Online rapid prototyping of 3D objects using GPU-based 3D cloud computing: Application to 3D face modelling

Minh Nguyen, Patrice Delmas, Georgy Gimel'farb, Y.H. Chan, Alfonso Gastelum Strozzi
Department of Computer Science, The University of Auckland, New Zealand
p.delmas@auckland.ac.nz

Alexander Woodward
Ikegami Lab, University of Tokyo, Japan
alex@sacral.c.u-tokyo.ac.jp

Abstract

An on-line web application that interacts with an Internet user's 3D webcam (e.g. Minoru stereo web-camera) is described. The application instantly captures and processes stereo images to retrieve 3D object coordinates for further 3D modelling tasks. It offers on-demand semi-automatic camera calibration and automatic image pair rectification for further stereo matching operations. The reconstructed 3D objects are displayed on a Java3D panel with free observational navigation. The extracted 3D coordinates are saved for further use in a professional 3D graphical tool such as Blender. Real objects geometry was extracted from stereo images processed by a custom matching algorithm using either server-side (CUDA on high-end server) or client-side (Java Applet on user computer) processing. Specific applications in 3D face animation are described. Depth map and associated face texture were parsed into our animated 3D face system to rapidly generate specific facial expressions. The described application is portable, easy to set up and operate. Its current version is available at: `http://www.ivs.auckland.ac.nz`.

1 Introduction

Although computational stereo vision is used in many important practical areas and has been under development for decades, fast capturing of stereo images of real-world scenes and fast reconstruction of depicted 3D objects remain a challenging problem that requires complex and expensive hardware and software tools. Web-based tools enabling acquisition, processing and (3D) display of the resulting 3D information are very rare. Recently, a low cost commercial web-camera (Novo Minoru 3D webcam) was introduced [1] and allows to record and send anaglyphic 3D videos over the Internet with Windows Live Messenger, Skype, AOL Instant Messenger. Building on our previously published advances in web-based computer vision processing [2], we introduce a general and versatile web-interface using stereo web-cameras to seemingly acquire, process and display 3D data using GPU-based server for intense processing. Such unique interface offers internet users a simple on-line tool for applications ranging from the rapid prototyping of 3D object models, 3D look-alike avatar generation and animation and so-on. The next chapter introduces our new interface and discusses improvements from previous versions. Chapter 3 briefly describes the algorithms and techniques embedded in our interface to carry image acquisition, calibration and rectification, stereo matching and 3D model construction and display. Chapter 4 discusses a specific application namely 3D face model reconstruction and animation using real face depth measurements as computed with our stereo acquisition device. Chapter 5 concludes this paper.

2 Web interface rationale and design

2.1 Improvements over previous works

Our previous web-based on-line application [2] combined a dynamic PHP web-page environment and Java Applet technology for online computational stereo. The newly introduced 3D web camera brings a number of changes and extensions as detailed below.

- Instead of upload already captured stereo pairs, the new interface offers on-line acquisition from a 3D Minoru webcam controlled by a Macromedia Flash interface. This application allows the Internet user to freely capture stereo pairs at any moment on a local or distant camera via a web browser interface such as Internet Explorer or Mozilla Firefox.

- The user now has the choice between an auto-calibration and rectification process or a full calibration and rectification from calibration which provides line-wise accuracy. The calibration/rectification parameters for a specific camera (as recognised by its unique ID) are stored in our online server and can be re-used over time if the camera optical settings did not vary.

- A number of image pre- and post-processing operations are added including image smoothing, object segmentation, and gradient balancing to separate the object to be modelled from the background. This is primarily used for object 3D model prototyping and enhanced web browser visualisation.

- A server based repository of users' camera settings and calibration parameters as well as acquired images and 3D models is enabled via a CGI gateway application and available for third party usage.

All these main changes with respect to our previous interface [2] are further described below.

2.2 General web interface and design

Both the web interface, the allowed actions, and the database system are designed bearing the easiness of usage in mind. There is an introduction page listing the requirements to set up the camera along with a few tutorial videos. The gallery stores a history of image captures and reconstructed 3D data. The main application is fully automatic, so that the user can easily manage the whole processes from image acquisition to 3D model visualisation with just a few clicks. Basic procedural steps are enumerated below:

***Initial preparation*:** The user may use our previous auto-calibration procedure or our online calibration process involving a calibration object (e.g. a downloadable chess board) and a background object (e.g. a flat colour textured panel). The application ports Zhang et al. [9] open cv calibration code to our interface.

***Rectification*:** The user next builds on the calibration extrinsic and intrinsic parameters to effectuate a rectification from calibration as described in our previous work [7]. Rectification outcomes can be visualised on the interface. If rectification is satisfactory, calibration and rectification parameters are stored on our server.A unique camera ID provides an effective way to retrieve previously calibrated camera parameters for future usage.

