Recent Advances in Online Stereo Web Application

Minh Nguyen, Patrice Delmas, and Georgy Gimel'farb
Dept. of Computer Science, The University of Auckland, Auckland 1142, New Zealand

***Abstract*—Our versatile web-based system allows users to dynamically generate visible surfaces of three-dimensional (3D) scenes from stereo pairs from monocular or stereo cameras including web-cams. The current advanced version accepts static images or live video sequences from different imaging sources (via direct or indirect Internet uploads), rectifies these images automatically, and processes the rectified images to reconstruct the scene by one of the available stereo matching algorithms. Results of processing are returned to the user in multiple formats, such as a disparity map, an anaglyphic image, an autostereogram, a virtual Java3D scene, a 3D .OBJ file, or a live depth video. The present system is portable, simple to set up and operate, and currently available online at `http://www.ivs.auckland.ac.nz/quick_stereo`. A variety of possible applications include remote camera control, on-line calibration and rectification, simple 3D object and avatar reconstruction, web-based real-time stereo matching, artistic creation of auto-stereograms, *etc*.**

I. Introduction

Current computer stereo vision research and applications are undergoing dramatic changes. More than four decades of intensive development has resulted in many efficient solutions, which are capable to handle large stereo images of complex 3D scenes under heavy occlusions, noise, and contrast variations. Recently the Internet-brought connectivity to both portable and desktop computers effectively moved the applied stereo vision from expensive high-end systems with a very limited range of the use outside professional photogrammetry and mapping to a large number of low-cost multimedia products such as digital photography, gaming platforms, and mobile technology.

This paper describes recent advances in our earlier web application [1] that allows for fast recovery of 3D data not only from professional stereo cameras, but also from low-cost off-the-shelf "point-and-shot" cameras, including web-cams. The functional and interface improvements of the current version comparing to [1] are as follows:

1) Input images from either web URLs or file uploads of many types: separate or combined left and right "crossed-eyes" or "parallel-eyes" images, anaglyphs, stereo .MPO files, auto-stereograms, and video streams.
2) More flexible image calibration and rectification: the built-in camera calibration and rectification for stereo camera/webcam with manual focusing and the automatic rectification for digital mono camera and stereo camera with automatic focusing.
3) Server-based processing: for stability, most of the complex works are now performed on a remote server.
4) Web-based online HTML5 and Java3D displays using the Bezier surface approximation.
5) More diverse output images: downloadable depth maps, anaglyphs, auto-stereograms, and 3D .OBJ files.

The present system can be used efficiently for real-time online stereo matching with side-by-side web-cam system, creation of anaglyphs and autostereograms, and modelling of 3D objects, e.g. faces. Fig. 1 displays a screen-shot of our web interface and control panel. This system communicates with a dedicated server for real-time processing and is accessible via the Internet to the public users from around the world. Section III below details the above-mentioned improvements.

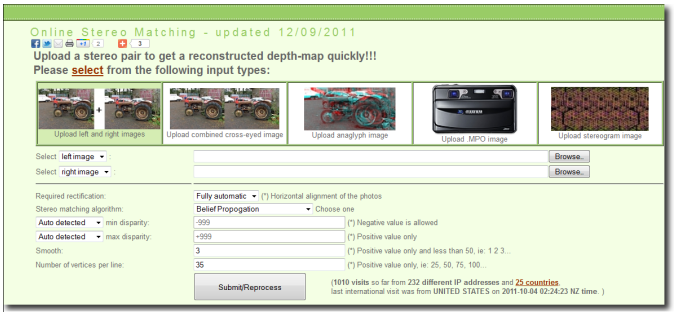

Fig. 1. Screenshot of the webpage with the input and control panel.

II. Background

A few known similar solutions, such as [2], [3], [4], [5] with too limited capabilities in processing unaligned stereo images (no image rectification for stereo matching and no web-based 3D display of results), demonstrate the current state-of-the-art in web applications related to computer stereo vision on the Internet. A few other web systems, like e.g. Photosynth [6], allow for processing multiple photos from various off-the-shelf cameras to reconstruct 3D positions and orientations of each camera at capturing moments and let users to browse these photos by navigating to different positions where the photos were acquired.

