MVA2009 IAPR Conference on Machine Vision Applications, May 20-22, 2009, Yokohama, JAPAN

Stereo Vision: a Java-based online platform

M. Nguyen, G. Gimel'farb, P. Delmas
Department of Computer Science, The University of Auckland, New Zealand.
p.delmas@auckland.ac.nz

Abstract

An on-line Java applet integrating interactive image controls with a view towards investigating stereo matching and 3D scene reconstruction is proposed. It is freely open to a wide audience and can serve as a teaching or research tools. The interface offers image uploads as well as several options to perform image pair rectification and stereo matching. Higher end features such as sub-pixel depth map resolution as well as a derived version of the well performing SDPS algorithm are proposed. Interactive 3D visualisation of the image pair scene (combining depth and textural information) is available via Java3D. The current version of this applet is stored at http://www.cs.auckland.ac.nz/∼mngu012.

1 Introduction

From the first research forays almost three decades ago [8] to today's latest available applications, binocular stereo-vision is one of the most actively research field in computer vision. As for other communities, the current trend is to online shared programming resources and online shared databases and results comparison with the benchmark being the Middlesbury website [5]. AS the most advanced online dedicated StereoVision webpage, it is dedicated to specialists only. It allows the users to submit new stereo algorithms or/and resulting depth maps for a database set of given stereo pairs and ground truth available freely. As a consequence this generates an easy and ready available comparison tool for the best-performing algorithms (with 48 algorithm referenced today) on the limited set available. Code for some of the algorithms presented can be downloaded. Only a few similar approaches [1, 2, 3] can be found elsewhere on the WWWeb. One of the most advanced interactive tool [1] presents a straightforward stereo matching algorithm based on rectangular sub-regioning and allows the user to submit existing URLs of stereo images, to input some parameters, and to output a gray-scale depth map. Both Stereo Photo Viewer [2] and Stereo Photo Maker [3] allow the users to input stereo images as well as to choose between outputs format such as cross- or parallel-eyed pairs, red-cyan or red-blue anaglyphs, mirrored images, etc. Simple rectification algorithms are implemented, but no stereo matching algorithms are provided. While outside of our study it is worth noticing the new and stylish Photosynth software [4] presents an attempt at multi-view matching. It supposedly allows users to upload their photos of a given landmark construction online and given a sufficient number of images may reconstruct an interactive 3D view of the scene. It is unclear if the software requires prior knowledge of the 3D structure of the object to be reconstructed. The novelty of Photosynth resides in its slick scalable 3D viewing interface. While Photosynth may be seen as the future of multiple views 3D scene reconstruction, it is not able to convey real-time processes for online applications. Web-based tools offering interactive approaches to compute 3D views (through depth maps) from stereo pair images are almost unavailable for both general public and specialists. None could be found offering both rectification and several stereo-matching processes in an interactive interface. Our intend is to make this research field more accessible to any novice users who wish to quickly built 3D views from any pair of images captured by an off-shelf hand-held digital camera and more or less respecting. Moreover, with the sharing system implemented where specialist can plug-in their own stereo matching programs (by uploading java codes following our set of image processing classes) we aim to provide a platform where specialists can showcase their research while protecting IP rights. To cater for this, we have developed a more general stereo vision tool with multiple functional capabilities. Our work offers automatic or manual rectification procedures as well as the choice of four different stereo algorithms up to sub-pixel disparity accuracy and an interactive 3D viewing experience based on the java 3D engine. The interface functionalities will be briefly described in the next section (for more details see [12]). The following sections further details our best performing algorithm and its performance as well as introduce further results using our interface on stereo pairs acquired in challenging environment.

2 The online Java applet

The developed webpage applet (www.cs.auckland.ac.nz/∼mngu012) is currently running on our server and is accessible to all the Internet users. It allows registered users to input and process any pair of digital images. Users not wishing to register will still be able to process any of the provided sample image pairs. Upon registration, a user obtains a folder to upload additional image pairs. The user's images are password protected to forbid third party access. The registered users have the choice to make their own stereo-pairs private or public thus allowing others to try their data. The same works for uploaded stereo matching algorithms.