***Object capturing*:** A stereo pair of images of a particular single object-of-interest placed in front of the background object is captured by the Minoru 3D webcam.

***3D surface acquisition*:** A dense depth map is reconstructed from the stereo pair captured (the map can be grey-coded for visualisation).

***3D object modelling*:** Slope reduction, smoothing and segmentation procedures are applied to separate the reconstructed object surface from the background. The resulting 3D model is displayed on a Java 3D screen.

***Saving 3D coordinates*:** The reconstructed 3D coordinates are saved for further usage in our online server.

3 Algorithms and programming

We further describe in this section the new components of our web-based stereo vision system e.g. webcam control, calibration and rectification for stereo vision. We then describe our GPU-server based processing and our automatic 3D modelling approach comprising (slant) image background removal, object segmentation and 3D object meshing.

3.1 Webcam control with Macromedia Flash and CGI saving/restoring

Macromedia Flash is one of the most popular and easiest way for Internet users to gain control of their webcam using a web-browser. We therefore developed a Flash photo capturing interface made for the Minoru 3D webcam, hard-coded with its highest capturing resolution (1280x480 pixels). This setup produces a side by side VGA resolution left and right image capture. Captured scene are displayed on the browser as live video, letting users to hold and navigate the camera to point to an object of interest. The acquired information is automatically transfered to and stored on our web-server by an HTTP post. From there, the Flash program terminates and pass the information to a Java Applet which handles the more complex calculation.

Saving processed information from a Java applet is very difficult because of Sun Microsystem's Applet security restrictions and was not described in our previous work [2]. Fortunately, an applet is allowed to send and save a stream of HTTP data to a public web server. We therefore created a CGI program at our server located at http://www.ivs.auckland.ac.nz/cgi-bin to serve such purpose. The calculated 3D coordinates are converted into a string and wrapped into a HTTP packet (containing the point cloud and the user's information) before being sent to a public web server. At the server side, the CGI application is set to wait for any incoming data, unwraps and stores the received information into a MySQL database system correspondingly to the photos taken and the user's information. The saved coordinates can only be retrieved back by its owner via our PHP gallery page.

3.2 Online stereo camera calibration and stereo matching

The Minoru 3D webcam is a commercial low-cost camera with expected decentring and radial distorsion effects. As a result, image calibration, distorsion removal and rectification is a crucially important intermediate processing step. An online stereo camera calibration is implemented as a separated process based on Zhang's calibration [9] to precisely estimate the intrinsic and extrinsic camera parameters.This process requires about a minute to capture 16 different shots of a 6x8 square calibration board. The captured images are sent to our server. Automatic corner detection is carried out to allocate all the corners with sub-pixel accuracy. Intrinsic and extrinsic parameters are estimated, rectified sample images shown on our web-browser and calibration and rectification paramteres saved on our online server.

In order to process and retrieve 3D coordinates of the captured object depicted within the stereo image pair, we use the "Colour-based symmetric dynamic programming stereo" described in [2]. The symmetric stereo algorithm is based on dynamic programming optimisation on a graph of Markov chains of a epipolar profile. Each point in the profile may be monocularly visible from left or right, or may be visible by both camera. The algorithm maximises the log-likelihood of the reconstructed profile compared to a random profile, taking into account partial occlusions from a single continuous surface profile. The implementation used also takes into account non-uniform relative photometric distortions between the left and right image, improving its usefulness for images captured in non-ideal conditions.

3.3 GPU base remote server processing

The use of local stereo algorithm has some limitations as the processing time and the ability to run some of the available algorithms will be dependent on the type of stereo algorithm that the user wants to implement, the size of the image acquired and the processing power available to the local computer. In some cases local processing will not be the best solution, and it can even be impossible to perform such processing on systems such as a portable smart-phone or home computer systems with very low computational capability.

For such situations we provide a web-page solution using a remote GPU CUDA server. In this case the user uploads the pair of images to the web-page and the system processes the image pairs on the server side. To improve the time and the abilities of the server to respond to multiple requests the stereo algorithms are implemented in parallel using CUDA on a GPU. This may be refered as CUDA (GPU) cloud computing.

After the server finishes with the stereo-process it updates the user file database and displays the results on the web-page side. The user then has access to the resulting disparity map, and can proceeed with the 3D model extraction or 3D java interactive display.

3.4 3D object modelling

The single-layer 3D modelling implemented previously in the Stereo Vision applet [2] simply built a continuous scene surface showing distances between every visible 3D point and the virtual visualisation optical centre. However, the 3D scene visualisation of a depth-map does not usually result in scene objects with clear surfaces and boundaries. Here we improve on our previous solution to form a more realistic visual 3D view using object segmentation to separate objects.