The Photosynth system is able to find a big number of characteristic corresponding points in the images and keep track of them in order to form a sparse 3D point cloud for an object or a scene and observe it then from different spatial positions. An alternative method for reconstructing a 3D point cloud from multiple photos of the same object in [7] allows the users to submit photos acquired by any hand-held camera, too. However, this system is running locally and has a too long processing time: up to 5 to 6 hours per 10–20 images – to be useful online.

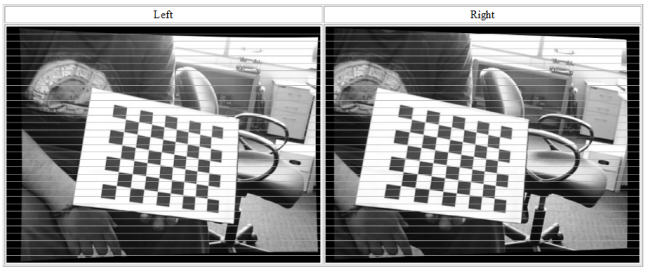

Fig. 2. A sample output from a single online calibration and rectification run on 16 input images.

III. Improvements in the current web application

A. Data inputs

Previously, the users had to upload for correct processing the same-size left and right images of a stereo pair separately. But stereo images are available on the Internet in a number of other formats where the left and right images are mounted together in a single file (a stereo, "cross-eye", "parallel-eye", or "wall-eye" pair, an "anaglyph", or a stereogram). Furthermore, the Internet provides public access to hundred of thousands such images around the World via public URLs. Therefore, our current system is able to handle these types of inputs either by file uploads or public URLs. This is also a promising way to attract the public attention to our site. The current system allows the user also to plug in a side-by-side web-cam system such as a Minoru 3D web-cam to send a video stream over the Internet and retrieve a real-time depth map.

B. Calibration / rectification for a side-by-side stereo camera

Images obtained from special stereo cameras such as Minoru 3D web-cam, uEye, or any two cameras rigidly mounted side by side to form a stereo set-up are rectified online. These systems typically have identical left and right image sensors with lenses of manually adjustable focal length. Assuming a chosen fixed focal length, internal and external parameters of each camera can be identified from the acquired images by a camera calibration process [8].

The calibration allows for accurate alignment of the images and effective removal of camera distortions. Typically, either images of some known calibration objects (e.g. a cube, chessboard patterns, *etc*.), or auto-calibration techniques, such as e.g. [9], [10] requiring no special calibration objects are used. However, to control the quality of the calibration, the calibration technique with the calibration objects, such as the Zhang's method [11] is employed in our case. The latter method has minimal equipment requirements (simply a printed checkerboard pattern on a paper sheet) while an alternative Tsai's method [12] requires non-coplanar calibration gauges, which are harder to set-up.

Images are placed to the server either by uploading files or from online web-cam controlled with Adobe Flash on a web browser. The server automatically detects corner points on the checkerboard pattern and identifies them at sub-pixel accuracy. The intrinsic and extrinsic camera parameters are then estimated from the identified points and saved on the server for the future use, so that the user need not recalibrate the cameras for every session. The calibration images are rectified and displayed to the user through the web browser for visual verification. Figure 2 exemplifies results of our online calibration and rectification. Including the time to acquire the images, the whole process takes less than two minutes of the user interaction, which we could be considered as brief and comfortable for public consumers.

TABLE I
Rectification quality vs. the number of input images.

Number	5	6	7	8	9	10
Error	0.41	0.17	0.10	0.37	0.29	0.26
Number	11	12	13	14	15	16
Error	0.39	0.43	0.45	0.42	0.42	0.41

For good accuracy, more than one image in different capturing orientations should be uploaded. A quick test to estimate the quantitative image rectification error for 5 to 16 inputs is illustrated in Table I. Less than 5 inputs in this test produced too erroneous alignments due to insufficient data for handling web-cams errors. The alignment errors were under 0.5 pixels for more than 4 inputs, which looks reasonable for a consumer web application. The sufficiently good calibration was achieved with 7 to 10 input pairs, whereas the rectification failed with less than 5, but needed no more than 13 inputs.