The whole process of 3D stereo reconstruction from non-rectified pair of images to a 3D visual experience is illustrated in Fig. 1 using our interface processing steps. First a user may upload a pair of images to our Stereo Vision website. Our only requirement is that the input images are expected to be reasonably close to the epipolar geometry constraints [8]. To do so, the user is asked to acquire left and right images

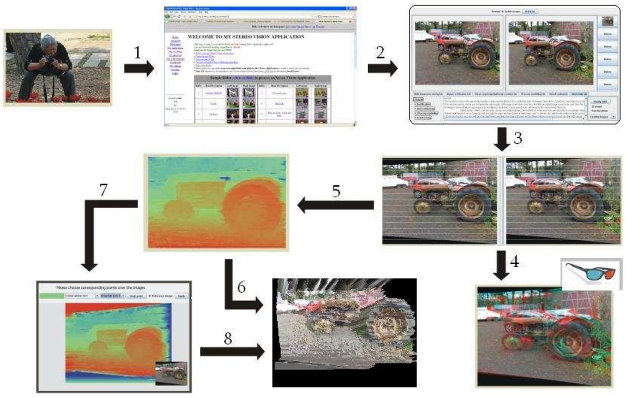

Figure 1: Main stereo processing steps.

with roughly the same camera orientation (e.g. almost parallel optical axis) and slightly translated positions. Next the pair of images are rectified to form an epipolar stereo pair (left and right image pixels corresponding to the same feature are situated on the same horizontal lines in both image) thus allowing most current stereo-matching algorithms to be used to generate a depth map. Several modes are available for the rectification: Automatic procedure involves the RANSAC-based [7] 8-point algorithm [6] following best matching feature points detection using The KLT feature tracking algorithm by Shi and Tomasi [9]. The first n strongest reliable point pairs (situated in well separated regions spreading the whole image) in a given stereo pair are selected as the candidate correspondences (e.g. see Fig. 2). Left-to-right and right-to-left feature point matches are carried out. Only the strongest mutual matches are kept. Gaussian filtering is first applied to both images in order to reduce image noise that might lead to high eigen-values and cause false selection. A semi-automatic procedure involves the manual removal of feature points which leads to poor rectification outcome. A full manual process requires the user to click both images for matching image features towards rectification. Further details on these processes can be found in [12].

Next, one of the available stereo matching algorithms may be applied on the rectified stereo pair in order to reconstruct a dense grey- or colour-coded depth map. Four algorithms are provided namely SAD, Fast SAD, SDPS [11] and colour SDPS, our latest improvement of the well performing SDPS algorithm further detailed in the next section. We also offer the possibility to compute sub-pixel disparities to generate smoother depth-maps which improves the 3D viewing experience. A graphical 3D model of the scene is generated by combining the computed depth map and registered image texture. The user can observe different views of the 3D scene using Java 3D functionalities.

Our main applet integrates several interactive functionalities such as image zooming, history tracking results, a didactic description of the processes involved and their controls.

3 Stereo matching algorithms

Stereo matching is a very active research area and recent efficient energy minimisation techniques such as graph-cut or belief propagation have achieved high re-

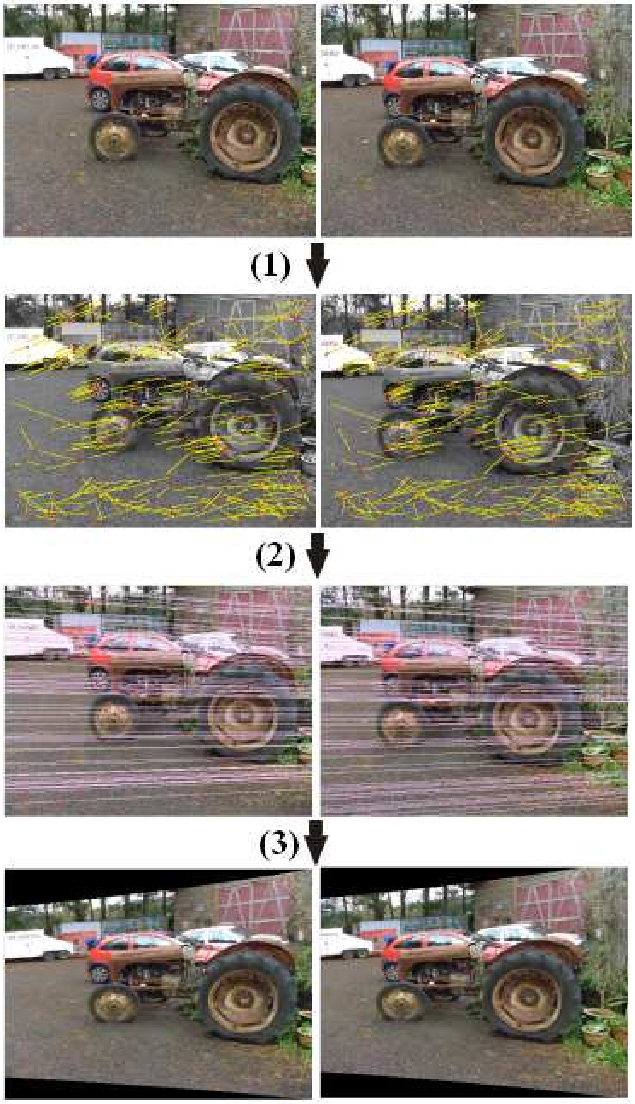

Figure 2: Rectification of an uploaded pair of images (the tractor): (1) finding stereo correspondences, (2) estimating epipolar lines, and (3) rectifying the images into an epipolar stereo pair.

