulation of ice melting regarded as a result of the interactions between ice, melted water, and air. Although a number of efficient methods for simulation of melting ice have been proposed, the majority of them neglect the influence of the natural air convection [PPLT06] or significantly simplify it using heuristic functions [IUDN10]. On the other hand, the more accurate approaches that consider ice-air interactions in the melting process use computationally expensive techniques [FM07] and hence they are far from “interactive-time” methods.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

In general, the widespread methods of ice melting simulation can be divided into two categories: the grid-based approaches and the particle-based approaches. The methods in the first category represent a modeled physical system as a 3D uniform grid of voxel cells. The cells remain static during simulation and store local physical quantities of the system, and calculations are done between neighboring cells [FM07]. The main issue with this approach is that it is difficult to handle details that are essential not only for the final visual appearance but also play important role in the melting process itself (e.g. droplets of melting water on the ice surface).

The methods from the second category rely on discretization of objects into particles. From computational point of view, particles are utilized in a similar way as cells. However, in opposite to cells, particles can move freely. As a consequence, usually with lesser storage and computational requirements, tiny aspects of phenomenon, such as the mentioned water droplets, can be involved in simulation [IUDN10].

2. OUR CONTRIBUTION

Our main goal was to develop a method for ice melting simulation which would consider the effects of the natural air convection in ice-air interactions and, at the same time, could be run at interactive speed. As a result of our work we propose in this paper a new method for ice melting simulation, in which ice (and possibly other solids, e.g. glass) as well as fluids (air and water) involved in the heat transfer, are represented by sets of particles. Such a representation seems to be a natural extension of the previous particle-based ice melting approaches and allows one to consider the fine-grained effects of the natural air convection in simulation. The movement of air particles caused by local changes in temperature and their interactions with particles representing other objects are computed with the aid of a variant of the Smoothed Particle Hydrodynamics (SPH) algorithm. As a consequence, our method combines advantages of the voxel-based and particle-based approaches in that it accurately computes air-ice interactions and, at the same time, considers fine-grained details such as droplets of melting water. Moreover, the proposed approach allows us to augment the ice melting simulation by the possibility of influencing the temperature and, thus, movement of air particles with the aid of external heat sources. We propose a new external heat source model, which acts similarly to hair dryer and allows one to melt ice object in a controllable way.

3. RELATED WORK

One of the early attempts to simulate ice melting was a method by Fujishiro and Aoki [FA01] in which morphology operations and form factors were utilized. The computation related to heat transfer and melting was based on voxels and didn't take into account generation of water due to the phase transition. To deal with this challenge, Carlson et al. [CMIT02] treated solids as high viscosity fluids and based their simulation on solving the Navier-Stokes equations. The phase transition was realized by influencing on the fluids viscosity with respect to temperature changes. Nevertheless, due to the low viscosity of water, the method cannot be used to simulate the flows of melted water. In turn, Matsumura et al. [MT05] used one of the grid-based techniques (MAC method) and simulated the melting of ice including the natural convection of surrounding air. However, due to the rough and static representation of objects with the grid of voxels, the simulation did not handle the fine-grained effects of the ice melting process (such as droplets of melting water). A different model of air founded on a voxel grid and utilizing a technique analogous to photon mapping for calculating thermal radiation was presented by Fujisawa et al. [FM07]. While the results of the method are quite satisfying, its computational cost is high and interactive simulations are impossible. In the context of the method presented in this paper, the most relevant method was presented by Iwasaki et al. [IUDN10]. The solution is based on particles and takes into account many aspects of the phenomenon, in particular visualization of tiny elements, such as water droplets. However, the influence of the air surrounding the ice object was simplified by applying a constant ambient temperature. The heat energy transferred to the ice surface was brought to a heuristic function which depends on an area exposed to air influence.

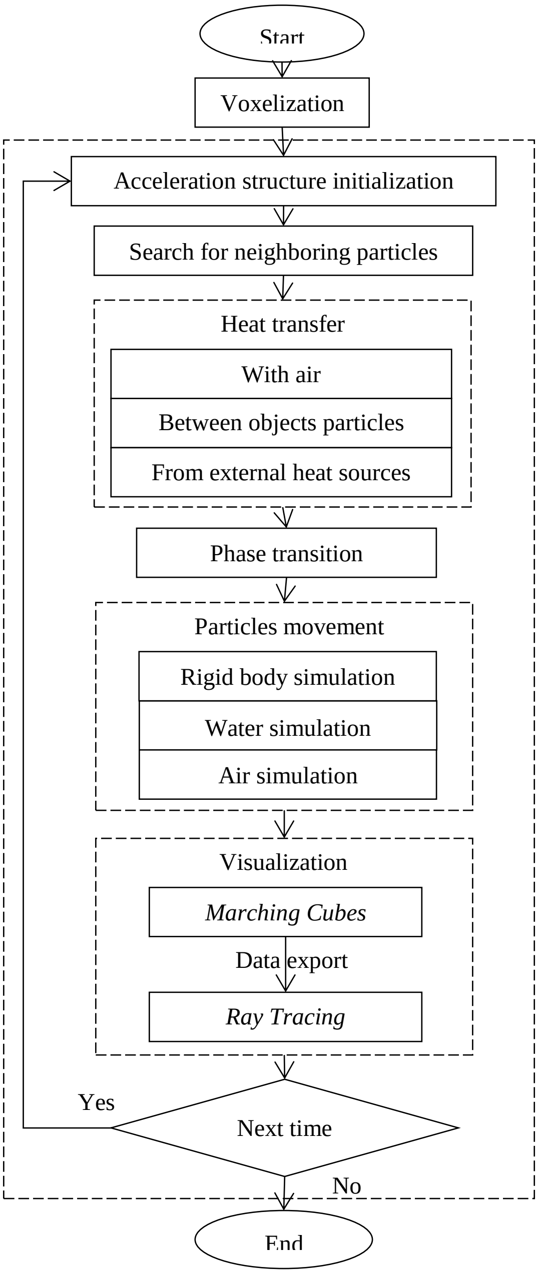


Figure 1. Flowchart of ice melting simulator

To our knowledge, there is no research on representing air with particles for the purpose of the ice-melting simulation. On the other hand, Müller et al.

[MSKG05] showed how to handle multiple fluids with different rest densities. One of the exemplary results of their method were air bubbles rising in the water. However, as shown in [SP08], miscible fluids with a density ratio larger than 10 cannot be realistically simulated with the standard SPH algorithm. Therefore they proposed a different density model, in which all neighbor particles are treated as if they belong to the same fluid. Although densities in the interface are computed correctly, but the structure created by particles cannot be broken to small volumes to simulate, for instance, water droplets. Recently some research on multifluid dynamics systems with an interesting method of density calculation has been presented in [OCD13].

4. OVERVIEW OF THE PROPOSED METHOD

An overview of our simulation method is presented in Fig. 1. There are distinguished operations that are repeated for every time step. From the standpoint of the simulation execution time, the most important part is the initialization of an acceleration structure. A commonly used uniform grid [G10] was chosen, due to preferable local region of interest. Subsequently, all computations are performed on particles on the basis of the data “carried” by neighboring particles of a given particle. Then, the heat transfer between particles is calculated. Depending on their temperatures, this may lead to the phase transition of some ice particles into water particles. Next, the movement of particles is computed, considering the motion of melted water and surrounding air, as well as the ice object collisions and its stability under gravitation. The final stage is visualization. The particles are treated as metaballs and surfaces of visible objects are reconstructed with the use of the marching cubes algorithm [LC87], and a realistic image of the current scene are obtained with ray tracing.

Attribute	Description
M	mass
r	position
u	velocity
ρ_0	rest density
ρ	current density
μ	viscosity coefficient
T	temperature
K	gas constant (stiffness)

Table 1. Particle attributes

5. PARTICLE-BASED ICE MELTING SIMULATION

In our method of ice melting simulation each physical object is modeled using a set of moveable particles. Each particle stores a collection of attributes (Table 1). Their values specify physical quantities of the volume represented by the particle and are updated at every simulation step.

5.1. Heat transfer

The heat is transferred between particles of all media: ice and its surroundings. The increase in temperature can be calculated using the equation [MSKG05]:

$$\frac{\partial T_i}{\partial t} = \alpha \sum_{j \in N_i} m_j \frac{\left(T_j - T_i\right)}{\rho_j} \nabla^2 W(r_{ij}, h_H), \quad (1)$$

where α is the thermal diffusion constant, t is the time, N_i is the set of particles whose distances are smaller than h_H from particle i, W is a smoothing kernel, and r_{ij} is the distance vector $r_i - r_j$.

Other forms of heat transfer are described in Sec. 6 and 7.

5.2. Water simulation

In order to simulate water, we used technique called Smoothed Particle Hydrodynamics (SPH) [MCG03], which solves the Navier-Stokes equations for incompressible fluids. The SPH is based on the assumption that we can distinct certain forces acting on water particles, namely the pressure force f_{press}, the viscosity force f_{vis}, and external forces.

The pressure force is trying to keep the fluid in the incompressible state and is described by the equation:

$$f_{press} = -\sum_{j \in N_i} \frac{p_i + p_j}{2} \frac{m_j}{\rho_j} \nabla W\left(r_{ij}, h_F\right), \quad (2)$$

where p_i is the current pressure for particle i, and is calculated as:

$$p = k(\rho - \rho_0). \quad (3)$$

The viscosity force is used as an internal friction between particles. It is computed using the equation:

$$f_{vis} = \frac{\mu}{\rho_j} \sum_{j \in N_i} u_{ji} m_j \nabla^2 W(r_{ij}, h_F), \quad (4)$$

where u_{ji} is the velocity difference.

The external forces include all other forces acting on fluid, such as gravity, buoyancy, and the interfacial tension which was used in [IUDN10] to create water droplets.

All the mentioned forces result in particle acceleration that is integrated using the Leap-Frog scheme.

The SPH algorithm is also used to simulate the motion of the air particles, but in a slightly different manner (Sec. 6).

5.3. Ice simulation as rigid body

An ice object is represented by a set of particles, and its movement depends on forces acting on those particles. These include the forces generated by water particles and the forces resulting from objects collisions (the repulsive, the frictional, and the damping force). To compute the force interaction between ice and water (e.g., to simulate ice floating on water) we treat an ice particle as if it were a water particle. The total force acting on the ice object is the sum of forces coming from all its particles [TSK07].

6. SPH AIR MODEL

In this section we present a new air model based on the SPH technique. Our model takes advantage of the results from [MSKG05] on simulating air bubbles and handling multiphase environment in SPH.

6.1. Air simulation

The air surrounding an ice object is built from particles and like in the case of water, the SPH algorithm is used to simulate their motion. One should note, however, that we cannot use the basic form of SPH for this purpose. The main problem is that SPH has to handle particles representing physical objects with very different densities and viscosities[1]. In order to keep the simulation stable and produce realistic results, some changes must be made in the standard version of the algorithm.

First of all, in the spirit of [MSKG05], we need to average viscosity coefficients in the viscosity forces. This results in the new formulation of Eq. 4:

$$f_{vis} = \sum_{j \in N_i} \frac{\mu_i + \mu_j}{2} \frac{u_{ji}}{\rho_j} m_j \nabla^2 W(r_{ij}, h_F). \qquad (5)$$

This way the viscosity force will act differently in the interface area and, at the same time, remain unchanged for particles of the same sort.

Furthermore, as it was proved in [SP08], SPH technique does not operate well in the case when the density ratio of different fluids is larger than 10. Unfortunately, we cannot take advantage of the solution proposed in that paper, as we would like to take into account water droplets phenomenon. As a result, we just set the density of the air particles to 10 times smaller than the water density.

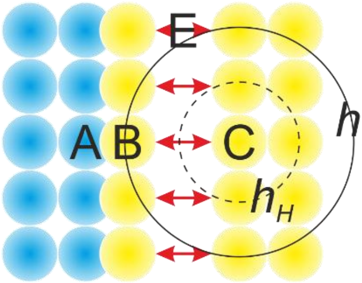


Figure 2. Thin layer of air particles is attracted to ice surface

Additionally, we have to resolve the problem of spurious tension. As shown in Fig. 2, it is present in the interface between the fluids and its variation between ice and air, where a thin layer B of air particles is attracted to the ice surface A. The reason for this is the pressure force in the SPH algorithm. The air particles move from high to low dense areas on the basis of their current density value. When an air particle with a low rest density interacts with an ice particle whose mass is relatively very high, the air particle density exceeds its rest density very quickly. Nevertheless, in such a situation the ice particle and its relation between the current density and the rest density are involved in computation of the pressure force (Eq. 2). The ice particles are in a constant relation to the other particles that belong to the same ice object, and the particles on the boundary do not have enough amount of ice particle neighbors for their densities to reach the rest density of ice. As a result, some air particles are drawn into the ice surface to increase the density of ice particles. On the other hand, the remaining air particles cannot approach to the air particles on the ice surface due to their high calculated densities caused by the ice particles located nearby.

The mentioned situation is especially important, because such a free space E prevents the air particles B and C from exchanging their temperatures (Fig. 2), due to the shorter smoothing radius h_H of the kernel in Eq. 1 than the smoothing radius h_F in Eq. 2.

At first glance, a simple solution to this problem is to shrink the smoothing radius h_F (Fig. 2) of the pressure force equation (Eq. 2). However, according to [K06], such a short radius would not be usable with fluid simulation. Taking everything into consideration, we propose a different solution in which the pressures of particles located on the interface between different mediums are computed as:

$$p = \max(k(\rho - \rho_0), 0), \qquad (6)$$

rather than using the original equation (Eq. 3). Such a modification prevents the air particles from attracting to the ice surface by introducing a slight disturbance on the interface (Fig. 3). As a consequence, a relatively small time step is required to keep the simulation stable.

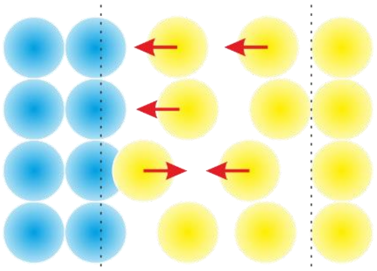


Figure 3. Disturbance on interface between air and other mediums

[1] One should note that ice as a solid has no viscosity in a physical sense. Nevertheless, for the purpose of the SPH simulation of ice-water interactions, the ice particles are usually assigned the viscosity coefficient of water (see e.g. [TSK07]).

6.2. Heat transfer

Thanks to the particle structure of our air model, the air surrounding an ice object can be quite naturally incorporated into the heat transfer process described by the equation (1) (see Fig. 4). In addition, the model provides, without any additional cost, local changes in air temperature.

6.3. Natural air convection

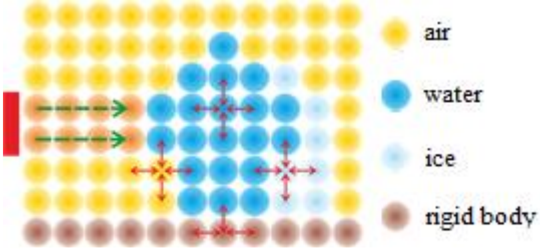


Figure 4. Heat transfer between particles of different mediums

As it was stated at the beginning, our goal is to achieve the simulation of ice as much realistic as it can be. According to laws of physics, volumes of fluid of different temperature move relative to each other due to the differences in their densities – the phenomenon called convection takes place. The proposed particle-based air model is well suited to enrich the simulation of ice melting with the influence of air convection, however, we must slightly alter the SPH algorithm.

First of all, we need to influence on rest densities of air depending on particles temperatures (similarly to [MSKG05]). We want the hot parts of air to transfer to the top areas and cold downwards. To achieve this, we propose the following equation:

$$\rho_0(T,\theta) = \frac{\rho_0}{1.0 + T * \theta} \qquad (7)$$

where $\rho_0(T,\theta)$ the current rest density of an individual air particle, ρ_0 is the rest density of air particle in 0 degrees Celsius, θ is a parameter to steer the rate of change in the air rest density (experimentally set to 0.005).

Unfortunately, due to the relatively low gas constant of air together with its increased viscosity we chose in Sec. 6.1, the result of the application of the equation (7) is not as satisfactory as it was expected to be. Therefore to enhance the simulation of the phenomenon in question, we propose an additional, artificial force, which we include to the set of external forces of SPH technique. The force is calculated with the use of the equation:

$$F_b = \frac{\sigma}{T_{max} - T_i} v_{up} \qquad (8)$$

where σ is an experimentally defined constant to steer the effect of the force, T_{max} is the maximum positive temperature allowed in simulation, T_i is the current temperature of particle i, and v_{up} is the unit upward vector.

7. EXTERNAL HEAT SOURCES

Yet another benefit of the particle-based model of air is that it allows us quite easily to affect ice melting with external heat sources.

In our simulation, an external heat source is modeled to act like hair dryer. It is built with a directional emitter, which effects on air particles in a specified area, so that both the particles' temperatures and their velocities are increased (Fig. 5). A stream of warm air is created, and a portion of its thermal energy is passed to the ice surface, locally altering its temperature.

We noticed that the shape of the air stream partially depends on the viscosity coefficient of air particles. The more "viscous" air is, the more the stream is concentrated, thereby resulting in lesser energy losses.

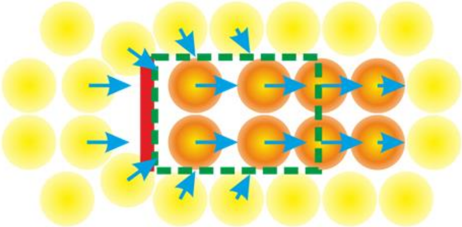


Figure 5. Motion of air particles due to external heat source

During experiments with our external heat sources, we successfully managed to influence on the ice surface in a number of ways: from a precise, shallow melting of a specified symbol, to dividing the ice object into two pieces.