We apply the mean-shift segmentation algorithm [3] to separate objects from the background using the reconstructed depth-map image (grey colour image where the closer the 3D points to the camera center, the brighter the grey pixels are and vice versa). As previously stated, we first captured objects placed on a flat coloured background thus easily separating both by cleansing away the entire background plane whether flat or slant. To do so we devised a slant background removal process.

Slopes in 3D can be integrated into slope of z-axis against x-axis and y-axis, where the gradients of z against x and y represent the changes of z. An approximation of the gradient may be obtained quickly by sampling four points near the corners of the image and applying the formula below:

$$Gradient_{x,y} = (\frac{Z(x_2) - Z(x_1)}{x_2 - x_1}, \frac{Z(y_2) - Z(y_1)}{y_2 - y_1})$$

Subtracting this gradient from the grey-scaled depth-map yields enough slope reduction to transform the slant background to have an almost flat depth.

From there, the object segmentation is carried out. The main object is fully decorated with a clearly brighter grey colour when compared to the background. The results of this step are displayed in Fig. 1 on images labeled B, C and D.

Observation of 3D results are provided thanks to a Java 3D display on the browser environment. Users

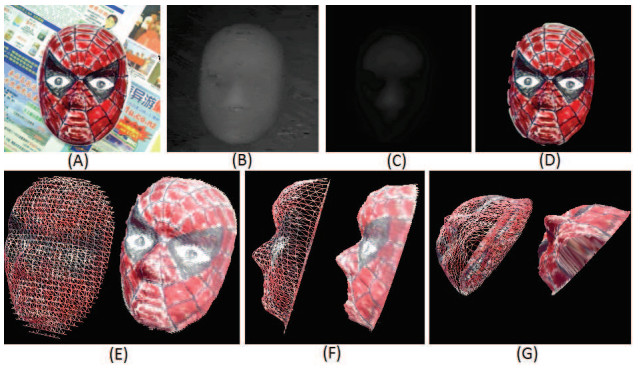

Figure 1. Steps and results of reconstructing a non-symetric object of a spider-man mask where A: reference photo taken from the Minoru 3D Webcam, B: depth-map reconstructed, C: Segmented mask, D: segmented texture, E-G: Gridded and full 3D model of the mask.

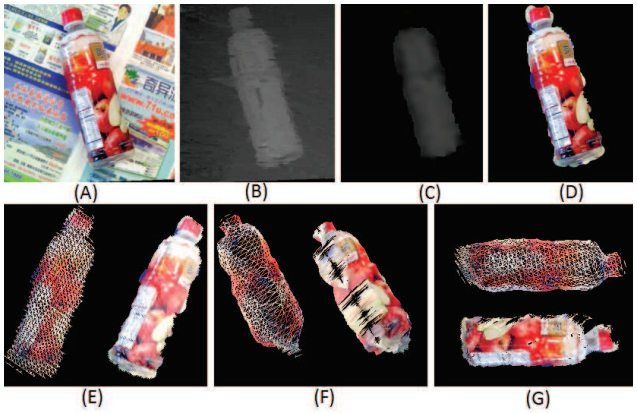

Figure 2. Steps and results of reconstructing a non-symmetric object of a drinking bottle where A: reference photo taken from the Minoru 3D Webcam, B: depth-map from Coloured SDPS, C: Segmented object of the bottle, D: segmented texture, E-G: Gridded and full 3D model of the complete bottle.

can choose to display the reconstructed object as symmetric or symmetric. A non-symmetric object has a flat back face while the symmetric object has identical back and front faces. Fig. 1 displays a modelling result of a non-symmetric spider-man mask while Fig. 2 displays a modelling result of a symmetric drinking bottle. Gridded models are available providing designers with early 3D sketches of the acquired object to derive full modelling strategies.

4 Low cost 3D face generation and animation

This proposed application makes use of a low cost 3D face generation and animation application described in [4]. It starts with our online capture of human face, skin segmentation and reconstructiong of a static depthmap in Fig. 4. The fully automatic depth map to 3D face mapping first applies a boosted classifier to locate the face within an image, then an Active Appearance Model (AAM) [5] to refine location and extract correspondence points for mapping. Next, Ra-

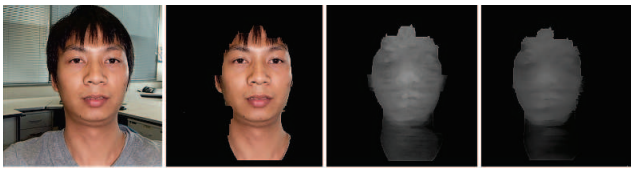

Figure 3. Steps and results of a human face reconstruction: skin segmentation, depth reconstruction and smooth depth displayed.