construction accuracy on available test sets [10]. Unfortunately high accuracy algorithms carry very high time complexity which is incompatible with web-based applications. Instead, two simpler and faster algorithms, namely, the moving window SAD (Sum of Absolute Differences) and the Symmetric Dynamic Programming Stereo (SDPS) [11] have been considered.

The original SDPS algorithm uses only greyscale images and a fixed occlusion cost. For our application an enhanced version of SDPS with colour signals without adaptation to local contrast deviations (to cut down computational complexity) was developed. Here, occlusion costs depend on local image homogeneity along the considered epipolar line. Sub-pixel accuracy was further implemented for more accurate 3D reconstruction.

Let (i, j) be integer x-coordinates of corresponding pixels along the same epipolar scan line in the left (l) and right (r) stereo images, respectively, and let $R_{l:i}$, $G_{l:i}$, and $B_{l:i}$ denote the red, blue and green component of the signal for the pixel i in the left image. The sig-

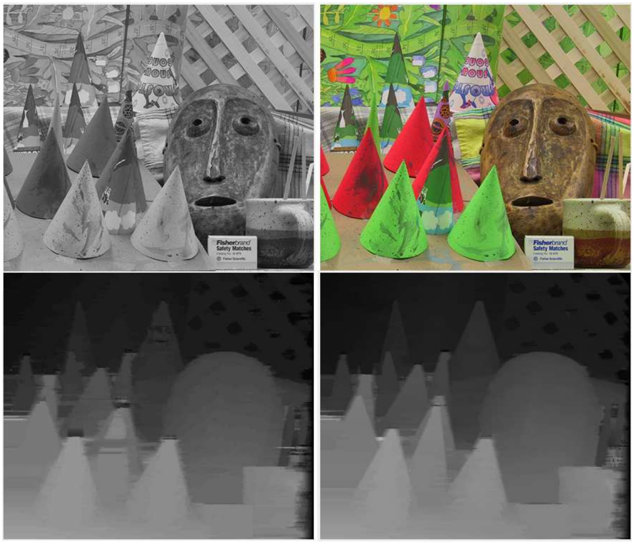

Figure 3: 3D reconstruction for the stereo pair "Cones" [5] using the original SDPS (left column) and the enhanced SDPS (right column).

nal dissimilarity cost for a binocularly visible 3D point (BVP) is calculated as:

$$D_{\mathrm{BVP}:ij} = (0.3D_{R:ij} + 0.59D_{G:ij} + 0.11D_{B:ij})$$

where

$$\begin{aligned} D_{R:ij} &= |R_{\mathrm{l}:i} - R_{\mathrm{r}:j}| \\ D_{G:ij} &= |G_{\mathrm{l}:i} - G_{\mathrm{r}:j}| \\ D_{B:ij} &= |B_{\mathrm{l}:i} - B_{\mathrm{r}:j}| \end{aligned}$$

Figure 3 compares depth maps reconstructed by the original SDPS algorithm and by the new version using colour signals. Note the better accuracy of the latter for overlapping red and green cones in the middle of the scene. Tests ran on the Teddy and Cones (as provided in the Middlebury dataset) showed an increase by 15

The occlusion cost for a monocularly visible point (MVP) is calculated assuming that two similar pixels from the left and right images (i.e. having the low $D_{\mathrm{BVP}:ij}$) will likely map to the BVP than the MVP. Here, the occlusion costs are complementary to the dissimilarity ones:

$$D_{\mathrm{MVP}:ij} = \max\{\beta - D_{\mathrm{BVP}:ij}, \alpha\}$$

where α is a fixed noise cost and $\alpha < \beta$. The experimentally chosen values were $\beta = 150$ and $\alpha = 0$.

For better smoothness, the depth map can be reconstructed with sub-pixel accuracy (only horizontally for sake of simplicity). Sub-pixel colour signals (n being the sub-sampling rate) are linearly interpolated from the original pixel signals. The interpolating larger images are processed normally albeit with a n times larger disparity range. When returned to its original size, the resulting depth map image has floating values depth increments (e.g. see Fig. 4).