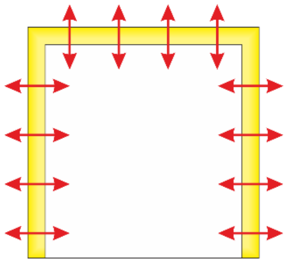


Figure 6. Dissipation of energy on the borders of simulation area

However, due to the finite area of simulation, the continuous application of an external heat source may easily alter the temperature of the entire volume of air. To prevent this we dissipate the excess energy through walls of the "simulation tank": For every particle located near the simulation boarders, we exchange the temperature of the particle with some ambient temperature. This way, we can provide an impression of unlimited area of air (Fig. 6).

8. RESULTS

The proposed method of ice melting simulation requires a significant number of particles for describing all mediums. Hence, the relevant computations could take large amount of time if executed sequentially on CPU. However, the majority of the required operations are performed on each particle using the same set of instructions. Moreover, once input data is delivered to simulator, there is no need to supply any more information. Therefore, in our implementation we can benefit from computational power of modern Graphics Processing Units (GPUs). For this goal, we use the OpenCL framework, which is still relatively new and under constant improvement.

All simulations discussed in this section where performed using a medium class mobile personal computer equipped with Radeon 6770M graphics card. Nevertheless, it allowed us to run the simulation at average speed of 10 FPS with 260k particles (this includes all the types of particles) on each scene.

In order to test the behavior of our air model, the simulations with the Utah teapot and the more complex model of the Stanford Asian Dragon's head were performed. In our opinion the results are more than satisfying. As it can be observed in Fig. 7 and Fig. 8, the ice sculptures are partially melted, mostly in areas with the highest exposure to the surrounding hot air, such as edges and thin elements like teapot ear, dragon horns and teeth.

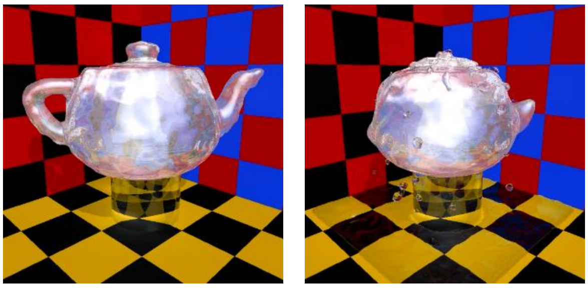


Figure 7. Melting ice teapot due to hot surrounding air

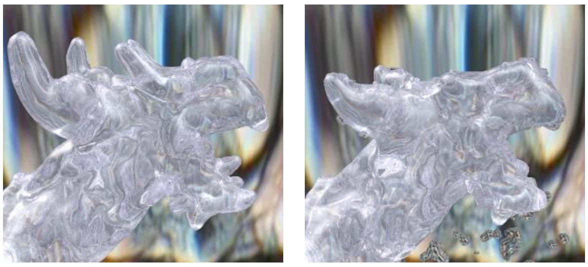


Figure 8. Melting Stanford Asian Dragon's head due to hot surrounding air

Furthermore, it should be noticed, that the amount of energy transferred between air particles themselves as well as air particles and ice particles, firmly depends on distances between particles. This is a result of the presence of the smoothing kernel in the heat transfer equation (1). Thus, by modifying the air viscosity coefficient one can influence the speed of the melting process.

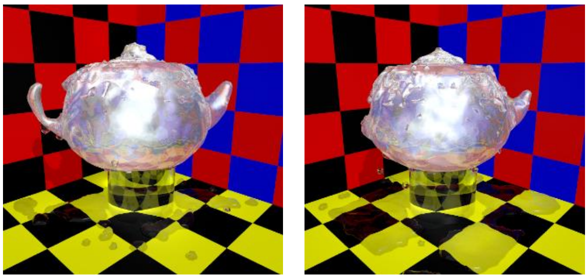


Figure 9. Different viscosity coefficients of air particles: on the left hand side viscosity is five times larger than on the right hand side

One can observe in Fig. 9 that for five times larger viscosity coefficient, the original ice teapot shape is better preserved after the same simulation time. It is a consequence of the lower movement speed of particles due to their higher viscosity, which prevents them from approaching each other and exchanging energy.

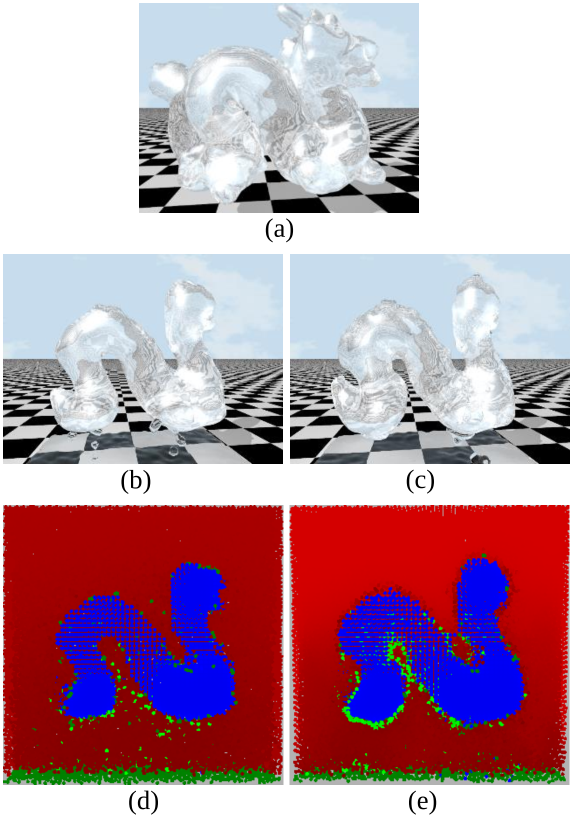


Figure 10. Melting of ice dragon due to hot air without (left column) and with (right column) the natural air convection

With regards to the natural air convection, the result of our efforts are presented in Fig. 10. The image (a) shows the original object before melting, and the left column and the right column show the results of simulation without and, respectively, with considering the natural air convection.

The difference between these two outcomes is subtle. Nevertheless, couple of differences should be pointed out. First, studying the particle views (Fig. 10 d, e), which informs of the particles' temperatures (with blue below 0 and red over 30 degrees), we can observe the behavior of particles with lower temperatures. In the simulation without convection (Fig. 10 d) those particles surround almost the entire ice sculpture. However, when the natural air convection is considered in simulation (Fig. 10 e) the "colder" particles move downwards and locate in the lower parts of the object. Secondly, there are differences in melting between the upper and lower parts of the object. It can be noticed that in the first case (Fig. 10 b) ice melts rather uniformly. Comparing to the second situation (Fig. 10 c), the upper part (where air was hot in both examples) seems to be in the same stadium of melting. However, the lower part with the natural air convection keeps more of its volume intact.

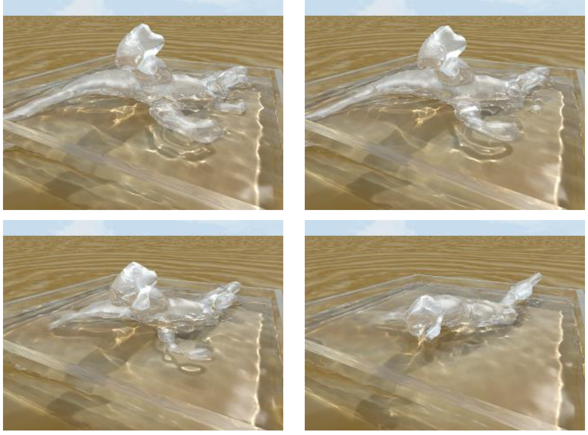


Figure 11. Melting of ice dinosaur partially submerged in water (water and air have the same temperature)

What is more, although the air particles have their density barely 10 times lower than the density of water and there is unnatural movement on the interface (due to the changes we made in the pressure equation), we observed that the heat transfer ratio between those two mediums and ice is very realistic. To confirm that, we conducted an experiment in which an ice dinosaur were partially submerged into a tank filled with water (Fig. 11). The temperature of both air and water was set to the same value. As a result, the submerged part of the object melted significantly faster than the other exposed to surrounding air. Such a phenomenon is consistent with the laws of physics.

As it was stated in Sec. 7, we also successfully managed to precisely alter the ice surface through melting process. The goal of the experiment was melting specified symbol in a block of ice with the use of a number of external heat sources – the results are depicted in Fig. 12.

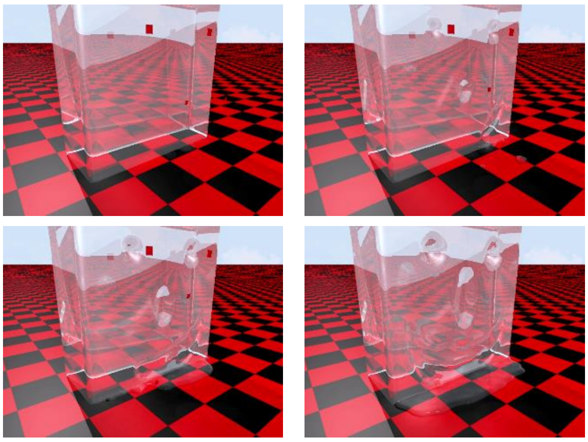


Figure 12. Melting symbol in block of ice with the use of external heat sources

Another test included the external heat source affecting on a number of ice objects. The purpose was to examine how the warm stream of air, created by the heat source, would behave after collision with a streamline object. Hence, we decided to model several different in shape icicles arranged in a line and direct the heat source to one of them.

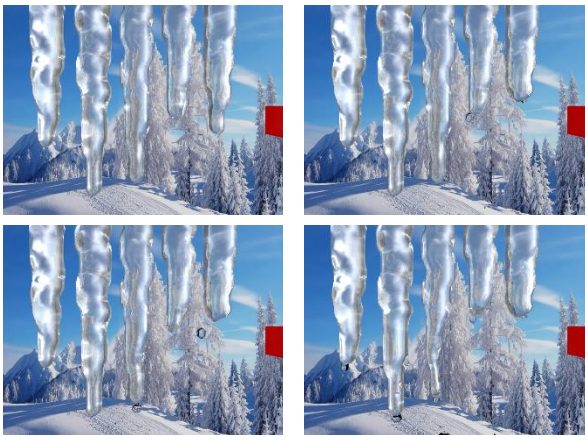


Figure 13. Melting icicles with an external heat source

One can observe in Fig. 13 that the heat source is affecting consecutive icicles respectively, melting one after another. The main volume of the warm stream of air, after collision with an icicle, is being spread sideways. It is also worth noticing that water droplets resulting from the melting process are blown away due to the motion of air.

9. CONCLUSIONS

In this paper we have proposed a simulation of ice melting based on particle representation and performed with the use of the SPH technique. The presented approach combines in a single method the capabilities of the previous approaches. Namely, it allows one to consider in ice melting simulation such phenomena like water, rigid body movement, heat transfer, and creation of water droplets. Moreover, we enhanced the simulation with a new particle-based air model which allows for considering local changes in air temperature, simulating the natural air convection, and controlled ice melting with the use of external heat sources. We have also shown on the example of the proposed air model how to handle multiphase environment. The effectiveness of our approach and the physical correctness of its outcomes was confirmed with a number of test examples. As a consequence, it seems that the presented method can be useful for both animation and physical simulation purposes. Furthermore, thanks to the GPU-based implementation the simulation can be run at interactive speed even on a medium class mobile personal computer.

10. REFERENCES

[CMIT02] Carlson M., Mucha P., III B. V. H., Turk G.: Melting and flowing. In Proc. ACM SIGGRAPH Symposium on Computer Animation (2002), pp. 167-174.

[FA01] Fujishiro I. and Aoki E.: Volume graphics modeling of ice thawing. In Volume Graphics 2001 (2001), Springer-Verlag, pp. 69-80.

[FM07] Fujisawa M. and Miura K. T.: Animation of ice melting phenomenon based on thermodynamics with termal radiation. In Proc. GRAPHITE (2007), pp. 249-256.

[G10] Green S.: Particle Simulation using CUDA [online]. Nvidia Corporation (2010).

[IUDN10] Iwasaki K., Uchida H., Dobashi Y., Nishita T.: Fast Particle-based Visual Simulation of Ice Melting. In Pacific Graphics 2010 (2010).

[K06] Kelager M.: Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics, MS Thesis, Univ. Copenhagen (2006).

[LC87] Lorensen W., Cline H.: Marching Cubes: A high resolution 3D surface construction algorithm. In Proceedings of the 14th annual conference on Computer graphics and interactive techniques (1987), vol. 21, pp. 163-169.

[MCG03] Müller M., Charypar D., Gross M.: Particle-based fluid simulation for interactive applications. In Proc ACM SIGGRAPH/Eurographics Symposium on Computer Animation (2003), pp.154-159.

[MSKG05] Müller M., Solenthaler B., Keiser R., Gross M.: Particle-based fluid-fluid interaction. In SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation (New York, USA, 2005), ACM, pp. 237-244.

[MT05] Matsumura M., Tsuruno R.: Visual Simulation of melting ice considering the natural convection. In SIGGRAPH 05: ACM SIGGRAPH 2005 Sketches (2005), Article No. 61.

[OCD13] Onderik J., Chládek M. and Ďurikovič R.: Animating multiple interacting miscible and immiscible fluids based on particle simulation, Journal of the Applied Mathematics, Statistics and Informatics, University of Saint Cyril and Metod Press, Trnava, Slovakia, De Gruyter, Landsberg, vol. 9, No.2 (2013), pp. 73-86.

[PPLT06] Paiva A., Petronetto F., Lewiner T. and Tavares G.: Particle-based non-Newtonian fluid animation for melting objects. In Sibgrapi 2006 (2006), pp. 78-85.

[SP08] Solenthaler B., Pajarola R.: Density Contrast SPH Interfaces. In SCA '08: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (2008), pp. 211-218.

[TSK07] Tanaka M., Sakaki M., Koshizuka S.: Particle-based rigid body simulation and coupling with fluid simulation. Transactions of Japan Society for Computational Engineering and Science, Paper No. 2007007 (2007).

Cross-domain image matching improved by visual attention

Ernani Viriato de Melo
Federal Institute of Triângulo Mineiro
Rua Edilson Lamartine Mendes 300
38045-000, Uberaba, Minas Gerais, Brazil
ernanimelo@iftm.edu.br

Sandra de Amo
School of Computing
Federal University of Uberlândia
Av. João Naves de Ávila 2121
Bloco 1B
38400-902, Uberlândia, Minas Gerais, Brazil
deamo@ufu.br

Denise Guliato
School of Computing
Federal University of Uberlândia
Av. João Naves de Ávila 2121
Bloco 1B
38400-902, Uberlândia, Minas Gerais, Brazil
guliato@ufu.br

ABSTRACT

A good accuracy in image retrieval across different visual domains, such as photos taken over different seasons or lighting conditions, paintings, drawings, hand-drawn sketches, still is a big challenge. This paper proposes the use of visual attention to estimate the relative importance of some regions in a given query image. Recently, researchers used different databases in specific domains to validate their hypothesis. In this paper, we also propose a database with multiple image domains, called UFU-DDD. We used the UFU-DDD database to demonstrate the performance and accuracy gains from the association of visual attention with orientation-based feature descriptors. The analysis of the results showed that our approach outperforms all the standard descriptors used in the experiments. We hope the UFU-DDD database constitutes a valuable benchmark to the future research in cross-domain similarity searching.

Keywords

Visual attention, image matching, saliency, image retrieval in cross domain, painting, sketches.

1. INTRODUCTION

With ever-faster computers and internet connection, the acquisition of collections of images and videos has become an action of our daily lives. Multiple images may possess exactly the same content across a wide range of visual domains, e.g., photos, paintings, sketches, computer-generated images (CG images), with dramatic variations in lighting conditions, seasons, ages, and rendering styles. The development of methods to efficiently compute the visual similarity between images in different domains is a challenge and an urgent need for various applications, such as scene completion [Hay07], Sketch2Photo [Cao11, Eit11], Internet re-photography [Shr11], painting2GPS [Shr11], and CG2Real [Joh11]. Figure 1 illustrates an example of an application where the user gives a painting of the Coliseum as the query and wants to retrieve photos, paintings, sketches and drawings from the same tourist spot.

The task of comparing images in different domains is very challenging, because small perceptual differences can result in arbitrarily large differences at the raw pixel level. In addition, it is very difficult to develop a generalized solution for multiple potential visual domains. For this task, it is necessary to capture the important visual structures that make two images appear similar. Several different image descriptors have been proposed in the literature based on color, shape and texture. In particular, aiming at representing the salient regions (i.e., high gradient and high contrast) of the image, some descriptors have been proposed in the state of art, such as: SIFT - Scale Invariant Feature Transform [Low99], GIST [Oli06] and HOG - Histogram of Gradients [Dal05].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

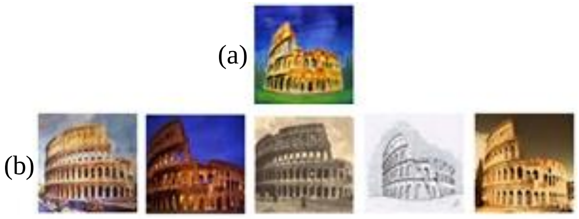


Figure 1. An example of a desired answers list in a cross-domain database: (a) the query; (b) the top-5 answers list of Coliseum in different visual domains.

Recently, researchers have made significant progress in the study of visual similarity in different domains, such as data-driven uniqueness paradigm proposed by Shrivastava et al. [Shr11]. This paradigm aims at focusing on the most important visual parts of the query image. The central idea is to identify the parts of the image are more unique or rare. To that end, Shrivastava et al. [Shr11] proposed to train a linear classifier SVM (Support Vector Machine) at query time, using one feature descriptor with the aim of identifying the uniqueness parts of the image. This method was modified in [Sun13], with an approach that uses multiple features for training the classifier at query time. Although promising results have been achieved, the computational cost to train a SVM classifier for each query is extremely high, preventing its application in real time problems.