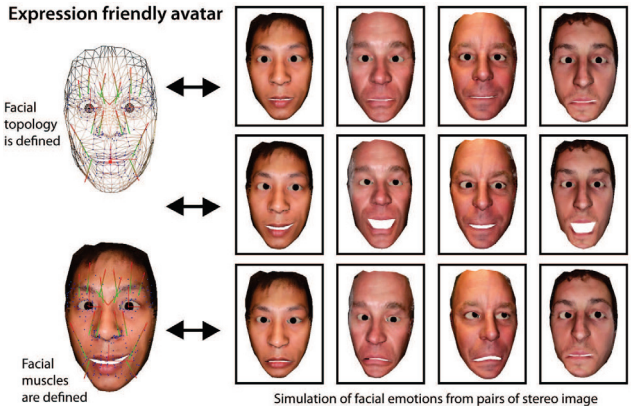

Figure 4. Automatic 3D face expression generated from the users face texture and measured depth. From top to bottom: Neutral, happiness and fear.

dial Basis Functions network [6] performs the mapping between data and model. This was applied to muscle and eye locations along with a final texture mapping through orthogonal projection. The entire procedure is summarised as follows:

- Acquire a set of face images and run a reconstruction algorithm to generate 3D face data.
- Run the trained boost classifier detector to locate the bounding box of the face region in the reference image.
- Based on this location and the size of the bounding box, initialise an AAM search.
- Once the AAM has located face features; extract a predefined set of correspondence points between the face depth data and the animatable face model.
- Apply a RBF mapping to the data so the animatable face model has a new shape that matches the particular test subject data.
- Generate a texture map based on the input face images and extract an eyeball texture using the AAM location and a Hough transform for circles.

Eventually muscle and eye locations are allocated along with a final texture mapping through orthogonal projection as described in [10]. If all these 3D points of facial object are mapped correctly, the animation of facial emotion can be artificially simulated on the the constructed 3D face.

5 Conclusion and further work

This described online tool prvides the complete chain of creation of 3D data from camera control to 3D model building. It provides an effective and portable support to 3D model designer interested in reproducing 3D model of a simple object instantly by using Computational Stereo Vision.To the best of our knowledge there is no such application available to the general public. Due to this tools portability and time efficiency, 3D models of many real life objects can be easily built without much setups and the whole process can be carried out within one minute.

Still, the quality of the object segmentation and the accuracy of the stereo matching algorithm must be adressed. So far, while the main object is separated from the background correctly its boundaries are not very precise which is likely due to the limited image quality of the Minoru 3D Webcam, the complex background textures and inherent stereo matching errors.We are currently carrying out segmentation from depth-map and referenced photos simultaneously. We are also working on the improvement of the 3D model generation by taking multiple shots of the same object from different view angles. An interesting avenue is the live generation of 3D models from stereo video stream which could reach several frames persecond given suitable high internet bandwidth, latest parallel GPU technology and VGA size stereo-cameras.

References

[1] Minoru is the worlds‘ first consumer 3D webcam. Retrieved August 2010, [On-line]. `http://www.minoru3d.com/`

[2] M. Nguyen, G. Gimel‘farb, P. Delmas. Web-based online computational stereo vision. In *Proc. IVCNZ08*, pp.1-6, 2008.

[3] D. Comaniciu and P. Meer, Mean shift: a robust approach toward feature space analysis, Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.24, no.5, pp. 603-619, 2002.

[4] A. Woodward. 3D human face reconstruction and expression modelling, PhD thesis, The University of Auckland, 2009.

[5] T.F. Cootes, G. J. Edwards, and C. J. Taylor, Active appearance models, In *Proc. ECCV98*, vol.2, pp.484-498, 1998.

[6] M. Buhmann, Radial Basis Functions: Theory and Implementations, Cambridge University Press, ISBN 978-0-521-63338-3.

[7] A. Woodward, D. An, Y. Lin, P. Delmas, G. Gimelfarb and J. Morris, An Evaluation of Three Popular Computer Vision Approaches for 3-D Face Synthesis. In *Proc. of SSPR/SPR 2006*, pp. 270-278, 2006.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. Int. Journal on Computer Vision, Vol. 60, pp. 91–110, 2004.

[9] Z. Zhang, A flexible new technique for camera calibration. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, pp. 1330-1334, 2000.

[10] A. Woodward, P. Delmas, Combining Computer Graphics and Image Processing for Low Cost Realistic 3D Face Generation and Animation. In *Proc. MVA05*, pp. 120-123, 2005.