C. Rectification for a single-sensor camera

If two consecutive stereo images acquired by a conventional single-sensor camera (i.e. two images taken slightly to the left and slightly to the right) are uploaded, their rectification is carried out without the calibration. As was already described in [1], the fundamental matrix F is estimated with the well-known 8-point algorithm [13] using the RANSAC technique [14]. The KLT feature tracking algorithm by Shi and Tomasi [15] is used first to determine reliable feature points, such that the minimum eigenvalue of the matrix $Z(x, y)$ below is above a fixed threshold, $\min(\lambda_1, \lambda_2) > \theta$:

$$Z(x,y) = \begin{bmatrix} g_x^2(x,y) & g_x(x,y)g_y(x,y) \\ g_x(x,y)g_y(x,y) & g_y^2(x,y) \end{bmatrix}$$

where $g_x(x, y)$ and $g_y(x, y)$ denote the x- and y-components of the image gradient at pixel (x, y).

To run our application well on most of the users' computers, a larger than in [1] number of reliable feature points ($n = 500$) were chosen in the images as candidates for finding stereo correspondences. Instead of an earlier simple window-based intensity matching, the Lucas-Kanade pyramidal optical flow tracking algorithm [16] is employed to improve the search for correspondences. The mismatches are further eliminated by running this forward/backward tracking iteratively between the left and right images. Experiments showed that only a very few mismatches remain visible after about a dozen iterations, leaving the remaining good matches for the RANSAC evaluations. After epipolar lines are determined in both the

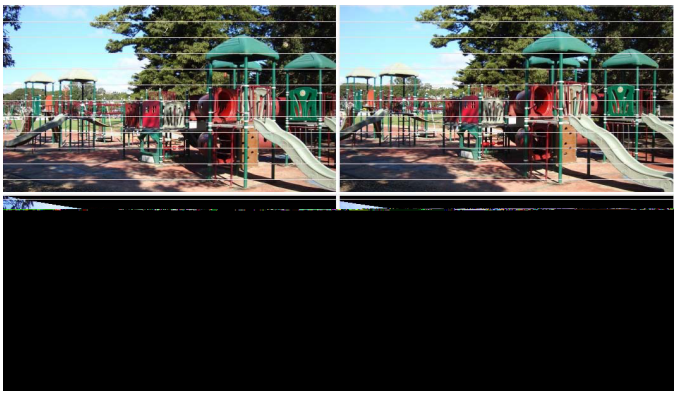

Fig. 3. Raw (top row) and rectified (bottom row) images of a children's playground.

images for the estimated matrix F, the rectified image pair is produced by aligning pixels located on the conjugate epipolar lines. Figure 3 shows the rectification results, parallel lines across the images visualising the alignment. This process can be also applied to images from auto-focussing stereo cameras (such as e.g. Fujifilm Finepix W1/W3). Since the focal length is changing at every shooting, the camera calibration is almost unable to find and reuse parameters for auto-rectification.

D. Stereo matching

Our system uses dense stereo matching algorithms that produce disparities for all the visible locations in a 3D scene. This depth information is commonly visualised as a disparity or depth map using either a greyscale or false-colour coding of the disparity or depth range (the disparities are readily converted to the depth and real-world 3D coordinates if the camera calibration and image rectification parameters are available). All the processing in our earlier system [1] was on-site via an Java Applet on a web browser, so that its ability to run most of the available algorithms was affected by the size of an acquired image and the available memory and power of a local computer. Moreover, it was difficult to retrieve results due to Java Applet Security restrictions. Such a local processing is not the best option in many practical cases and even might be impossible on home computers or portable smart-phones of very low computational capabilities.

Thus, all the complex operations in the present version are moved to a powerful web-server, dedicated to listen to clients around the world, process their data, and return results. Stereo matching algorithms from our previous client-side version have been reimplemented on the server (the SAD/SSD window-based matching [17] and semi-global algorithms such as Symmetric Dynamic Programming Stereo (SDPS) and its variants [18], [1], [19], [20]) in addition to a new 1D Belief Propagation (BP) algorithm [21] yielding a bit better matching results. The user can select, from an HTML combo-box, one of the algorithms and use it with any of the input types. This processing mechanism is more flexible, independent and responsive compared to the previous client-based one.