A promising alternative to identify relevant parts of an image, with low computational cost, can be found in the studies being conducted on the psychology field related to visual attention. Its central idea is that the most important regions of the image are those that most attracts people's attention. Then, the features extracted from these regions may be strongly weighted for the image retrieval task. Several works [Bor09, Sat10, Soa12] use visual attention to identify different ways to obtain regions of interest and are focused on other tasks such as classification, separation of foreground and background, object recognition, image retrieval. To the best of our knowledge, there is no reference in the literature addressing the use of visual attention in the context of cross-domain image retrieval.

The main goal of this paper is to show that visual attention models can identify the relevant parts of the query image and when associated with image descriptors can contribute to the improvement of the similarity searching accuracy in different visual domains with low computational cost. Differently from the strategy proposed in [Shr11, Sun13], our approach can be computed in real time. An example is illustrated in Figure 2.

The main contributions of this paper can be summarized as follows:

1. We built a new database with images in different visual domains, called UFU-DDD. The database contains 22 classes and each class is composed of 50 images of the same scene in different visual domains. Some examples of images in our database are showed in Figure 4. To our best knowledge, this database is the first one that put together, in a same images class, scenes obtained from several different visual domains, such as photos took over different seasons or lighting conditions, paintings, drawings, computer graphic (CG) images, and sketches. To date, the state-of-the-art databases are constructed to evaluate methods to perform the matching between specific domains [Cho08, Eit11, Rus11] and not for multiple visual domains as proposed in this work. Sometimes, the databases are dynamically created for each query, limiting the comparison and the importance of the experiments [Shr11, Sun13].
2. We proposed a new strategy to associate visual attention maps with well-known orientation based image descriptors such as SIFT [Low99], GIST [Oli06] and HOG [Dal05]. The results showed that our approach overcomes the conventional ones in cross-domain image retrieval.

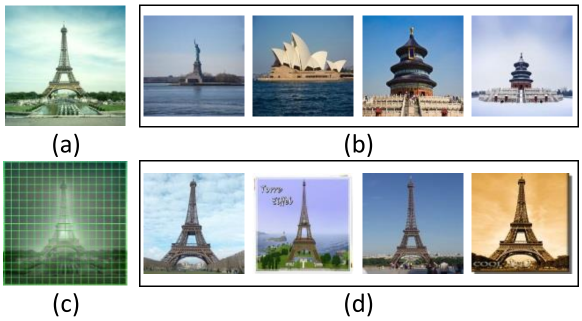


Figure 2. An example of the use of visual attention in an image retrieval task: (a) a photography of the Tower Eiffel; (b) the top-4 answers for the image query in (a) without the use of visual attention, models; (c) the visual attention map of the query image, superimposed by a regular grid. Note that the tower region is now highlighted with respect to the image background; (d) the top-4 answers for the query using an association of the visual attention map with image descriptors in the image retrieval process.

The remainder of this paper is organized as follows. We first give an overview of the related work in Section 2. Section 3 describes our proposal to associate visual attention with orientation-based feature descriptors. The methodology of the experiment and the analysis of the results are provided in Sections 4 and 5 respectively. We discuss about the limitations of our approach and future work in Section 6.

2. BACKGROUND REVIEW

We briefly review related works on cross-domain matching, orientation-based feature descriptors, and models of visual attention.

2.1 Cross-domain Matching

A place, scene, or objects can be recorded in an image in different visual forms, which we call visual domains. Nowadays it is common to find databases containing images with the same semantic content, but in different domains, such as photographs taken over different seasons or lighting conditions, paintings, sketches, drawings, CG images, etc. Many studies have been dedicated to match images between specific domains, such as photos under different lighting conditions [Cho08], sketches to photographs [Cao11, Eit11], paintings to photographs [Rus11], and CG images to photographs [Joh11]. However, these specific domain solutions are not directly extensible to multiple domains.

For a generalized solution for this problem, we highlight three works. In the first one, proposed by Shechtman and Irani in [She07], the authors describe an image in terms of local self-similarity descriptors (SSIM) that are invariant for cross visual domains applications. The second one proposed by Shrivastava et al. [Shr11] and third one proposed by Sun et al. [Sun13] are based on the data-driven uniqueness paradigm. Shrivastava et al. proposed to train a linear Support Vector Machine (SVM) classifier for each query, in query time, to set weights to each dimension of either the HOG or SIFT descriptors. Based on this same idea, Sun et al. proposed to use weighted vectors for multiple features (Filter Bank, SSIM and HOG simultaneously) with weights associated with the each dimensions of these descriptors. The time spent to run a query on a database retrieval of 5,000 images in the method proposed in [Shr11] is under around three minutes on a 200-node cluster, while in the method proposed in [Sun13] is greater than 10 minutes on a PC with a 3.40 GHz Intel i7 CPU and 8 GB RAM. Although promising results have been achieved in both the solutions, these strategies presented a high computational cost, preventing their application in real time similarity searching.

2.2 Orientation-based Descriptors

The descriptors extractors are methods to derive automatically visual information from an image and organize them into a feature vector that represents the image content. In image retrieval run in different visual domains, the locally salient parts of the image are highly relevant information in the calculation of visual similarity. With this aim, several descriptors have been presented in the literature, among them are: SIFT, GIST, and HOG.

2.2.1 Spatial Pyramid SIFT Descriptor

Scale Invariant Feature Transform – SIFT, proposed by Lowe [Low99], is a descriptor that detects a set of keypoints and describe a neighbor of each one in terms of the frequency of gradient orientation. The result is a 128-dimensional vector to describe each keypoint. The SIFT descriptor is invariant to translation, rotation, scale, and illumination conditions.

With the aim of addressing the similar image retrieval images of objects taken in different views, several works use the Bag of Visual Words (BoVW) [Siv03, Csu04, Laz09, Soa12]. In a general way, the BoVW consists of identifying, sparsely or densely, a set keypoints, in a training image database, and cluster them in a predefined number of groups. Each group is referred as a visual word. Then, all the image in the database and the query image are represented by a frequency histogram of visual words. Two images are said similar if their histograms are close to each other according to a similarity measure.

In this paper, we are particularly interested in an extension for the BoVW proposed by Lazebnik, Schmid and Ponce [Laz09] termed Spatial Pyramids (SP). They use a dense regular grid to detect the keypoints. Firstly, SIFT descriptors of *16* x *16* pixel patches of all image database are computed over a grid with spacing of *n* pixels. Then, using the k-means algorithm, SIFT descriptors are grouped and the representative of each group are used to build the visual words dictionary. The frequency histogram of visual words is computed for each image, as in BoVW. Now, the spatial pyramid is computed by partitioning the image into regular sub-regions in several levels, and assessing a frequency histogram for each sub-region. The process continues until it reaches a predetermined number of pyramid levels. The final descriptor is composed by concatenating all frequency histograms derived in the process.

2.2.2 GIST Descriptor

The GIST descriptor presents good results in scene categorization and image retrieval [Oli06]. The idea is to develop a statistical representation with low dimensionality of the scene. The GIST descriptor computes the energy using a bank of Gabor-like filters evaluated at all orientations and different scales for each of the cells obtained by chopping up the image into *N* by *N* pieces. The format of the Gist descriptor is a vector with [*scales*] * [*orientations*] * [*number of cells*] dimensions.

2.2.3 HOG Descriptor

The Histogram of Oriented Gradients – HOG – descriptor [Dal05] was firstly proposed to deal with human detection task and later became a very popular feature in object detection area. When extracting HOG features, the orientations of gradients are usually quantized into histogram bins and each bin has an orientation range. An image is divided into overlapping cells and in each cell a histogram of oriented gradients falling into each bin is computed and then normalized to overcome illumination variation problems. The features extracted from all the cells are then concatenated together to form the HOG descriptor of the whole image, with [*cells*] * [*bins*] dimensions. The HOG features are similar to SIFT descriptor, but HOG features are computed in dense grids at some single scale without orientation alignment. In this paper, we used the HOG descriptor algorithm proposed in [Fel10], which uses 31 orientations bins to compute the histogram of oriented gradients. The similarity between the image query and other image can be computed by various functions of distance, for example, the Canberra distance.

2.3 Models of Visual Attention

Humans are faced with an overwhelming amount of visual information. However, this amount of information is much larger than all the information that the brain processes and assimilates. By rapid eye movements, referred to as saccades movements, the brain must prioritize and receive only part of the visual information at every instant.

Visual attention is the ability of the human visual system to select and process only the most important regions in a scene, while ignoring the rest of the image information. Intuitively, saliency characterizes some parts of a scene which appear with high relevance for an observer. A saliency map indicates the conspicuity of each pixel of the scene, i.e. the probability of parts of the scene to attract the attention of humans. The saliency map is visualized as a grayscale image, where the brightness of a pixel is proportional to its salience. Models of visual attention try to represent the mechanism of visual attention by saliency map. A nice survey about saliency map is presented by Borji and Itti in [Bor13]. The authors presented a classification of attention models into seven categories, considering the strategy used to obtain the saliency map. We evaluate one model of each one category and the GBVS model, proposed by Harel et al. [Har07], presented the best results for the cross-domain problem. Harel et al. proposed a bottom-up saliency map which uses the Markov chains over various feature maps and treats the equilibrium distribution over map locations as activation and saliency values. They proposed to unify the activation map and normalization/ combination maps steps by using dissimilarity and saliency to define edges on graphs which are interpreted as Markov chain. In this work, we'll use the GBVS model in the experiments.

3. OUR APPROACH

In this paper we propose combining visual attention models with image descriptors for image retrieval in different visual domains. Our hypothesis is that the relevant regions of an image, highlighted by a saliency map, are more important to characterize an image for content based image retrieval. Figure 3 summarizes the idea proposed in this paper.

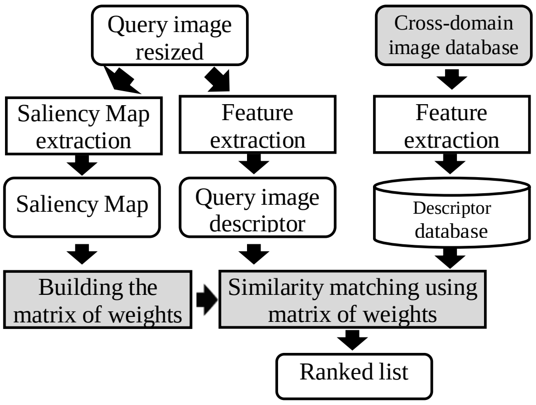


Figure 3. The flowchart of the association of visual attention with image descriptors. The darker boxes highlight our contributions.

Following Figure 3, for the query image, two processes occur in parallel: saliency map extraction, and feature extraction. The saliency map extraction can be performed by any of the visual attention models already developed. Because the nature of cross domain database, we will use the feature extraction method based on gradient / contrast orientation, that are SIFT, GIST and HOG descriptors, but the image descriptors are not limited to them and depend on the application.

After extracting the saliency map, the saliency magnitude is normalized to the range [0, 1], and then it is superimposed by an *NxN* size grid. After that, the matrix of weights, with *NxN* dimension, is derived with the aim of highlighting the relevant parts of the image while discard or attenuate the importance of the remainder regions. To that end, the value of the position (*i,j*) of the matrix of weights is assessed by computing the median saliency magnitude value of the normalized saliency map in the corresponding grid cell. Those median values that are lower than a given threshold *T* is set zero, indicating the low importance of that image for the representation of the scene. The matrix of weights is built as showed in Algorithm 1. Finally, we use information of the matrix of weights in same way to run the similarity searching.

Algorithm 1: Building of matrix of weights

```
MatrixOfWeights(I, MV, T, N)
  Input: I: image; MV: model of visual attention;
         T: threshold; N: grid size
  Output: W: N x N matrix of weights
begin
  SM = SaliencyMapExtraction(I, MV)
  SM = Normalize(SM, 0, 1)
  for each par(i, j) Є grid do
      Md = median(SM, i, j)
      if (Md < T)
          W[i,j] = 0
      else
          W[i,j] = Md
  return W
End
```

3.1 Visual Attention with Spatial Pyramid/ SIFT (VA-SP-SIFT)

In our approach, only the SIFT descriptors extracted from the relevant regions are used to construct the Spatial Pyramid, as shown in Algorithm 2. The similarity measure between the image query and other images is computed by the histogram intersection function, as described in [Laz09].

3.2 Visual Attention with GIST (VA-GIST)

The matrix of weights, different from the strategy described in Section 3.1, is used to weight the similarity measure in query time.

Let *I* and *J* be the query image and the target image. Let W^I, G^I and G^J be the matrix of weights of *I*, obtained as described in Algorithm 1, the GIST vector for *I* and the GIST vector for *J*, respectively. *G(c,s,r)* is GIST vector value of the cell *c* in a scale *s* and orientation *r*, where *c = {1, ..., m}*, *m= NxN* is a number of cells, *s = {1, .., p}*, *p* is a number of scales, *r = {1,..., q}*, *q* is the number of gradient orientations. In this work we propose to use the Weighted Euclidean distance as defined in Eq. 1, to determine the similarity between *I* and *J*.

$$D(I,J): \sqrt[2]{\sum_{c=1}^{m}\sum_{s=1}^{p}\sum_{r=1}^{q}\left(1-\left(G^I(c,s,r)-G^J(c,s,r)\right)^2\right)W^I(c)} \quad (1)$$

where $W^I(c)$ is a weight associated with the c^{th} cell in the matrix of weights.

```
Algorithm 2: VA-SP-SIFT Descriptor

  Data: I: image query; MV: model of visual attention;
        T: threshold; N: grid size;     D: dictionary
  Result: V: VA-SP-SIFT descriptors vector
begin
    W = MatrixOfWeights (I, MV, T, N)
    SIFTS = Dense-SIFT(I)   //as in [Laz09]
    SIFTS_VA = Fusion (SIFTS, W) //only the SIFT
                   descriptors that fall within the cells
                   where the corresponding position in W
                   is different from zero are kept.
    V = ProduceSP(SIFTS_VA, D) //as in [Laz09]
    return V
end
```

3.3 Visual Attention with HOG (VA-HOG)

For HOG, the proposal is similar to that one described in Section 3.2.

Let *I* and *J* be the query image and the target image. Let W^I, H^I and H^J be the matrix of weights of *I*, see Algorithm 1, the HOG vector for *I* and the HOG vector *J*, respectively. *H(c,i)* is the normalized count of the i^{th} orientation bin of the c^{th} cell, where *c = {1,..,m}*, *m* is a number of cells, *i = {1, .., 31}*. $W^I(c)$ is a weight associated to c^{th} cell. The similarity between *I* and *J* is computed by using the Weighted Canberra distance, as shown in Eq. 2.

$$D(I,J) = \sum_{c=1}^{m}\left(\sum_{i=1}^{31} 1 - \frac{|H^I(c,i) - H^J(c,i)|}{|H^I(c,i)| + |H^J(c,i)|}\right)W^I(c) \qquad (2)$$

4. METHODOLOGY

We run several experiments in order to analyze the performance of the use of visual attention for cross-domain image matching. The experiments are divided according to the feature descriptor used. The results of quantitative analysis are reported in terms of the Average Precision (AP) values at the top-k answers.

4.1 Databases

The scientific community needs a unique database to evaluate methods towards image retrieval across visual domains. Aiming to address this need, we created a public database, called UFU-DDD (Database of Different domains of University of Uberlândia). We also use 10,000 images from the MIRFLICKR Database [Hui08] to test the robustness of our proposal.

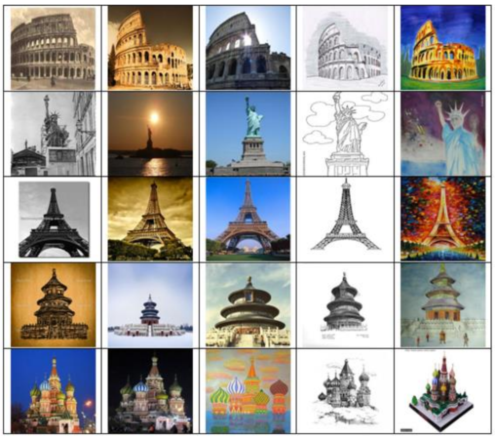


Figure 4. A sub-set of classes of UFU-DDD: Coliseum, Statue of Liberty, Eiffel, Temple of Heaven, Saint Basil's Cathedral.

Database of Different Domains of University of Uberlândia (UFU – DDD) – we have created a new database comprised of 1,100 images. The UFU-DDD database was collected by crawling images from google images website using keywords about tourist spot such as "painting of Tower Eiffel", "sketch of Cathedral San Basilio". This procedure was necessary because we did not find a database with the particularities that we consider important to evaluate our approach. In order to obtain classes with a variety of domains, we decided that each class would be a tourist spot with exactly 50 images across different domains, totaling 22 classes. The tourist spots are many, such as: waterfall, church, coliseum, temple, stadium, castle, museum, opera house, etc. The database contains 91 very old photographs, 677 photographs under different lighting and stations, 150 sketches, drawings and CG images, and 182 paintings. In all the cases the foreground is centered in the image. Figure 4 shows five classes of UFU-DDD and each class with images in different visual domains. With UFU-DDD it is possible to design experiments such as: given a query painting, what are the photographs, drawings, and sketches more visually similar? Given an old photograph, what are the recent photographs of the same place? Given a sketch, which are the paintings of the corresponding place?

MIRFLICKR Database (MIF) – this database contain 1 million Flickr images under the Creative Commons license. It is commonly used for the visual concept detection, photo annotation and image retrieval task. We used 10,000 photographs of this database just to test the performance of our approach under different database sizes.

4.2 Experiment Setup

In all experiments, the images were resized to 200 x 200 pixels. The extraction of the saliency map was done using GBVS model proposed by Harel et al. [Har07]. For each image descriptor we compared our approach against a publicly available, third-party authored reference, implemented in Matlab. The parameters used to derive each descriptor are described following.

VA-SP-SIFT: for each image, we compute spatial pyramid representation with 3 pyramid levels using Dense-SIFT descriptors of 16x16 pixels patches computed over a grid with spacing of 8 pixels. We used a vocabulary of 400 visual words. After several tests, we empirically set the threshold value at 0.2 according to Algorithm 1.

VA-GIST: we computed the GIST representation for each image using an 8 x 8 grid, 4 scales, and 8 orientations. The threshold T in Algorithm 1 is set at zero. All the weights are used in the Weighted Euclidean distance, as defined in Eq. 1.

VA-Normalized-HOG (VA-NHOG): for each image, we compute HOG representation with 625 cells of 8 x 8 pixels divided in a 25 x 25 grid, and 31 orientations bins. We experimented both implementation, the standard HOG descriptor as well a simple normalized HOG (NHOG). The NHOG vector (VNHOG) is defined as a zero-centered version of HOG vector (VHOG), where VNHOG = VHOG – mean(VHOG). We perceived slightly better results with NHOG and then we adopted it in our experiments. The threshold T in Algorithm 1 is empirically set at 0.3. We also evaluate different similarity measures, such as Cosine, Chi-square, Euclidean, Manhattan, Canberra distances. We adopted the Weighted Canberra distance, as defined in Eq. 2.

5. EXPERIMENTS

In order to evaluate the performance of our approach, we conducted three experiments, each one using a specific domain as the query image. The experiments are run using the UFU-DDD database. We also used images from the MIRFLICKR database as distractors, in three different versions: UFU-DDD + 3,000 MIF; UFU-DDD + 6,000 MIF; and UFU-DDD + 10,000 MIF.

5.1 Photograph as Queries

In this experiment, the query images are photos took over different ages, seasons, weather or lighting conditions. We collected from the UFU-DDD a dataset of 44 photos (2 photos of each class) to be used as queries. Table 1 shows the Average Precision at top 10 (AP@10) and top 30 (AP@30). In all the cases, our approach obtained an important improvement in the results when compared to the standard descriptors. The gain obtained for our proposal varies from 9% to 30% for top 10 and from 5% to 15% for top 30. The gain depends on the descriptor and the database size. It is worth noting that the inclusion of distractors in the database did not affect the gain obtained with the use of visual attention. Figure 5 (a) shows the top 3 answers for the Sydney Opera House with and without the use of visual attention. This example illustrates the superiority of our approach.

Methods	Databases							
	UFU-DDD		UFU-DDD+ 3,000MIF		UFU-DDD+ 6,000MIF		UFU-DDD+ 10,000MIF	
	@10	@30	@10	@30	@10	@30	@10	@30
SP-SIFT	0.46	0.29	0.40	0.24	0.38	0.22	0.31	0.20
VA-SP-SIFT	**0.64**	**0.39**	**0.51**	**0.31**	**0.47**	**0.27**	**0.43**	**0.25**
GIST	0.58	0.36	0.55	0.32	0.53	0.31	0.52	0.29
VA-GIST	**0.79**	**0.46**	**0.75**	**0.42**	**0.74**	**0.40**	**0.72**	**0.38**
NHOG	0.53	0.35	0.46	0.29	0.43	0.27	0.40	0.24
VA-NHOG	**0.83**	**0.50**	**0.71**	**0.40**	**0.66**	**0.36**	**0.63**	**0.33**

Table 1. Average Precision at top 10 and top 30 with different database sizes. Queries: Photo.

Methods	Databases							
	UFU-DDD		UFU-DDD+ 3,000MIF		UFU-DDD+ 6,000MIF		UFU-DDD+ 10,000MIF	
	@10	@30	@10	@30	@10	@30	@10	@30
SP-SIFT	0.31	0.20	0.28	0.16	0.26	0.14	0.25	0.13
VA-SP-SIFT	**0.43**	**0.25**	**0.36**	**0.20**	**0.32**	**0.19**	**0.31**	**0.17**
GIST	0.43	0.26	0.38	0.21	0.36	0.20	0.34	0.18
VA-GIST	**0.57**	**0.34**	**0.51**	**0.30**	**0.47**	**0.27**	**0.46**	**0.26**
NHOG	0.41	0.26	0.36	0.20	0.33	0.18	0.31	0.16
VA-NHOG	**0.65**	**0.41**	**0.53**	**0.32**	**0.49**	**0.29**	**0.46**	**0.27**

Table 2. Average Precision at top 10 and top 30 with different database sizes. Queries: Sketch / Drawing.

5.2 Sketch/Drawing as Queries

Here, the query images are sketches and drawings. We collected a dataset of 44 sketches and drawings (0 to 3 images of each class) to be used as queries (two classes do not contain sketches). Matching sketches/drawings to real scenes is a difficult task. The sketches and drawings are abstract and show strong local deformations with respect to the real scene. Table 2 shows the AP@10 and AP@30. In all cases, it is possible to note an improvement in the results that vary from 6% to 24% for top 10 and from 4% to 15% for top 30. Figure 5 (b) shows a qualitative examples corresponding to the top 3 answers for each descriptor, using or not the attention model. It can be seen that our approach returned 3 relevant photos as answers for sketch used as query. The 3 images returned are relevant to the 3 feature descriptors.

5.3 Painting as Queries

We collected a dataset of 44 paintings (1 to 3 images of each class) to be used as queries. Matching paintings to scenes is also a difficult task because: i) the presence of strong local gradients due to brush strokes; and ii) the painting styles may vary from painter to painter. Table 3 shows the AP@10 and AP@30. In all cases, our approach outperforms the standard descriptors. The gain reached by our proposal varies from 6% to 17% for top 10 and from 3% to 9% for top 30. A Qualitative example is showed in Figure 5 (c). Due to the difficult of match painting to photos or sketches, the standard descriptors failed in all the answers while our approach returned at least two relevant answers in three.

Methods	Databases							
	UFU-DDD		UFU-DDD+ 3,000MIF		UFU-DDD+ 6,000MIF		UFU-DDD+ 10,000MIF	
	@10	@30	@10	@30	@10	@30	@10	@30
SP-SIFT	0.38	0.24	0.31	0.18	0.29	0.16	0.26	0.14
VA-SP-SIFT	**0.50**	**0.31**	**0.39**	**0.24**	**0.35**	**0.20**	**0.32**	**0.17**
GIST	0.44	0.26	0.40	0.22	0.37	0.20	0.35	0.19
VA-GIST	**0.58**	**0.34**	**0.53**	**0.29**	**0.49**	**0.27**	**0.46**	**0.26**
NHOG	0.43	0.28	0.36	0.20	0.31	0.16	0.28	0.14
VA-NHOG	**0.60**	**0.37**	**0.48**	**0.28**	**0.44**	**0.24**	**0.40**	**0.21**

Table 3. Average Precision at top 10 and top 30 with different database sizes. Queries: Painting.

6. CONCLUSIONS

In this paper, we presented a new strategy with low computational cost to highlight the most important parts of an image query with the purpose of images retrieval in databases that contain images in different visual domains. The strategy was evaluated with a different database sizes. We showed that our approach outperforms the standard descriptors. However, this strategy is strongly dependent on the model of visual attention to be used. A typical failure is showed in Figure 6. In this example, our approach fails to find good top matches because the attention model was not able to identify all the body of the Statue of Liberty. Further works are in progress to detect relevant parts of an image, interactively by using eye tracker device.

7. ACKNOWLEDGMENTS

We thank Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG), and Conselho de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for the financial support.

8. REFERENCES

[Bor09] Borba, G. B., Gamba, H. R., Marques, O., and Mayron, L. M. Extraction of salient regions of interest using visual attention models. In IS&T/SPIE Electronic Imaging, 2009.

[Bor13] Borji, A., and Itti, L. State-of-the-Art in Visual Attention Modeling. TPAMI, 35(1): 185-207, 2013.

[Cao11] Cao, Y., Wang, C., Zhang, L., and Zhang, L. Edgel index for large scale sketch-based image search. CVPR, pp. 761–768, 2011.

[Cho08] Chong, H. Y., Gortler, S. J., and Zickler, T. A perception-based color space for illumination-invariant image processing. ACM Trans. Graph. 27(3), 61, 2008.

[Csu04] Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. Visual categorization with bags of keypoints. ECCV, 2004.

[Dal05] Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection.CVPR, 2005.

[Eit11] Eitz, M., Hildebrand, K. , Boubekeur, T. and Alexa, M. Sketch-based image retrieval: benchmark and bag-of-features descriptors. TVCG, 17(11), 1624–1636, 2011.

[Fel10] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. TPAMI, 32(9), 2010.

[Har07] Harel, J., Koch, C., and Perona, P. Graph-based visual saliency. Advances in Neural Information Processing Systems, 19, 2007.

[Hay07] Hays, J. and Efros, A. A. Scene completion using millions of photographs. In ACM Trans. Graph. 26(3), 2007.

[Hui08] Huiskes, M. J., and Lew, M. S. The MIR Flickr Retrieval Evaluation. ACM MIR'08, Vancouver, Canada, 2008.

[Joh11] Johnson, M. K., Dale, K., Avidan, S., Pfister, H., Freeman, W. T., and Matusik, W. CG2Real: improving the realism of computer generated images using a large collection of photographs. TVCG. 17(9), 1273–1285, 2011.

[Laz09] Lazebnik, S., Schmid, C., and Ponce, J. Spatial pyramid matching. In Object Categorization: Computer and Human Vision Perspectives. Cambridge University Press, 2009.

[Low99] Lowe, D. G. Object Recognition from local scale–invariant features. ICCV, 1999.

[Oli06] Oliva, A., and Torralba, A. Building the gist of a scene: the role of global image features in recognition. Progress in Brain Research. 155, 23-36, 2006.

[Rus11] Russell, B. C., Sivic, J., Ponce, J., and Dessales, H. Automatic alignment of paintings and photographs depicting a 3D scene. 3dRR, 2011.

[Sat10] Sato, M., and Katto, J. Performance improvement of generic object recognition by using seam carving and saliency map. IWAIT, 2010.

[She07] Shechtman, E., and Irani, M. Matching local self-similarities across images and videos. CVPR, 1–8, 2007.

[Shr11] Shrivastava, A., Malisiewicz, T., Gupta, A., and Efros, A. A. Datadriven visual similarity for cross-domain image matching. ACM Trans. Graph. 30(6), 154, 2011.

[Siv03] Sivic, J., and Zisserman, A. Video google: A text retrieval approach to object matching in videos. IEEE Int. Conf. on Computer Vision,. 1470-1477, 2003.

[Soa12] Soares, R. C., Silva, I. R., and Guliato, D. Spatial Locality Weighting of Features using Saliency Map with a Bag-of-Visual-Features. ICTAI, 1-6, 2012.

[Sun13] Sun, G., Wang, S., Liu, X., Huang, Q., Chen, Y., and Wu, E. Accurate and efficient cross-domain visual matching leveraging multiple feature representations. The Visual Computer, 29, 565-575, 2013.

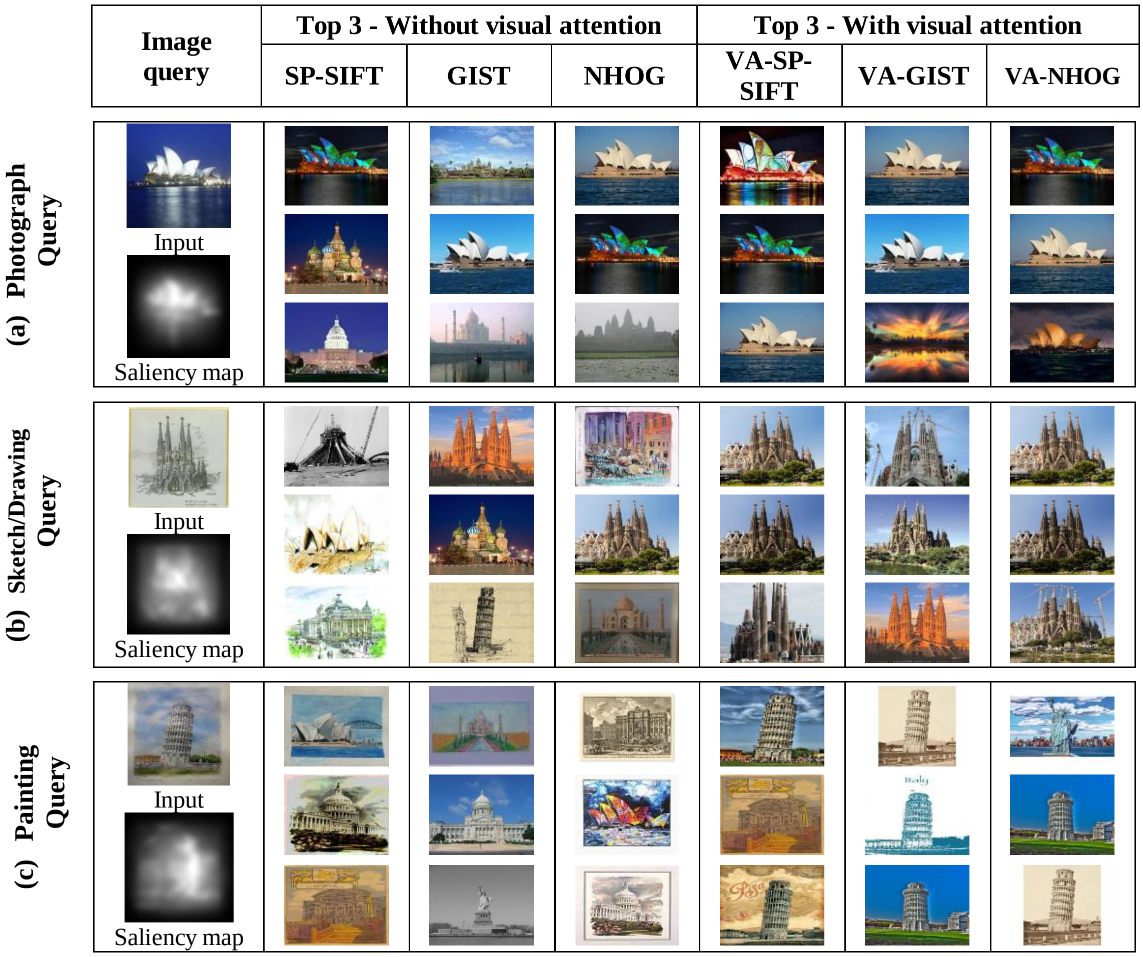


Figure 5. Qualitative comparison of feature descriptors with and without the GBVS model.

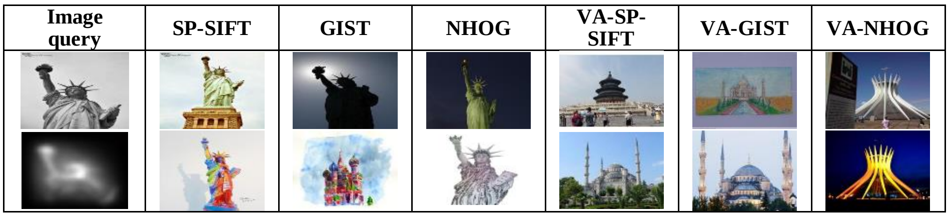


Figure 6. Typical failure case. The GBVS model mainly highlighted the head and torch of the Statue of Liberty as relevant regions. However, regions as the body and the sky around the Statue of Liberty are important for CBIR.

Acquiring Bidirectional Texture Functions for Large-Scale Material Samples

Heinz Christian Steinhausen Dennis den Brok Matthias B. Hullin Reinhard Klein
Universität Bonn
Institute of Computer Science II
Friedrich-Ebert-Allee 144
53113 Bonn, Germany
{steinhau, denbrok, hullin, rk}@cs.uni-bonn.de

Abstract

Most current acquisition setups for bidirectional texture functions (BTFs) are incapable of capturing large-scale material samples. We propose a method based on controlled texture synthesis to produce BTFs of appealing visual quality for such materials. Our approach uses as input data a complete measurement of a small fraction of the sample, together with few images of the large-scale structure controlling the synthesis process. We evaluate the applicability of our approach by reconstructing sparsified ground truth data and investigate the consequences of choosing different kinds and numbers of constraint images.

Keywords
bidirectional texture functions - texture synthesis - material appearance.

1 INTRODUCTION

To produce realistically looking scenes in computer graphics, suitable material representations beyond two-dimensional textures are needed. Data-driven models like photographically acquired *bidirectional texture functions* (*BTFs*) are capable of capturing a wide range of optical effects. However, due to practical constraints the capture of full BTFs is limited to small sample sizes. The goal of this work is to capture materials with large-scale structure such as wood (see Fig. 1), ornamented cloth, or structured wallpapers, for which tiling or uncontrolled synthesis fails to produce realistic results.

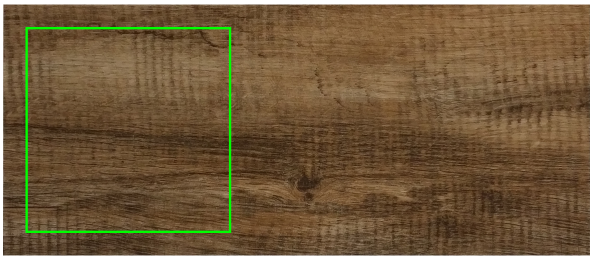


Figure 1: Example of a piece of wood for which a BTF measurement using typical sample sizes, bordered green, would not be able to capture all of its structure.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

BTFs are usually acquired in a data-driven fashion by taking and combining series of photographs shot using different lighting and viewing directions. Not only is the capture volume of available setups typically much smaller than a material's characteristic structure, complete measurements are also often prohibitively expensive regarding both the time needed for acquisition and postprocessing (hours to days) and the storage requirements (terabytes). In agreement with literature [Don13], [Hai13], we observe that BTF data is highly redundant. In particular, disjoint regions typically share very similar visual properties.