E. HTML5 and Java3D displays with Bezier surfaces

For better interactivity, our 3D displays are built as separate components mounted to the web page and allow the user to view easily 3D representations of results on a browser without a dedicated 3D viewer. The earlier system employed only Java3D mounted to the main applet and the fixed (but obviously incorrect in most of the cases) disparity mapping to the grayscale range $[0, 255]$ was used to display 3D points. The current advanced version uses WebGL on HTML5 in addition to the Java3D display. Compared to Java3D, WebGL standard is newer and supported recently by many browsers including Firefox, Google Chrome, and some latest versions of Internet Explorer. Now any reconstructed 3D surface can be viewed directly on the browser without installing a special plug-in.

The output of 3D scene reconstruction is also modified by forming surfaces from the obtained 3D depth data. Given the common focal length, f, and the baseline, B, from the camera calibration, the coordinates (X, Y, Z) of each visible 3D point are restored, assuming the standard epipolar geometry of a stereo pair, as

$$Z = f\frac{B}{x_l - x_r} \equiv f\frac{B}{d};\ X = x_l\frac{B}{d}, Y = y_l\frac{B}{d}$$

where (x_l, y) and (x_r, y) are 2D image coordinates of corresponding points in the left and right image of an epipolar stereo pair and $d = x_l - x_r$ is the x-disparity.

The reconstructed depth map is generally noisy and containing incorrect depth values creating unexpected spikes. To suppress these reconstruction errors, the map is represented with a Bezier surface [22], each 3D surface point being now calculated by blending two orthogonal Bezier curves. Fig. 4 presents an example of our Java3D display of the reconstructed depth map, the display showing a smoother surface and natural depth proportons comparing to the real world object.

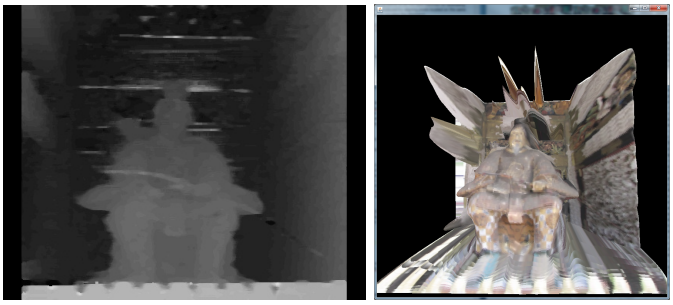

Fig. 4. Depth map (left) of a statue and its Java3D display (right).

IV. Four possible applications

Advantages of our present system for the general-public users is illustrated below with the following four applications: (*i*) online stereo matching for video streams from a side-by-side stereo web-cam; (*ii*) 3D model retouching using 3D .OBJ files generated from the user's stereo images and editable by a number of professional graphical tools; (*iii*) a low cost 3D face generation from a pair of facial images, and (*iv*) creation

of auto-stereograms with advanced functionalities over other auto-stereogram makers available online.

A. Fast online stereo matching for a stereo web-cam

The implemented system can display in real-time depth maps generated for video streams obtained from a user's side-by-side USB web-cam, after calibration and rectification parameters are restored to rectify the input frames on the server. Typically, our server is set ready for stereo matching and immediate return of reconstructed depth maps to the user. At the current stage, our system can output the depth maps at the rate of 2 frames per second under the input impage size of 160×120 pixels.

B. 3D object modelling

Generating 3D surfaces from photos for further professional editing is of obvious public interest, inspiring us to provide the users with more useful outputs than just a bland depth map. After determining the depth at each pixel (x, y) from the reconstructed disparity map, a complete set of 3D points is generated further and stored in a popular .OBJ format. Wave-front 3D .OBJ files are an *ad-hoc* standard adopted by many vendors of 3D graphics applications [23] such as Cheetah3D, Autodesk's Maya, Blender, MeshLab, *etc*. Figure 5 shows a screen-shot of viewing and editing a spider-man mask and a real human face with Blender after clearing noisy regions.