We therefore propose to use a texture synthesis approach to capture the visual appearance of materials with large-scale structure. Our method performs the following steps:

1. complete measurement of a small representative spatial region,
2. acquisition of few images of the full sample, and
3. synthesis of a complete BTF.

The term "representative" in this context means that the region to be measured is chosen in such a way that it contains all effects regarding reflectance and surface structure which can be observed throughout the complete material sample. The modeling of appearance as BTFs gives rise to large amounts of data that need to be managed efficiently. We therefore propose to perform all steps in a compressed representation, similar to the work on material interpolation by Ruiters *et al.* [Rui13].

The focus of this work lies on surveying the applicability of texture synthesis to the task of completing sparse BTF measurements, widely disregarding the further challenges mentioned above. Evaluation takes place on sparsified ground truth data for which our method is able to produce visually pleasing results. We start by providing an overview of related work in Section 2, followed by a detailed description of our technique in Section 3, including considerations on the right choice of controlling constraints. We then deliver some experimental results in Section 4.

2 RELATED WORK

2.1 Bidirectional Texture Functions

Bidirectional texture functions (BTFs) as introduced by Dana *et al.* [Dan99] are one model to describe the reflection behavior of materials, closely related to *bidirectional reflectance distribution functions (BRDFs)* and their extension to the spatial domain, *spatially varying BRDFs (SVBRDFs)*. For a taxonomy of reflectance models, see *e.g.* the 2013 textbook by Haindl and Filip [Hai13]. While plain BRDFs describe the reflectance of a material depending on in- and outgoing light directions only, the other two also take into account spatial variations in the reflection behaviour of a material. Both models share properties like energy conservation and Helmholtz reciprocity. SVBRDFs may be a sufficient representation for materials without complex surface structure, but they are not able to capture non-local effects such as self-shadowing, interreflections and subsurface scattering.

For modeling materials exhibiting such effects, BTFs are regarded to be more suitable. A BTF can be interpreted as a generalization of SVBRDFs where the per-texel BRDFs are not true BRDFs anymore but *apparent BRDFs (ABRDFs)* which account for the effects ignored by (SV)BRDFs. Formally, a *bidirectional texture function (BTF)* is a six-dimensional function $\mathcal{B}(\mathbf{x}, \theta_i, \phi_i, \theta_v, \phi_v)$, parametrized over surface position $\mathbf{x} = (x, y)$, direction of incoming light θ_i, ϕ_i and viewing direction θ_v, ϕ_v. Commonly, a discrete approximation of this function is captured using photographic devices like the camera domes proposed by Müller *et al.* [Mül05] and Schwartz *et al.* [Sch13]. This leads to a more intuitive interpretation of a BTF as a “stack of textures”, where each texel does not longer contain only one color value, but one for each combination of lighting and viewing direction. The devices mentioned above provide a good tradeoff between sample sizes and sampling density in spatial as well as in angular domain and support sample sizes of approximately 10 cm × 10 cm. For an overview on methods for material appearance acquisition, see again the textbook by Haindl and Filip [Hai13] or the recent survey by Schwartz *et al.* [Sch14] focusing on BTFs.

Several methods have been developed to reduce the large amounts of data resulting from a BTF measurement. In our work, we utilize compression based on *singular value decomposition* as it was first used by Suen and Healey [Sue00] for dimensionality analysis on BTFs. Applied to a matrix A, the result is a decomposition into three matrices that, when multiplied, approximate the original matrix:

$$A \approx \tilde{A} = U \cdot \Sigma \cdot V^T. \tag{1}$$

The decomposition of a BTF is sketched in Fig. 2. The number of eigenvalues and corresponding eigenvectors which are kept determines both memory consumption and visual fidelity of the compressed BTF.

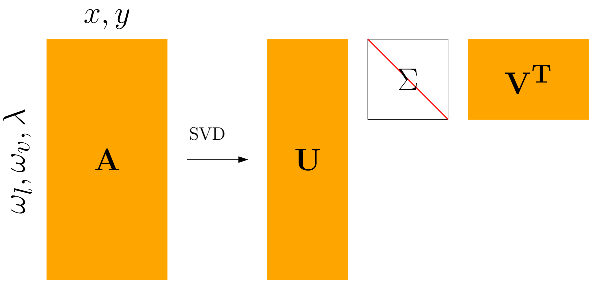


Figure 2: Schema of SVD-based compression of a measured BTF, stored in matrix A. The columns of A contain ABRDFs, one for each surface coordinate (x, y), while the rows contain textures, one for each combination of lighting direction ω_l, viewing direction ω_v and bandwidth or color channel λ. A is decomposed into a matrix U of eigen-ABRDFs, a diagonal matrix Σ containing singular values and a matrix V of eigentextures.

2.2 Synthesis and Reconstruction

Texture synthesis has been an active field of research for several years now. An elaborate survey of the example-based texture synthesis methods from the 1990s to 2009 is given by Wei *et al.* [Wei09].

Also, the idea of synthesizing bidirectional texture functions has been around for quite some years now. Tong *et al.* [Ton02] propose a pixel-wise method to synthesize a new BTF directly onto a surface. Another approach, based on image quilting, is given by Zhou *et al.* [Zho05]. Synthesis of realistic textures with complex geometry like fur is the goal of the approach by Furukawa *et al.* [Fur05]. Haindl and Hatka [Hai05] propose a tiling method for BTFs. Leung *et al.* [Leu07] also rely on tiling, generating a set of seamless Wang tiles from a compressed representation of measured BTF data. A recent approach by Ruiters *et al.* [Rui13] transfers the idea of texture interpolation [Rui10] to BTFs, allowing to generate a new texture which is perceived as lying in-between the appearance of two given input examples.

What these methods for BTF synthesis lack is the possibility to generate a BTF for a large material sample from a measurement of a smaller portion of it. There are two main reasons for choosing *Texture Optimization* as proposed by Kwatra *et al.* [Kwa05] for this task. On the one hand, their algorithm allows for controlled texture synthesis, and on the other hand, it was already successfully utilized for the synthesis of BTFs by Ruiters *et al.* in their interpolation method. The algorithm is regarded lying in-between pixel- and patch-based approaches, as it optimizes the output texture pixelwise, minimizing a global energy function of neighborhood similarities.

An approach for reconstructing spatially varying appearance with reduced measurement effort was proposed by Dong *et al.* [Don10], but only for SVBRDFs. They also acquire data in two phases. First, they take a set of representative BRDF measurements for manually selected surface positions using a one-pixel camera. In a second step, they capture a set of *key measurements*, measuring reflectance densely over the surface, but angularly sparse. The actual reflectance vector for a surface point is then constructed by fitting a manifold of analytical BRDFs to the representative vectors, controlled by the key measurements. It is not obvious how to adopt their method for BTF enlargement, as ABRDFs contain non-local effects not captured by (SV)BRDFs.

3 DESCRIPTION OF METHOD

Our method for generating BTFs for large-scale material samples can be divided into three main parts:

1. Completely measure a BTF $\mathcal{S}$ for a small representative region,
2. acquire a set $\mathcal{C}$ of images of the large-scale structure using as few lighting and viewing directions as possible,
3. and finally use controlled texture synthesis to combine the gathered information resulting in a BTF for the large sample.

In this work, we focus on the third step. We demonstrate how texture synthesis can be applied to the problem of completing sparse measurements to obtain a BTF for the full sample. Synthesis does not take place on the raw data of a measurement, but on the $\Sigma \cdot V^T$-part of its compressed and logarithmic range reduced version (see Eq. 1 in Section 2.1). This approach saves computation time and memory and was already successfully used for BTF synthesis and even interpolation by Ruiters *et al.* [Rui13]. If the region for $\mathcal{S}$ is carefully chosen to contain all aspects of reflectance and surface structure present in the full sample, the ABRDFs stored in U can be expected to contain all information needed for a faithful completion.

We evaluate our method by reconstructing ground truth data from BTF measurements, applying the workflow depicted in Fig. 3a. In the following subsections, we explain our reconstruction method as well as the preparation of input data in more detail.

3.1 Reconstruction Method

The main part of our reconstruction method is based on Texture Optimization [Kwa05]. The algorithm performs pixelwise optimization of the output texture, minimizing a global energy function assembled from local similarity measures of neighborhoods in the input and output image. The energy function is constructed as follows:

$$E_t(\mathbf{x};\{\mathbf{z}_p\}) = \sum_{p \in X^\dagger} \|\mathbf{x}_p - \mathbf{z}_p\|^2, \qquad (2)$$

where $\mathbf{x}$ is the vectorized version of the texture X being optimized, $\mathbf{x}_p$ the subvector of $\mathbf{x}$ containing pixels in a neighborhood of fixed size around p, $\mathbf{z}_p$ a vectorized neighborhood in the example texture Z which is most similar to $\mathbf{x}_p$ under the Euclidean norm, and $X^\dagger \subset X$ the set of neighborhood centers to consider. Optimization takes place in an Expectation-Maximization-like manner, alternating between optimizing the $\mathbf{x}_p$ and $\mathbf{z}_p$. For constrained synthesis, E_t is augmented by a control term E_c, depending on $\mathbf{x}$ and a control vector $\mathbf{u}$, weighted by a coefficient λ:

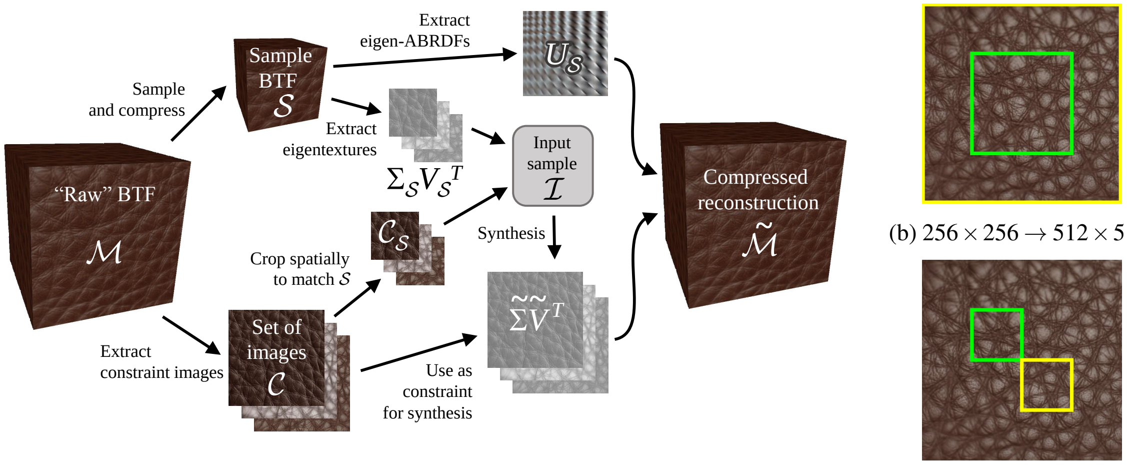
$$E(\mathbf{x}) = E_t(\mathbf{x};\{\mathbf{z}_p\}) + \lambda E_c(\mathbf{x};\mathbf{u}). \qquad (3)$$

In this paper, we use a value of $\lambda = 20\,000$, leading to a ratio of nearly one-to-one between constraint satisfaction and neighborhood similarity on the $151 \cdot 151 = 22\,801$ views in $\mathcal{S}$, see also Subsection 3.2.

To capture material structure at different scales, Kwatra *et al.* propose to perform synthesis in a multi-level fashion, starting from coarse image resolution and large neighborhoods, successively reducing neighborhood size and downsampling factor. We use a neighborhood size of 8 ABRDFs and downsampling factors $1/4, 1/2$ and 1 for our experiments.

Results from works like those by Ruiters *et al.* [Rui10] have shown that during texture synthesis, high-dimensional details are often lost. Therefore, they propose to apply an additional step of *Statistical Synthesis* as introduced by Portilla and Simoncelli [Por00] to compensate for this effect. Their method uses a wavelet transform of a sample image to generate constraints which are then used to generate new images obeying the same statistics. We follow this approach, but incorporate it into a second optimization step applied to the result of Texture Optimization.

Statistical Synthesis applies its modifications uniformly over the Texture Optimization output, leading to the



(b) $256 \times 256 \rightarrow 512 \times 512$.

(a) Workflow for evaluating our methods on ground-truth data.

(c) $128 \times 128 \rightarrow 128 \times 128$.

Figure 3: Our experimental set-up: (a) illustrates the workflow for input generation and ground-truth reconstruction, while (b) and (c) denote the regions in $\mathcal{M}$ from which the inputs are cropped: From the region bordered green, the complete measurement $\mathcal{S}$ is taken, while the images for $\mathcal{C}$ are taken from the area bordered yellow (outer region in (b)).

visible effect that sharpness in the region corresponding to the sample gets exaggerated. To overcome this issue, we then search for an optimal trade-off between the results before and after Statistical Synthesis while still obeying the constraints. Formally, we minimize for each position p on the material surface the function

$$f(p) = \alpha ||x^p - b_1^p||^2 + \beta ||x^p - b_2^p||^2 \qquad (4)$$

under the constraint $A \cdot x^p = b^p$, where x^p are the color values of the ABRDF to be optimized, b_1^p, b_2^p the ABRDF's color values before and after Statistical Synthesis, α and β constant factors levelling between the two objectives, A those rows of the basis $U_{\mathcal{S}}$ of $\mathcal{S}$ which correspond to the constraint images, and b^p being the color values in the corresponding positions in the synthesis constraints unrolled as vectors. From now on, we will omit the superscript p in favor of readability.

By transforming Equation 4, we get

$$\begin{aligned} f(p) &= {}^1\!/_2\, x^T \cdot \big(2(\alpha+\beta)I\big) \cdot x && (5) \\ &\quad + \big(-2(\alpha b_1^T + \beta b_2^T)\big) \cdot x + C \\ &= {}^1\!/_2\, x^T \cdot Q \cdot x + c^T \cdot x + C && (6) \end{aligned}$$

with c, C constants, which is known to be solvable using quadratic programming.

We implemented this optimization in a MATLAB script, iteratively alternating between applying Statistical Synthesis to the result of the previous step and optimizing for x that minimizes Equation 6 until the result does not change significantly anymore. In our experiments, two to five iterations were sufficient to reach convergence. Algorithm 1 summarizes our complete approach. The result of this algorithm, applied to the inputs $\mathcal{I}$ and $\mathcal{C}$ as constructed in the following subsection, is a reconstruction $\widetilde{\Sigma} \cdot \widetilde{V}^T$ of the eigentextures of $\mathcal{M}$. Combined with $U_{\mathcal{S}}$, it forms an approximation $\widetilde{\mathcal{M}}$ of $\mathcal{M}$:

$$\mathcal{M} \approx \widetilde{\mathcal{M}} = U_{\mathcal{S}} \cdot \widetilde{\Sigma} \cdot \widetilde{V}^T. \qquad (7)$$

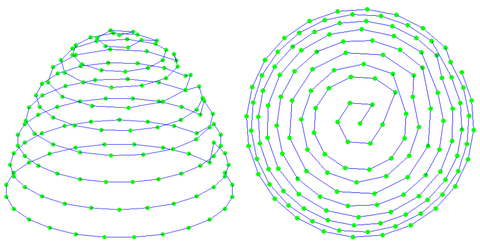
Algorithm 1 BTF Enlargement

```
Input: Preprocessed sample BTF $\mathcal{I}$, constraint set $\mathcal{C}$.
Output: Synthesized $\widetilde{\Sigma} \cdot \widetilde{V}^T$ for $\widetilde{\mathcal{M}}$.
  $\mathcal{R} \leftarrow$ Texture Optimization on $\mathcal{I}$ and $\mathcal{C}$;
  $\mathcal{R}' \leftarrow \emptyset$;
  while $\mathcal{R} \neq \mathcal{R}'$ do
    $\mathcal{R}' \leftarrow$ Statistical Synthesis on $\mathcal{R}$;
    $\mathcal{R} \leftarrow$ Quadratic Programming on $\mathcal{R}$ and $\mathcal{R}'$;
  end while
  $\Sigma V \leftarrow$ Remove constraint information from $\mathcal{R}'$;
  return $\Sigma V$.
```

3.2 Input Generation

All inputs for ground truth reconstruction are extracted from a matrix $\mathcal{M}$ containing the raw data of an angularly and spatially complete BTF measurement captured using the Dome I device described by Müller *et al.* [Mül05]. Materials are sampled using 151 lighting and viewing directions, resulting in 22 801 textures stored in $\mathcal{M}$. The full sampling is sketched in Fig. 4, where each green dot denotes the position of a camera with built-in flashlight.

From $\mathcal{M}$, we extract a subset $\mathcal{S}$ cropped in the spatial domain, taking the role of the representative measure-



(a) Viewed from the side. (b) Viewed from top.

Figure 4: Positions of the 151 cameras and light sources in the Dome I BTF acquisition device.

ment. It is compressed using singular value decomposition, as sketched in Fig. 2:

$$\mathcal{S} \approx U_{\mathcal{S}} \cdot \Sigma_{\mathcal{S}} \cdot V_{\mathcal{S}}^{T}. \tag{8}$$

The input sample for Texture Optimization is then given by

$$Z = \Sigma_{\mathcal{S}} \cdot V_{\mathcal{S}}^{T}. \tag{9}$$

Also from $\mathcal{M}$, a set $\mathcal{C}$ is extracted, consisting of images in full spatial resolution, but only from few lighting and viewing directions. These are used as the constraint images. For the synthesis algorithm, the input sample also has to contain a constraint set $\mathcal{C}_{\mathcal{S}}$ corresponding to $\mathcal{C}$. In a real-world scenario, the images constituting $\mathcal{C}_{\mathcal{S}}$ should ideally be taken under the same conditions as those in $\mathcal{C}$. For our experiments, $\mathcal{C}_{\mathcal{S}}$ is extracted from $\mathcal{C}$ by spatial cropping and then stacked together with $\Sigma_{\mathcal{S}} \cdot V_{\mathcal{S}}$ to form a multi-channel image serving as the input sample $\mathcal{I}$ in Algorithm 1.

The cropping of $\mathcal{M}$ used to extract $\mathcal{S}$, and also of $\mathcal{C}$ to extract $\mathcal{C}_{\mathcal{S}}$ is illustrated in Fig. 3b. The inner region, bordered green, depicts the spatial region from which $\mathcal{S}$, spanning 256×256 ABRDFs, is taken, while the yellow bordered outer region is the area to reconstruct. From this region, sized 512×512 ABRDFs, the images in $\mathcal{C}$ are extracted.

Naturally, one would not want to take too many photographs, so we only considered samplings containing no more than ten images:

1. Top light, top view image as only constraint, and
2. a sampling consisting of the top light, top view perspective plus nine randomly chosen combinations of lighting and viewing directions.

To simulate the situation of taking the constraint photographs under diffuse daylight, we also evaluate the use of a *mean color image* as constraint. One motivation for this idea is the possible effect that from some lighting directions, highlights might mask the material's structure such that, for a bad choice of constraint images, the Texture Optimization algorithm might still lack information for a faithful reproduction of these features. The constraint image c is constructed from the material measurement by fixing one viewing position (top view in our case) while averaging between the color values for all viewing directions:

$$c(x,y) = \sum_{(\theta_l,\phi_l \in \Omega)} \mathcal{M}(x,y,\theta_l,\phi_l,\theta_v,\phi_v), \tag{10}$$

with θ_v, ϕ_v fixed, and Ω the set of all lighting and viewing angles from which $\mathcal{M}$ is sampled. An example image generated for a piece of wood is shown in Fig. 5.

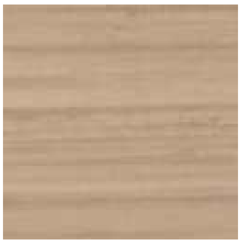


Figure 5: Example of a mean color image generated from a BTF measurement for a piece of wood. (Brightness enhanced for viewing.)

4 EXPERIMENTAL RESULTS

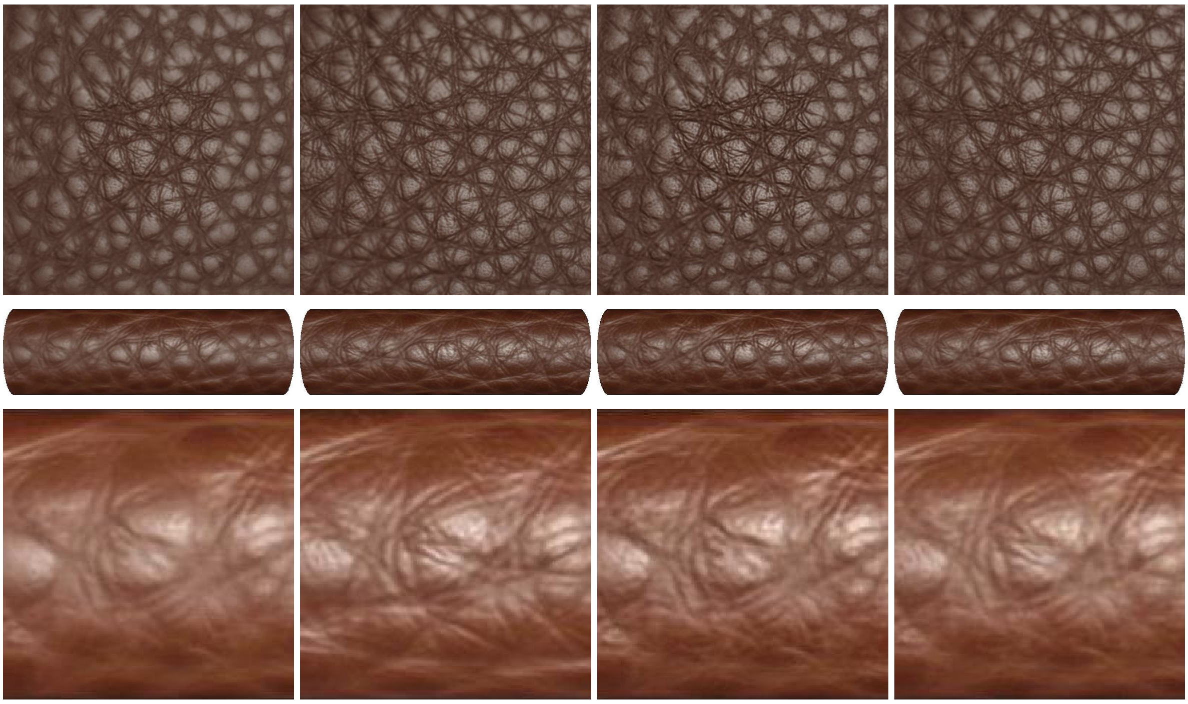
4.1 Single Constraint

All computations were performed on a desktop computer with Intel Core i7-2600K CPU at 3.4 GHz and 16 GB of RAM, using a C++ implementation of Texture Optimization, a MATLAB MEX implementation of Statistical Synthesis and a MATLAB script for iterative optimization, all operating in 64 bit. Texture Optimization and statistical synthesis contain CUDA-accelerated computations performed on an NVIDIA GeForce GTX 560 Ti graphics adapter.

Our first experiments took place using only the three color channels of the top light, top view image as constraints. An example for the reconstruction of a leather measurement is shown in Fig. 6a, with the compressed reference in Fig. 6b. One observation from these images is the severe blurring outside of the measured region of the reconstructed BTF. Fig. 6c shows that the additional step of statistical synthesis after texture optimization indeed helps deblurring the result, but at the cost of oversharpening in the sample region. After performing the additional optimization step, the result appears to be visually much closer to the ground truth data, as can be observed in Fig. 6d.

Running times for these results are about one hour for Texture Optimization plus additional time of approximately 20 minutes for the iterative optimization.

As one can see in the renderings onto a cylinder in the middle row of Fig. 6, details of the material's structure are lost for flat viewing and lighting angles. The effect is best visible when comparing the bottom thirds of



(a) Reconstruction (Texture Optimization only)

(b) Reference

(c) Reconstruction after Statistical Synthesis

(d) Reconstruction after iterative optimization

Figure 6: Comparison of the reconstruction results for a sample of leather, rendered both onto a flat surface (top row) and onto a cylinder (middle row). The bottom row shows a detail of the cylinder with enhanced contrast to make differences more recognizable.

the zoomed and contrast enhanced images in the bottom row. This behaviour is exactly what one would expect, as the synthesis algorithm lacks information on how to choose the correct color values for those angles when only one constraint image is provided.

4.2 Multiple and Alternative Constraints

An obvious idea is to add more constraint images, which corresponds to taking more photographs in a real-world setting. Thus, this subsection demonstrates the change in result quality when rising the number of constraint images from one to ten.

To enable a faster evaluation, we switched to the setting in Fig. 3c when producing the results in the remainder of this paper. Here, the sample area as well as the region to be reconstructed span 128×128 ABRDFs each, leading to computation times of about half an hour for Texture Optimization and iterative optimization together.

Result images are given in Fig. 7c. Comparison with the images rendered using only one constraint image (see Fig. 7b) suggest that additional constraints can indeed help improve details of the structure, but not without cost. Some aspects of the structure are still not met correctly, *e.g.* the results appear to be noisier now, especially the carpet.

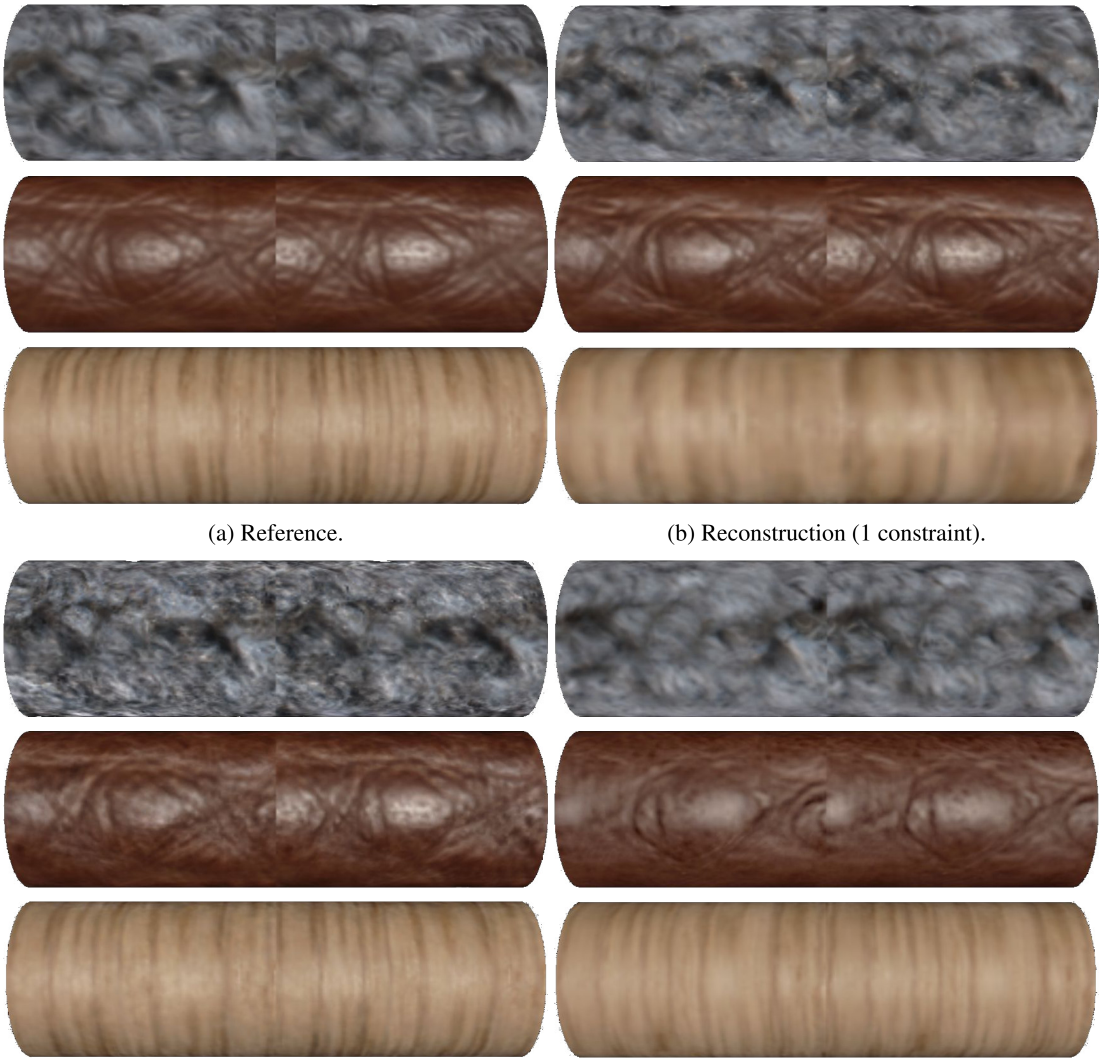
One approach to avoid these drawbacks is the use of a mean color image as a constraint, as proposed in Section 3.2. Here, the contribution of a single light is only $^1/_{151}$, such that theoretically, a highlight in an unfortunate position cannot introduce a too large error onto the overall reconstruction quality. When inspecting the result images in Fig. 7d, the results again look blurred, and a change in color or brightness has occurred. Thus, using mean color images as only constraint might not be the right choice, but it might still prove helpful in combination with different kinds of constraints.

As a numeric measure for the reconstruction error, we use the *average ABRDF RMSE* as also used by Ruiters *et al.* [Rui09]:

$$E = \frac{1}{n} \cdot \sum_{i=1}^{n} \sqrt{\frac{||M_i - \widetilde{M}_i||^2}{m}}, \tag{11}$$

with M_i the i-th ABRDF of the reference BDI matrix, $\widetilde{M}_i$ the i-th ABRDF of the reconstructed BTF matrix, n the number of ABRDFs of M (and $\widetilde{M}$, respectively) and m the number of components of each ABRDF vector. It must be pointed out that this error measure cannot be used to compare between reconstruction quality for *different* materials.

Error values for the reconstruction results illustrated in this section are given in Table 1. Note that the reference also exposes an error $E_{\text{ref}} > 0$, as we are comparing compressed BTFs with 64 components to the uncompressed data. We observe that in most cases, the



(a) Reference.

(b) Reconstruction (1 constraint).

(c) Reconstruction (10 constraints).

(d) Mean color image as only constraint.

Figure 7: Comparison of reconstruction results for carpet, leather and wood (from top to bottom). Constraints were: Top light, top view (Fig. 7b), random sampling (Fig. 7c) and mean color image (Fig. 7d). All images show the results after two phases of iterative optimization.

value of reconstruction error increases with rising effort in constraint generation. For the results with ten constraint images, this might be surprising, but it suits to the observation that they look less blurred, but noisier than those produced using only one constraint image.

5 DISCUSSION AND OUTLOOK

In this paper, we have demonstrated how a measurement of only a subset of a material's surface can be completed to obtain a BTF for the full material sample. It is obvious that even with our method working only on compressed data, one will soon reach the limitations of existing computer systems in terms of memory consumption and computational power.

Thus, one step toward the applicability of our ideas to larger problem instances has to be the development of faster reconstruction methods, *e.g.* based on fitting linear models. Ideally, all information needed would be preprocessed and stored in a more compact fashion, allowing fast on-demand reconstruction, ideally directly on the graphics hardware, instead of precomputing several mega- to gigabytes of reflectance data.

Also, the concept of *level-of-detail* should be considered: If only one BTF is visible, probably the object onto which it is mapped is viewed from a nearby viewing point, demanding all available details to be rendered. On the other hand, if multiple BTFs are visible in one scene, it is not implausible to assume that neither of them needs to be kept in memory in full detail.

When sticking to Texture Optimization as a method for BTF enlargement, there are also several entry points for further evaluation. A deeper understanding of the reasons for the blurring and the exaggerated sharpness

Table 1: Average ABRDF RMSE values for carpet, leather and wood reconstructions. Column legend: (1): Texture Optimization only, (2): Texture Optimization plus Statistical Synthesis, (3): Optimized.

(a) Carpet: $E_{\text{ref}} = 0.0095$.

	(1)	(2)	(3)
1 Constr.	0.0253	0.0306	0.0303
Random	0.0215	0.0262	0.0423
Mean Color	1.0030	1.0097	1.0079

(b) Leather: $E_{\text{ref}} = 0.0069$.

	(1)	(2)	(3)
1 Constr.	0.0160	0.0182	0.0185
Random	0.0153	0.0175	0.0238
Mean Color	1.0022	1.0017	1.0020

(c) Wood: $E_{\text{ref}} = 0.0029$.

	(1)	(2)	(3)
1 Constr.	0.0132	0.0161	0.0158
Random	0.0117	0.0151	0.0155
Mean Color	1.0072	1.0058	1.0064

might help to simplify the reconstruction method even to the point where the additional optimization step becomes obsolete. A systematical evaluation of the influence of different values for the weighting coefficient λ in Eq. 3 might be another aspect.

It is also not clear if a different method for preparing the representative measurement could provide improvements. This also includes questioning the use of logarithmic range reduction during the compression step.

As we have seen from the results in section 4, the choice of constraint images has a large impact on the visual quality of the reconstruction results. Thus, an extensive investigation on the choice of suitable samplings, or, more general, methods for constraint generation, might help improving result quality or reduce the number of images needed to be taken. Aspects under consideration should contain convenience of acquisition as well as a reduction of processing effort and memory consumption.

A major challenge which still has to be tackled is the step from our idealized input data to a more realistic setting. Currently, all inputs originate from existing BTF measurements. Especially the constraint images are already resampled and reprojected, such that all pixels are really taken from the same viewing angle and illuminated from the same lighting angle, which is not true for real photographs. Thus, one would either have to implement a constraint acquisition method exhibiting exactly these properties, or to perform preprocessing on real-world data to match the requirements.

6 ACKNOWLEDGMENTS

This work was developed in the graduate school on digital material appearance funded by X-Rite Inc. We would like to thank Gero Müller, Martin Rump and Roland Ruiters for helpful advice.

7 REFERENCES

[Dan99] Dana, K. J., Van Ginneken, B., Nayar, S. K., and Koenderink, J. J. Reflectance and texture of real-world surfaces. In ACM Transactions on Graphics (TOG), 18(1):pp. 1–34, 1999.

[Don10] Dong, Y., Wang, J., Tong, X., Snyder, J., Lan, Y., Ben-Ezra, M., and Guo, B. Manifold bootstrapping for SVBRDF capture. In ACM Transactions on Graphics (TOG), 29(4):p. 98, 2010.

[Don13] Dong, Y., Lin, S., and Guo, B. *Material Appearance Modeling: A Data-Coherent Approach.* Springer-Verlag New York Incorporated, 2013. ISBN 9783642357763.

[Fur05] Furukawa, R., Harada, M., Nakamura, Y., and Kawasaki, H. Synthesis of textures with intricate geometries using BTF and large number of textured micropolygons. In *Proc. of the 4th International Workshop on Texture Analysis and Synthesis*, pp. 77–82. Citeseer, 2005.

[Hai05] Haindl, M. and Hatka, M. BTF roller. In *Proceedings of the 4th International Workshop on Texture Analysis and Synthesis*, pp. 89–94. 2005.