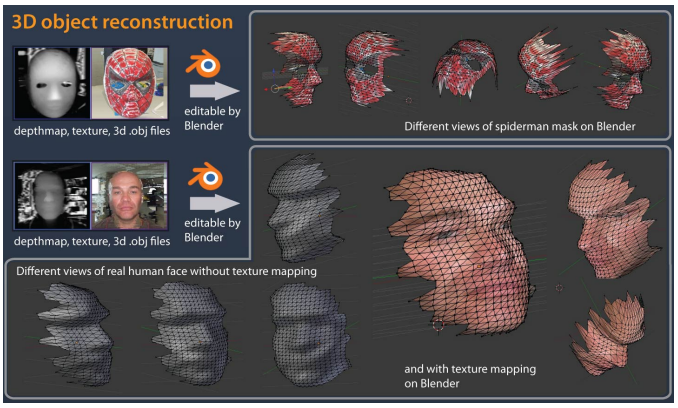

Fig. 5. 3D object viewing and editing with Blender.

A potential generation and fast display scenario for a simple 3D model with full 360-degree surface without much editing and experience can be implemented. At this stage, a single object has to be placed on top of a flat colour background with rich texture (such as e.g. a newspaper). The input photos, acquired by a camera placed right above the object, are then processed with our system using an automatic disparity range estimation to obtain a depth map and a texture map. The object is separated from the background with the mean-shift segmentation [24], iteratively calculating the mean location of a segment cluster in a chosen search window. Finally, the object-of-interest has clearly brighter intensities comparing to the background, so that it is segmented by removing the

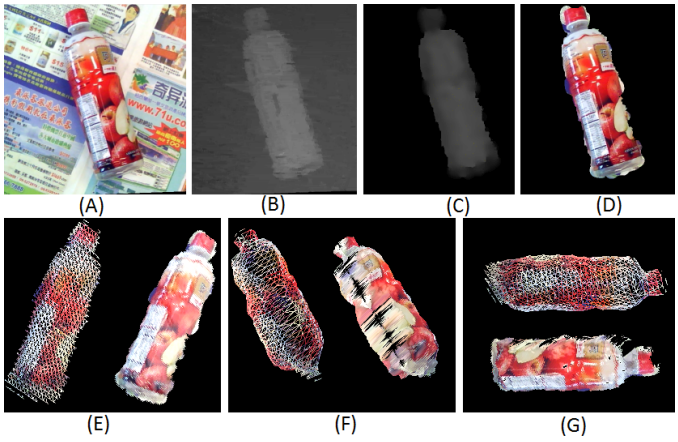

Fig. 6. Reconstructing a symmetric water bottle. Top: a reference photo acquired by the Minoru 3D web-cam (A), the reconstructed depth map (B), the segmented object (C), the segmented texture (D). Bottom: the meshed and the full 3D model of the symmetric bottle (E–G).

background colours surrounding the brighter object segment (see Fig. 6, B, C and D).

Two types of simple object reconstruction can be handled at this stage: an asymmetric object with a flat bottom (e.g. a box) and an object with symmetric surface (e.g. a water bottle). For the latter objects, the 360-degree meshed models are calculated by reflecting all the depth coordinates.

C. Low cost 3D face generation and expressive animation

A low cost 3D face generation and animation system in [25] can be combined with our present online system to generate 3D face data for customised avatars. The latter system captures human faces, segments skin area, reconstructs static depth maps, and generates 3D surface coordinates. The obtained data is then mapped in the former system onto a generic animatable face mesh model with a full muscle set acting upon the mesh. Fig. 7 displays seven 3D faces generated. Because the muscles are implemented on the generalised face, our system can interactively control the avatar to mimic emotional expressions. This application could be interesting for the use in computer games or instant chats where emotions could be caught and sent over the network.