[Hai13] Haindl, M. and Filip, J. *Visual Texture: Accurate Material Appearance Measurement, Representation and Modeling*. Springer, 2013.

[Kwa05] Kwatra, V., Essa, I., Bobick, A., and Kwatra, N. Texture optimization for example-based synthesis. In *ACM Transactions on Graphics (TOG)*, vol. 24, pp. 795–802. ACM, 2005.

[Leu07] Leung, M.-K., Pang, W.-M., Fu, C.-W., Wong, T.-T., and Heng, P.-A. Tileable BTF. In Visualization and Computer Graphics, IEEE Transactions on, 13(5):pp. 953–965, 2007.

[Mül05] Müller, G., Meseth, J., Sattler, M., Sarlette, R., and Klein, R. Acquisition, synthesis, and rendering of bidirectional texture functions. In *Computer Graphics Forum*, vol. 24, pp. 83–109. Wiley Online Library, 2005.

[Por00] Portilla, J. and Simoncelli, E. P. A parametric texture model based on joint statistics of complex wavelet coefficients. In International Journal of Computer Vision, 40(1):pp. 49–70, 2000.

[Rui09] Ruiters, R., Rump, M., and Klein, R. Parallelized matrix factorization for fast BTF compression. In *Proceedings of the 9th Eurographics conference on Parallel Graphics and Visualization*, pp. 25–32. Eurographics Association, 2009.

[Rui10] Ruiters, R., Schnabel, R., and Klein, R. Patch-based texture interpolation. In *Computer Graphics Forum*, vol. 29, pp. 1421–1429. Wiley Online Library, 2010.

[Rui13] Ruiters, R., Schwartz, C., and Klein, R. Example-based interpolation and synthesis of bidirectional texture functions. In Computer Graphics Forum (Proceedings of the Eurographics 2013), 32(2):pp. 361–370, 2013.

[Sch13] Schwartz, C., Sarlette, R., Weinmann, M., and Klein, R. DOME II: A parallelized BTF acquisition system. In *Eurographics Workshop on Material Appearance Modeling*, pp. 25–31. The Eurographics Association, 2013.

[Sch14] Schwartz, C., Sarlette, R., Weinmann, M., Rump, M., and Klein, R. Design and implementation of practical bidirectional texture function measurement devices focusing on the developments at the university of Bonn. In Sensors, 14(5), 2014. ISSN 1424-8220.

[Sue00] Suen, P.-h. and Healey, G. The analysis and recognition of real-world textures in three dimensions. In Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(5):pp. 491–503, 2000.

[Ton02] Tong, X., Zhang, J., Liu, L., Wang, X., Guo, B., and Shum, H.-Y. Synthesis of bidirectional texture functions on arbitrary surfaces. In *ACM Transactions on Graphics (TOG)*, vol. 21, pp. 665–672. ACM, 2002.

[Wei09] Wei, L.-Y., Lefebvre, S., Kwatra, V., Turk, G., et al. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR*, pp. 93–117. 2009.

[Zho05] Zhou, K., Du, P., Wang, L., Matsushita, Y., Shi, J., Guo, B., and Shum, H.-Y. Decorating surfaces with bidirectional texture functions. In Visualization and Computer Graphics, IEEE Transactions on, 11(5):pp. 519–528, 2005.

Patch-based sparse reconstruction of material BTFs

Dennis den Brok Heinz Christian Steinhausen Matthias Hullin Reinhard Klein
Universität Bonn
Institute of Computer Science II
Friedrich-Ebert-Allee 144
53113 Bonn, Germany
{ denbrok, steinhau, hullin, rk }@cs.uni-bonn.de

ABSTRACT

We propose a simple and efficient method to reconstruct materials' *bidirectional texture functions* (BTFs) from angularly sparse measurements. The key observation is that materials of similar types exhibit both similar surface structure and reflectance properties. We exploit this by manually clustering an existing database of fully measured material BTFs and fitting a linear model to each of the clusters. The models are computed not on per-texel data but on small spatial BTF patches we call *apparent BTFs*. Sparse reconstruction can then be performed by solving a linear least-squares problem without any regularization, using a per-cluster sampling strategy derived from the models. We demonstrate that our method is capable of faithfully reconstructing fully resolved BTFs from sparse measurements for a wide range of materials.

Keywords
bidirectional texture functions, sparse acquisition, material appearance

1 INTRODUCTION

In many applications, it is desirable or even imperative to reproduce a material's appearance faithfully and, possibly, in real-time. For a wide range of materials *bidirectional texture functions* (BTFs) – loosely speaking, an image-based variant of the better known *spatially varying bidirectional reflectance distribution functions* (SVBRDFs) – provide good reproduction quality, even at interactive frame rates. The acquisition of high-quality, high-resolution BTFs of real-world materials is, however, by many means expensive. In particular, measurement times of typically many hours per material make it very cumbersome to obtain large BTF databases, as pictures of the material to be measured have to be taken from many different viewing angles and under many different lighting conditions.

We propose a simple and efficient method for the sparse acquisition of material BTFs, assuming a sufficiently large and heterogenous database of fully measured materials is available:

We demonstrate that linear models describing material reflectance per texel are insufficient for this task because effects not local to texels frequently occur. We show that, instead, small BTF patches we call *apparent BTFs* (ABTFs) provide a suitable foundation for such models. In order to account for the high variance of material surfaces, we propose to fit models to patches clustered by semantic material class. From these models, sparse sampling strategies can be deduced that take advantage of the peculiarities of existing BTF acquisition devices. Reconstruction from such sparse measurements can then be achieved efficiently by solving a simple linear least-squares problem without regularization.

We demonstrate that our method is able to reconstruct fully resolved material BTFs of good quality from as little as 6% of the original samples. It can be used for substantially improving acquisition times or angular resolution, thus benefiting the most common BTF acquisition devices.

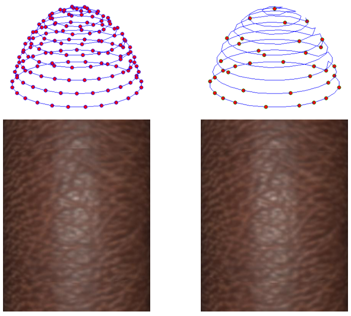


Figure 1: Left: Reference rendering along with the full sampling. Right: Sparse sampling used to produce the rendering of a sparse reconstruction shown below.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

2 BACKGROUND

2.1 Bidirectional texture functions

BTFs have been introduced by Dana *et al.* [1] as an image-based approach to spatially varying appearance. Like SVBRDFs, they are 6-dimensional functions of the form

$$\mathscr{B}(\mathbf{x}, \omega_i, \omega_o),$$

where $\omega_{i,o} \in \mathbf{R}^2$ are the incoming and outgoing light directions, respectively, and $\mathbf{x} \in \mathbf{R}^2$ is the position on a parameterized surface V. In the case of material BTFs, V is typically flat; it does not need to coincide with the material's actual surface geometry. It is generally assumed that light sources are directional and have the same spectrum. In particular, effects such as phosphorescene, fluorescence and subsurface scattering cannot be captured accurately.

The fundamental difference from SVBDRFs is that the function $\mathscr{B}(\mathbf{x}, -)$ need not be BRDF-valued: the corresponding per-texel reflectance function does not need to adhere to Helmholtz reciprocity and conservation of energy and is, therefore, capable of capturing non-local effects such as interreflections and self-shadowing. Moreover, because V does not necessarily coincide with the actual surface, the per-texel reflectance functions may also describe parallax effects. For these reasons, the term *apparent BRDF* (ABRDF) has been suggested by Wong *et al.* [18] for this kind of functions. Conversely, the values of the function $\mathscr{B}(-, \omega_i, \omega_o)$ are just 2D textures corresponding to specific pairs of incoming and outgoing light directions.

2.2 BTF acquisition

Several setups for the acquisition of BTFs have been proposed. We briefly review the most prominent paradigms, as our method benefits all of them to a greater or lesser extent. An in-depth overview can be found in [16].

2.2.1 *Gonioreflectometer*

In what is historically the first BTF acquisition setup, proposed by Dana *et al.* [1], the material sample is placed on a turntable, and a camera and a light source held by robot arms are moved across the hemisphere above the sample to capture images of the sample under different lighting and viewing conditions. The gonioreflectometer is very flexible in terms of possible samplings of the hemisphere, but measurement times are excessive – on the order of weeks for a moderate angular resolution – due to the little amounts of light sources and sensors and the movable parts' low speeds.

2.2.2 *Kaleidoscope*

Han *et al.* [7] introduced an intriguing parallel setup: The sample is placed underneath a tapered kaleidoscope, lit and captured from a projector and a camera placed at the other end, which allows for a number of lighting and viewing conditions to be measured in a single camera shot. By appropriately arranging the mirrors, the angular and spatial resolution can be adjusted; however, both are typically rather low, and increasing one leads to a decrease of the other, so there is always a tradeoff to be made.

2.2.3 *Camera domes*

Camera domes as proposed, for instance, by Müller *et al.* [11] and Schwartz *et al.* [15] ideally provide a highly parallel means to acquire BTFs: A number of cameras is spread across the hemisphere above the sample holder. Their flashes or separate LEDs are used as light sources. Parallelism may be traded for fewer cameras and lower cost by placing the sample on a turntable in order to achieve a similarly dense sampling of the hemisphere. Due to the number of cameras, data transfer times become a new bottleneck.

In all of the above setups, it is usually necessary to capture the same scene several times with different shutter times in order to obtain HDR data.

2.3 Related work

To the best of our knowledge, no method for sparse reconstruction of entire BTFs has been proposed so far. There exists, however, a number of methods for lower-dimensional reflectance models:

In [10], Matusik *et al.* perform *singular value decomposition* (SVD) on a database of 100 measured BRDFs of a wide range of isotropic materials to obtain a linear model.

In [9], the same authors introduce two methods for sparse reconstruction of isotropic BRDFs: The first method is based on a wavelet analysis of their BRDF database. A set of basis wavelets termed *common wavelet basis* is determined and used to reconstruct previously unseen BRDFs with approximately 1.5 million samples from approximately 70000 measurements. The second method uses the entire BRDF database itself as a linear model for reconstruction of fully measured BRDFs from as little as 800 out of the original approximately 1.5 million samples, at the cost of slightly increased reconstruction errors and the required availability of the BRDF database. Samples are chosen using a simple optimization algorithm such that the linear system to be solved for reconstruction is well-conditioned. They do not investigate how well their methods generalize to more complex reflectance such as anisotropic BRDFs or ABRDFs.

In [2], Dong *et al.* reconstruct a material's SVBRDF from a sparse measurement using a manifold constructed from analytical BRDFs fit to fully measured BRDFs of manually selected representative points on the material's surface. The algorithm is unlikely to scale to BTFs because of the typically much higher intrinsic dimensionality of the ABRDF manifold (*cf.* section 4.1.1). A generalization to previously unseen materials is not obvious, albeit conceivable.

Peers *et al.* [13] introduce *compressed sensing* [3] to the acquisition of *reflectance fields*, assuming both 2D *outgoing* (here: fixed viewing direction) and *incident light fields*. Their algorithm uses a hierarchical, multi-resolution Haar wavelet basis, taking spatial coherence into account. It is not clear how to extend this approach to a multi-view setup. Common BTF acquisition setups only have a very limited number of light sources, where the advantage of compressed sensing might be negligible. Due to these light sources' brightness, we expect shot noise to become a problem.

Conversely, in [8], Marwah *et al.* use sparsity-based methods related to compressed sensing in order to sparsely acquire 4D light fields with an angular resolution of 5×5. They compute a dictionary of what they call *light field atoms* – 11 × 11 spatial light field patches which allow for a sparse representation of natural light fields. Such a dictionary does not exist in the case of ABRDFs or ABTFs; as demonstrated in section 4.1.1, their dimensionality is likely too high.

Filip *et al.* [5] propose a vector quantization of BTFs for the purpose of compression, guided by a psychophysically validated metric. They conclude that as little as 10 – 35 % of the original textures are sufficient to maintain the same visual appearance in renderings. It would be interesting to investigate whether there is a common quantization for *all* materials, and if so, whether it could be used for sparse acquisition. A large user study would be needed in order to adapt the metric to a bigger BTF database.

3 LINEAR MODELS FOR MATERIAL BTFS

During measurement, a finite discretization of the measured material's BTF $\mathscr{B}$ is obtained. After rectification of the acquired images, the discrete BTF has a natural representation as a matrix $\mathbf{B} \in \mathbf{R}^{n \times m}$ with the columns representing the m discrete ABRDFs, each entry corresponding to some pair (ω_i, ω_o) of incoming and outgoing angle, and the rows representing the n rectified textures (*cf.* figure 2).

3.1 Linear models & reconstruction

The goal is now to recover $\mathbf{B}$ from a sparse measurement

$$\widetilde{\mathbf{B}} = \mathbf{M}\mathbf{B}$$

of n_s samples, where $\mathbf{M} \in \mathbf{R}^{n_s \times n}$ is a *measurement matrix*, typically binary, which determines the sparse sampling.

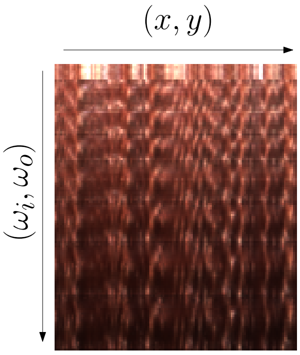


Figure 2: Representation of a discretized BTF as a matrix.

Arguably one of the most simple methods to attack this problem is, given fully resolved training data $\mathbf{D}$, to fit a linear model $\mathbf{D} \approx \mathbf{U}\mathbf{C}$. An optimal fitting method in terms of L^2 error is to compute a truncated SVD

$$\mathbf{D} \approx \mathbf{U}\Sigma\mathbf{V}^t$$

as established by the Eckart-Young theorem [4]. The hope is that the model both

- generalizes to previously unknown data; *i.e.*

$$\min_{\mathbf{C_B}} \|\mathbf{U}\mathbf{C_B} - \mathbf{B}\| < \varepsilon \tag{1}$$

- is expressive enough that a sparse sampling is sufficient to find reasonable coefficients; *i.e.*

$$\min_{\mathbf{C_B}} \|\mathbf{M}(\mathbf{U}\mathbf{C_B} - \mathbf{B})\| < \delta \implies \|\mathbf{U}\mathbf{C_B} - \mathbf{B}\| < \varepsilon \tag{2}$$

Provided $\widetilde{\mathbf{B}}$ has at least as many rows as columns, an approximation of $\mathbf{B}$ may then be obtained via

$$\mathbf{B} \approx \mathbf{U}(\mathbf{M}\mathbf{U})^{\dagger}\widetilde{\mathbf{B}}, \tag{3}$$

where $(\mathbf{M}\mathbf{U})^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{M}\mathbf{U}$.

It is well-known that the fitting of linear models through minimization of L^2 error is sensitive to outliers. In order to decrease the influence of specular highlights, we reduce the data's dynamic range by converting the measured HDR RGB data to YUV color space, dividing the U and V values by the corresponding Y value and applying log to the Y values.

Despite its simplicity, this approach has been demonstrated in [9] to be quite effective in the special case of isotropic BRDFs. It seems thus worthwhile to investigate whether this generalizes to ABRDFs.

3.2 Linear models for ABRDFs

Linear models for ABRDFs are already being used for compression and rendering of BTFs, often under the moniker *full matrix factorization* (FMF). In that case, models are fit to a certain material's ABRDFs only; *i.e.* to **B** instead of a whole database **D**. The columns of **U** and **V** are commonly refered to as *eigen-ABRDFs* and *eigentextures*, respectively [12], in reference to their semantic meaning.

It is reasonable to assume that for BTFs the ABRDFs of which are close to being true BRDFs, a linear model may perform similarly well as in [9] [10] with respect to equations 1 and 2. However, as soon as surface structure becomes significant, reconstructions from sparse measurements might easily miss effects such as self-shadowing, interreflections, occlusion and parallax. We shall demonstrate in section 4.2 that this is indeed the case.

3.3 Linear models for ABTFs

In order to overcome these problems, we take spatial information into account: instead of considering only ABRDFs, we consider entire collections of neighboring ABRDFs, which we call *apparent BTFs* (ABTFs), as similar to ABRDFs they capture effects not local to the specific patch such as interreflections or shadows cast from neighboring patches.

The matrix **B** then takes on a different form, with its columns representing discrete ABTFs, for instance as vectors of stacked discrete ABRDFs belonging to the same neighborhood. The corresponding measurement matrix becomes $\mathbf{1}_{p^2} \otimes \mathbf{M}$, where p denotes the spatial patch size and $\otimes$ the Kronecker product.

Note that an alternative to patches exists in the form of appropriate filter banks, as *e.g.* demonstrated by Peers *et al.* [13] A case has been made in favor of the simpler spatial patches by Varma *et al.* [17], albeit in the case of material classification: the authors demonstrate that classification using spatial patches, which can be as small as 3×3 texels, is superior to that using filter banks with equivalent support.

The intrinsic dimensionality of the ABTF database is likely higher than that of the ABRDF database; in the worst case by a factor equal to the patch size. To mitigate this to some extent, we propose to cluster the database such that each cluster contains only materials with similar surface structure, and determine the linear models $\mathbf{D}_{\text{cluster}} \approx \mathbf{U}\Sigma\mathbf{V}^t$ per cluster. The columns of **U** shall be called *eigen-ABTFs*.