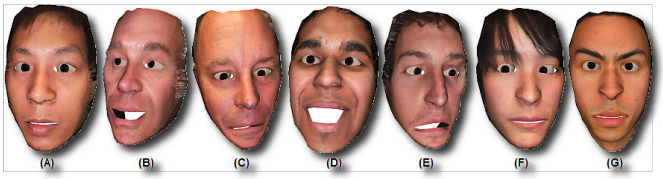

Fig. 7. Simulated 3D avatars from images taken by Fujifilm Finepix W1.

D. Creating autostereograms

Once a depth map has been created, the user has an option to choose from a number of texture patterns to generate an auto-stereogram, which represents all the points of the depth

map. Each pixel in a depth map corresponds to two points in the auto-stereogram, which act as virtual conjugate projections of the same 3D point and thus must have the same colour. An example in Fig. 8 is built for a pair of unaligned photos of a wooden roof. Benefits of our auto-stereogram creation process

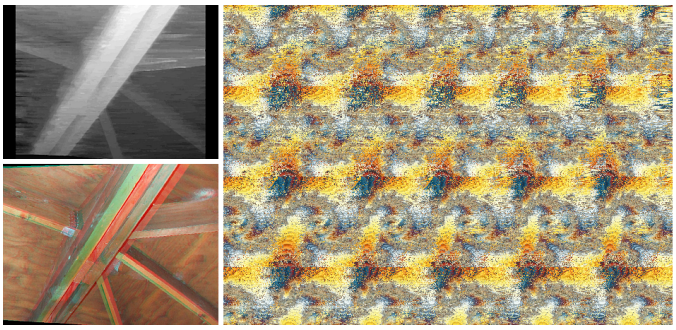

Fig. 8. Our auto-stereogram (right), generated from a pair of unaligned photos of a wooden roof, with the generated depth map (top-left) and anaglyph (bottom-left).

over the alternative online counterparts:

1) A 3D model to be hidden in the auto-stereogram is reconstructed from a stereo par provided by the user, rather than is chosen from a selection of given models.
2) An user's auto-stereogram can be uploaded as an input to generate the 3D data.
3) Input auto-stereograms can be converted into anaglyphs, viewable through red-cyan glasses as an additional viewing method.

V. Conclusions and Future Work

The current advanced web-based stereo vision system provides effective online interactions virtually to any user who is interested in stereo vision.At present, such systems available to the general public are uncommon, and the few known counterparts have considerably more limited processing tools. Our system is widely open and flexible for the use by both the professionals and amateurs. Since our website has been opened for public access for a two months, it received more than 1000 visits from 232 different IP-addresses from 25 countries around the world. The system is capable to process not only uploaded files but also public URLs, which can be found easily on the Internet. This flexibility helps us to increase the usability of the system and make it useful for teaching, demonstration, and knowledge sharing in the stereo vision research community as well as for the practical use.

In the near future, we are going to use it for controlling emotional expressions of a 3D facial avatar using a live stereo web-cam (Minoru) together with the Principal Component Analysis (PCA) [26]. Woodward et al. [27] identified a method of simulating facial muscle movements to show seven basic types of human emotions: neutral, happiness, sadness, surprise, fear, disgust, and anger – on a 3D avatar (see Fig. 7). The PCA can be applied to register the user's seven expressions via the web, the live Minoru web-cam being attached at the user's side and captureing every movement of the user's face. Whenever a real expression is detected as one of the seven emotions, the client will notify the system by a simple text data, which can be transmitted over the network. At the reception side, the system slowly changes the avatar's expression to match the user's one. Such systems can be used in the gaming industry as a simple way to control facial expressions of cartoon characters or in live interactions between the Internet users. This approach will potentially reduce requirements to the network bandwidth comparing to traditional transmission of live web-cam videos.

Implementation-wise, the challenge is how to express feelings of each person on the face when people reveal these feelings in a completely different manner. To tackle this problem, image recognition and machine learning techniques could be applied to train a system to recognise particular emotions. At the very initial stage, the OpenCV tools could be employed to crop the facial area of the live camera frame, resize it to a fixed area (say, 100×100 pixels), and develop a database of such training data by asking the users to provide the original images, tagged with corresponding emotions. The PCA can be used then to determine one emotion from a set of seven standard basic emotions. For a relatively small database, the detection should be reasonably accurate and fast enough for web applications; however, the actually quantitative tests will be carried out for a firm conclusion.

The foreseen additional challenge is that the PCA highly depends on different lighting conditions and the amount of training images. If lighting conditions differ between the training and test images, the desired classification of emotions will be most probably incorrect. To circumvent this problem, the PCA might be applied to both depth maps and normalised textures. These research directions will be explored using the described online web-based stere vision system.

References

[1] M. Nguyen, G. Gimel'farb, and P. Delmas, "Web-based on-line computational stereo vision," in *International Conference Image and Vision Computing New Zealand*, nov. 2008.

[2] C. Sun, "Fast stereo matching demo." May 2011. [Online]. Available: http://extra.cmis.csiro.au/IA/changs/stereo

[3] D. Scharstein, "Middlebury Stereo Vision Page," 2007. [Online]. Available: http://vision.middlebury.edu/stereo/. [Accessed: Sep. 15, 2010].

[4] "Stereophotoviewer applet." May 2011. [Online]. Available: http://www.stereomaker.net/java/spva/stereowe.htm

[5] "Stereophoto maker." May 2011. [Online]. Available: http://www.stereomaker.net/eng/stphmkr/

[6] N. Snavely, S. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 835–846.

[7] M. H. Nguyen, B. Wnsche, P. Delmas, and C. Lutteroth, "Realistic 3d scene reconstruction from unsconstrained and uncalibrated images taken with a handheld camera," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2011.

[8] F. Remondino and C. Fraser, "Digital camera calibration methods: considerations and comparisons," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 5, pp. 266–272, 2006.

[9] M. Pollefeys, R. Koch, and L. Gool, "Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters," *International Journal of Computer Vision*, vol. 32, no. 1, pp. 7–25, 1999.

[10] S. Maybank and O. Faugeras, "A theory of self-calibration of a moving camera," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 123–151, 1992.

[11] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330 – 1334, nov 2000.

[12] R. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1986.

[13] R. Hartley, "In defence of the 8-point algorithm," in *Computer Vision, 1995. Proceedings., Fifth International Conference on*. IEEE, 1995, pp. 1064–1070.

[14] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[15] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1993, pp. 593–600.

[16] J. Bouguet *et al.*, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," *Intel Corporation, Microprocessor Research Labs, OpenCV Documents*, vol. 3, 1999.

[17] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: theory and experiment," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 9, pp. 920 –932, sep 1994.

[18] G. Gimel'farb, "Stereo terrain reconstruction by dynamic programming," *Handbook of Computer Vision and Applications. Signal Processing and Pattern Recognition.*, vol. 2, pp. 505–530, 1999.

[19] M. Nguyen, G. Gimel'farb, and P. Delmas, "Stereo vision: A Java-based online platform," in *MVA2009 IAPR Conference on Machine Vision Applications*, May 2009.

[20] M. Nguyen, "Web-Based On-Line Computational Stereo Vision," Master's thesis, The University of Auckland, New Zealand, Computer Science department, 2008.

[21] R. Gong, "Belief Propagation Based Stereo Matching with Due Account of Visibility Conditions," Master's thesis, The University of Auckland, New Zealand, Computer Science department, 20011.

[22] P. Bourke, "Bézier curves," *Bézier curves*, 1996.

[23] K. McHenry and P. Bajcsy, "An overview of 3d data content, file formats and viewers," Technical Report ISDA08-002, Tech. Rep., 2008.

[24] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, pp. 603–619, 2002.

[25] M. Nguyen, G. Gimel'farb, P. Delmas, Y. H. Chan, A. G. Strozzi, and A. Woodward, "Web-based rapid prototyping of objects using gpu-based computing: Application to 3D face modelling," in *MVA2011 IAPR Conference on Machine Vision Applications.*, June 2011.

[26] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37 – 52, 1987.

[27] A. Woodward, "3D human face reconstruction and expression modelling," Ph.D. dissertation, The University of Auckland, New Zealand, Computer Science department, 2009.