3.4 Sampling strategies

Once a model satisfying equation 1 has been established, a measurement matrix **M** that takes advantage of the model needs to be devised. We chose to implement the simple optimization algorithm proposed in [10]: $\mathbf{M} \in \mathbf{R}^{n_s \times n}$ is initialized as random binary matrix with precisely one 1 on each row. The algorithm then randomly replaces one row of **M** with a different random binary unit row vector. If the condition number $\kappa(\mathbf{MU})$ does not decrease, the change is reverted. This is repeated until convergence or a maximum number of steps is reached (*cf.* algorithm 1). For ABTFs, the condition number $\kappa((\mathbf{1}_{p^2} \otimes \mathbf{M})\mathbf{U})$ is tested instead.

The intuition behind this choice is that the condition number $\kappa(\mathbf{MU})$ is an indicator of how robustly **MU** can be inverted; *i.e.* of how well coefficients $\mathbf{C_B}$ as in equation 2 can be found.

In its present form the algorithm is free to choose whatever pairs of incoming and outgoing light directions lead to well-conditioned linear system. This approach suits best the gonioreflectometer setup, where all such pairs have equal costs. The algorithm can easily be modified to take the parallelism of camera dome setups into account.

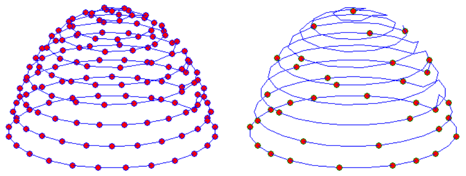
While undersampling could be used in the kaleidoscope setup as well, we argue it is more beneficial to use the proposed method in order to increase the kaleidoscope's limited angular resolution.

Algorithm 1 Generation of a measurement matrix.

```
Input: desired number $n_s$ of samples
Output: optimized measurement matrix $\mathbf{M} \in \mathbf{R}^{n_s \times n}$
  $\mathbf{M} \leftarrow$ random binary with exactly one 1 per row
  while not converged do
    $\mathbf{M}' \leftarrow \mathbf{M}$
    $\mathbf{r} \leftarrow$ random binary row vector with $\|\mathbf{r}\|_0 = 1$
    random row of $M \leftarrow \mathbf{r}$
    if $\kappa(\mathbf{M}'\mathbf{U}) < \kappa(\mathbf{MU})$ then
      $\mathbf{M} \leftarrow \mathbf{M}'$
    end if
  end while
  return M
```

4 RESULTS

For our experiments, we used an existing database of high-quality measured BTFs. The measurement device used to create the database is a camera dome with 151 cameras, the flashes of which are used as light sources, resulting in an angular resolution of 151×151 (*cf.* figure 3a). The rectified textures have a spatial resolution of 512×512 pixels and correspond to a part of the sample approximately 4 cm × 4 cm in size. The database consists of 14 semantic classes with 12 materials each. We selected the classes *carpet*, *cloth*, *gravel*, *leather*, *metal*, *stone*, *wall tile*, *wallpaper* and *wood*, which exhibit significant inter- and intraclass variance. We used 11 materials per class for fitting the linear models and the remaining material per class for the purpose of validation.



(a) (b)

Figure 3: (a) Sketch of the acquisition setup. The red points correspond to both light sources and cameras. (b) Sketch of the 7 × 7 parabolic map sampling. The red points correspond to both light sources and cameras.

All computations have been performed using MATLAB 2011b under Windows XP on a machine with two Intel Xeon E5645 processors and 144 GB of RAM.

4.1 Model fitting

Computing at once a truncated SVD for either of the entire database, or even single clusters thereof, is prohibitive due to the computation time required. We therefore used *eigenspace merging* to compute the SVD hierarchically; *i.e.* we first use EM-PCA (*cf.* [14]) to obtain approximate truncated SVDs of the single BTFs and subsequently merge the resulting eigenspaces (*cf. e.g.* [6]). In order to further reduce computation times, we cropped the BTFs to a spatial extent of 128 × 128 texels. For the purpose of comparison, we fit linear models both to the ABRDF and the ABTF database.

4.1.1 ABRDFs

For a single one of our BTFs, 200 eigen-ABRDFs for the $\log(Y)$ channel and 100 eigen-ABRDFs for each the U/Y and the V/Y channel provide a very high reproduction quality. We merged the resulting eigenspaces first per cluster and then globally to obtain an ABRDF basis of 2048 eigen-ABRDFs. The entire process takes approximately 30 minutes per cluster, including disk I/O and the color space transformation, hence about 4.5 hours altogether.

Table 1 shows the relative projection errors ε that occur when projecting the $\log(Y)$ channel of the test material's BTF onto the corresponding bases for various numbers of basis ABRDFs; *i.e.*

$$\varepsilon = \frac{\|\mathbf{U}(\mathbf{U}'\mathbf{B}) - \mathbf{B}\|_\mathrm{F}}{\|\mathbf{B}\|_\mathrm{F}}$$

where $\|\cdot\|_\mathrm{F}$ denotes the Frobenius norm. For comparison, we include the relative projection errors for the fully measured BTFs after FMF-compression retaining 128 eigen-ABRDFs, a number suitable for high-quality real-time rendering. Typically, 1024 basis ABRDFs are sufficient to achieve good projection results, which is the lower limit on the number of samples necessary for sparse reconstruction via equation 3.

4.1.2 ABTFs

For ABTFs, we computed bases per cluster. Following the argument in [17], we used a spatial ABTF size of 3 × 3. For performance reasons, we selected ABTFs maximally without overlap, resulting in a database of 1764 ABTFs per material and 19404 ABTFs per cluster. We again first computed bases per-material, retaining 200 eigen-ABTFs for the $\log(Y)$ channel and 100 eigen-ABTFs for each the U/Y and the V/Y channel, and then merged the resulting eigenspaces per cluster. This process takes approximately 2 hours per cluster, or 18 hours in total.

Table 1 shows relative projection errors (*cf.* section 4.1.1) for the $\log(Y)$ channel for 1024 and 2048 basis ABTFs in comparison with errors for reconstructions from ABRDF-wise projections. The projections themselves were produced by collecting all possible 3 × 3 ABTFs from the test BTF and projecting them onto the appropriate cluster's basis. BTFs are obtained from this representation by computing the reconstruction and blending the patches, all texels weighted equally. Typically, 2048 basis ABTFs provide almost as good projections results as 1024 basis ABRDFs.

4.2 Reconstruction

Figure 4 shows renderings of BTFs reconstructed with the proposed method, table 2 the corresponding relative reconstruction errors

$$\varepsilon = \frac{\|\mathbf{U}((\mathbf{MU})^\dagger\widetilde{\mathbf{B}}) - \mathbf{B}\|_\mathrm{F}}{\|\mathbf{B}\|_\mathrm{F}}.$$

For comparison, we include renderings of the FMF-compressed original fully measured BTFs and their sparse reconstructions from ABRDF-wise linear models, along with the relative projection errors, which constitute lower limits for the relative reconstruction errors. BTFs were produced from ABTF-wise sparse reconstructions as described in section 4.1.2. We used two different sampling strategies: a 7 × 7 parabolic map mapped to the closest light and camera positions of the acquisition setup's full sampling, which may be considered a vague approximation of a kaleidoscope's sampling (*cf.* figure 3b), and optimized samplings with the same number of samples produced by algorithm 1. Both samplings consist of 1369 samples in total, or 6 % of the original 22801 samples.

4.2.1 ABRDFs

As predicted in section 3.2, ABRDF-wise reconstruction produces acceptable results only for materials with simple surface structure and reflectance – here: *stone* and *wood* – and even then only with the optimized sampling. *Leather* and *metal* already exhibit annoying artifacts; the results for even more complex materials are unsuitable for any practical purpose.

		# basis ABRDFs					# basis ABTFs	
class	FMF	128	256	512	1024	2048	1024	2048
carpet	5.4	6.2	5.6	5.1	4.6	4.0	5.3	5.1
cloth	2.7	4.0	3.2	2.6	2.1	1.7	2.9	2.7
gravel	5.0	8.0	6.7	5.6	4.7	4.0	5.4	4.9
leather	1.5	2.9	2.4	2.0	1.6	1.3	1.9	1.8
metal	1.0	3.2	2.5	2.1	1.8	1.6	2.7	2.2
stone	0.6	3.1	2.5	2.0	1.5	1.2	1.8	1.5
wall tile	0.4	5.6	4.7	3.8	2.9	2.1	2.2	1.7
wallpaper	2.7	5.2	4.4	3.6	3.0	2.4	3.4	3.1
wood	0.8	2.3	1.9	1.6	1.3	1.0	1.2	1.1

Table 1: $\log(Y)$ channel relative L^2 projection errors in percent for various numbers of basis ABRDFs. FMF: Projection onto per-material ABRDF basis with 128 eigen-ABRDFs.

	ABRDF			ABTF		
class	proj	pmap7	optimized	proj	pmap7	optimized
carpet	4.6	27.6	11.4	5.1	7.3	6.3
cloth	2.1	11.1	5.4	2.7	3.6	3.2
gravel	4.7	26.7	11.8	4.9	8.0	7.0
leather	1.6	8.0	4.0	1.8	2.3	2.1
metal	1.8	8.8	4.6	2.2	3.8	3.3
stone	1.5	6.3	3.4	1.5	2.3	2.3
wall tile	2.9	10.0	6.3	1.7	6.9	6.8
wallpaper	3.0	14.8	7.0	3.1	4.2	4.2
wood	1.3	6.3	2.9	1.1	1.6	1.5

Table 2: $\log(Y)$ channel relative L^2 reconstruction errors per cluster in percent. proj: Projection onto common basis. pmap7: Results for reconstruction from parabolic map sampling. optimized: Results for reconstruction from optimized sampling.

4.2.2 ABTFs

In contrast, even the non-optimized sampling is sufficient to produce convincing reconstructions of moderately complex materials using ABTF models. Where it is not, the optimized sampling often helps; only *gravel* and *wallpaper* exhibit perceivable artificats. The highlight of *wall tile* is not quite as sharp as it should be, and there are some artifacts in the highlight of *metal* visible mostly in the corresponding amplified error image (*cf.* figure 4f).

4.3 Limitations

While the proposed algorithm performs well in many situations, it has a number of limitations:

Most notably, it relies on the availability of a database of fully measured BTFs. Depending on the materials to be measured, that database must be quite encompassing; however, if *e.g.* only leathers are going to be measured, then a small database of a few measured leather BTFs might already be sufficient.

Without any regularization the lowest possible number of samples is precisely the number of basis ABTFs divided by the patch size. Typically, a greater number is necessary for robust results.

Reconstructions of material BTFs with highly complex surface structure may still suffer from artifacts visible in common lighting scenarios. It is not clear whether larger patch sizes could mitigate this. Even if so, this would likely lead to an undesirable significant increase of computation times and memory consumption.

For the same reason, the algorithm is constrained to moderate sampling rates. It would also be difficult to bootstrap a sufficiently large and heterogenous BTF database with substantially higher sampling rates.

5 CONCLUSION

We demonstrated the general possibility of efficient sparse acquisition of BTFs for a wide range of materials, provided a database of fully measured optically similar materials is available.

It would be interesting to investigate whether our results could be improved further. A possible approach is to further improve the linear bases, for instance by feature-aligning the ABTFs prior to fitting the models. It is also unclear how suitable our manual clustering of the database actually is. Automatic methods might be able to find a better optimization, possibly even consisting of fewer classes.

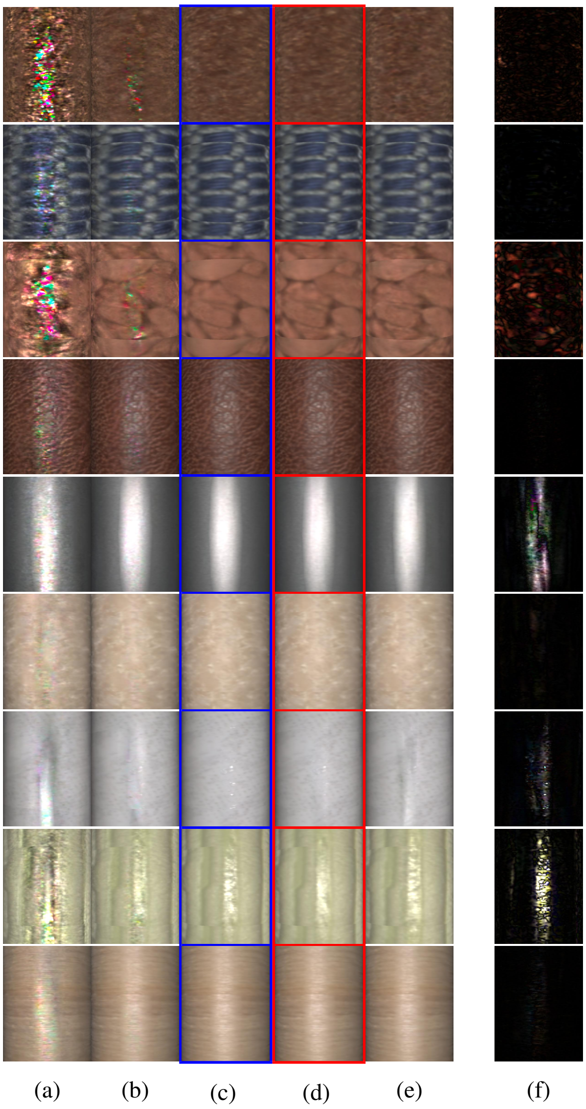


Figure 4: Renderings of reconstruction results.
(a) ABRDF-based reconstructions from 7 × 7 parabolic map sampling (1369 samples).
(b) ABRDF-based reconstructions from optimized sampling (1369 samples).
(c) FMF-compressed ground truth (22801 samples).
(d) ABTF-based reconstructions from optimized sampling (1369 samples).
(e) ABTF-based reconstructions from 7 × 7 parabolic map sampling (1369 samples).
(f) 20 × absolute differences between (c) and (d).

Moreover, our linear models might also be useful for purposes other than sparse reconstruction; for instance, it might be possible to use them to leverage the quality of BTF measurements produced with consumer-grade hardware, or under conditions less controlled than in the discussed setups.

Although the improvement in measurement cost is significant, the amount of samples needed still leaves room for further improvement. Depending on the material, it should not be impossible — at least given a rough estimate of the material's surface structure — to obtain satisfactory reconstruction results from less than 100 images. Both our experiments and compressed sensing theory suggest, however, that this barrier cannot be broken merely using unregularized linear methods. It thus seems worthwhile to investigate non-linear methods such as manifold learning or texture synthesis.

6 ACKNOWLEDGEMENTS

This work was funded by the X-Rite graduate school on Digital Material Appearance.

7 REFERENCES

[1] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18(1):1–34, January 1999.

[2] Yue Dong, Jiaping Wang, Xin Tong, John Snyder, Yanxiang Lan, Moshe Ben-Ezra, and Baining Guo. Manifold bootstrapping for SVBRDF capture. *ACM Trans. Graph.*, 29(4):98:1–98:10, July 2010.

[3] David L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52:1289–1306, 2006.

[4] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[5] Jiří Filip, Michael J. Chantler, Patrick R. Green, and Michal Haindl. A psychophysically validated metric for bidirectional texture data reduction. *ACM Trans. Graph.*, 27(5):138:1–138:11, December 2008.

[6] Peter Hall, David Marshall, and Ralph Martin. Merging and splitting eigenspace models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(9):1042–1049, September 2000.

[7] Jefferson Y. Han and Ken Perlin. Measuring bidirectional texture reflectance with a kaleidoscope. *ACM Trans. Graph.*, 22(3):741–748, July 2003.

[8] Kshitij Marwah, Gordon Wetzstein, Yosuke Bando, and Ramesh Raskar. Compressive light field photography using overcomplete dictionaries and optimized projections. *ACM Trans. Graph.*, 32(4):46:1–46:12, July 2013.

[9] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Trans. Graph.*, 22(3):759–769, July 2003.

[10] Wojciech Matusik, Hanspeter Pfister, Matthew Brand, and Leonard McMillan. Efficient isotropic BRDF measurement. In *Proceedings of the 14th Eurographics Workshop on Rendering*, EGRW

'03, pages 241–247, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[11] Gero Müller, Gerhard H. Bendels, and Reinhard Klein. Rapid synchronous acquisition of geometry and BTF for cultural heritage artefacts. In *The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*, pages 13–20. Eurographics Association, Eurographics Association, November 2005.

[12] Gero Müller, Jan Meseth, and Reinhard Klein. Compression and real-time rendering of measured BTFs using local PCA. In *Vision, Modeling and Visualisation 2003*, pages 271–280. Akademische Verlagsgesellschaft Aka GmbH, Berlin, November 2003.

[13] Pieter Peers, Dhruv K. Mahajan, Bruce Lamond, Abhijeet Ghosh, Wojciech Matusik, Ravi Ramamoorthi, and Paul Debevec. Compressive light transport sensing. *ACM Trans. Graph.*, 28(1):3:1–3:18, February 2009.

[14] Sam Roweis. EM algorithms for PCA and SPCA. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*, NIPS '97, pages 626–632, Cambridge, MA, USA, 1998. MIT Press.

[15] Christopher Schwartz, Ralf Sarlette, Michael Weinmann, and Reinhard Klein. DOME II: A parallelized BTF acquisition system. In *Eurographics Workshop on Material Appearance Modeling: Issues and Acquisition*, pages 25–31. Eurographics Association, June 2013.

[16] Christopher Schwartz, Ralf Sarlette, Michael Weinmann, Martin Rump, and Reinhard Klein. Design and implementation of practical bidirectional texture function measurement devices focusing on the developments at the university of bonn. *Sensors*, 14(5), April 2014.

[17] Manik Varma and Andrew Zisserman. A statistical approach to material classification using image patch exemplars. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(11):2032–2047, November 2009.

[18] Tien-Tsin Wong, Pheng-Ann Heng, Siu-Hang Or, and Wai-Yin Ng. Image-based rendering with controllable illumination. In *Proceedings of the Eurographics Workshop on Rendering Techniques '97*, pages 13–22, London, UK, UK, 1997. Springer-Verlag.