3D visualisation of a given scene as depicted in processed stereo images are formed by applying Java3D tools to a reconstructed depth map and registered texture information.

4 Web-based performance

Web applications are running on an internet browser with embedded Java applets. Performance might differ from a stand-alone application running on a local computer. Tests we performed on a top end computer with high broadband capabilities to assess the differences between online (Internet Explorer 6.0.29 running on our web server http://www.cs.auckland.ac.nz/∼mngu012) and stand-alone versions of our stereo vision program.

Seven image pairs (including Middlebury and our own sets) of different size (up to $600,000$ pixels) were tested in the rectification and matching process including the image display, automatic feature detection (using KLT), fundamental matrix computation (using the RANSAC algorithm), rectification, display of the rectified stereo pair and in a separate computation, stereo-matching and corresponding depth map display. We found no computational difference between the stand-alone and online application. The overall rectification process took between 2 and 10 seconds and was not well correlated to the image size but rather to the texture complexity within the image. By comparison the stereo-matching processes were well correlated with the image size (e.g. see Table 1)

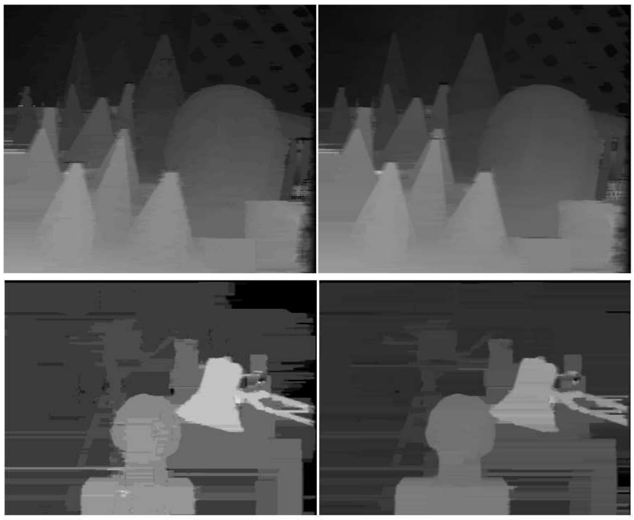

Figure 4: A comparative demonstration of the smoothness obtained by the sub-pixel effect on Colour SDPS for some Middlebury datasets[5].

	nP	R	FSAD	SDPS	CSDPS
Corridor	0.66	21	328	375	422
Tsukuba	1.1	16	297	563	589
Generic	7.7	38	344	625	688
Saw	1.65	34	657	1484	1703
Teddy	1.69	48	781	1890	2141
Cones	1.69	57	1078	2093	2312

Table 1: Stereo algorithms performance on several middlesbury image pairs. From left to right: Number of pixels (nP) $\times 10^5$; Disparity range (R); Processing time in miliseconds for fast SAD, SDPS and Colour SDPS.

5 Conclusion and further work

The described collaborative stereo vision application provides effective on-line support to novice and experience users who are interested in computational stereo vision and wants to convert their own image pairs into

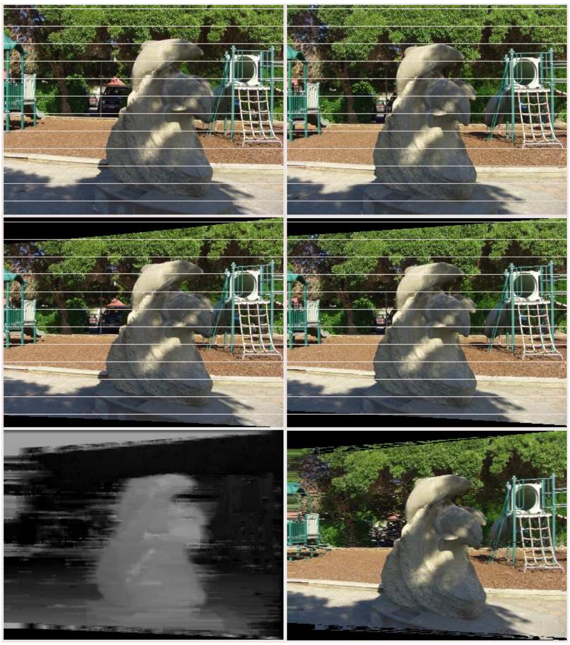

Figure 5: 3D reconstruction for the acquired dolphin stereo-pairs using the colour SDPS. Top: initial stereo pair; Center: rectified pair; Bottom: left: colour SDPS depth map, right: 3D visualisation

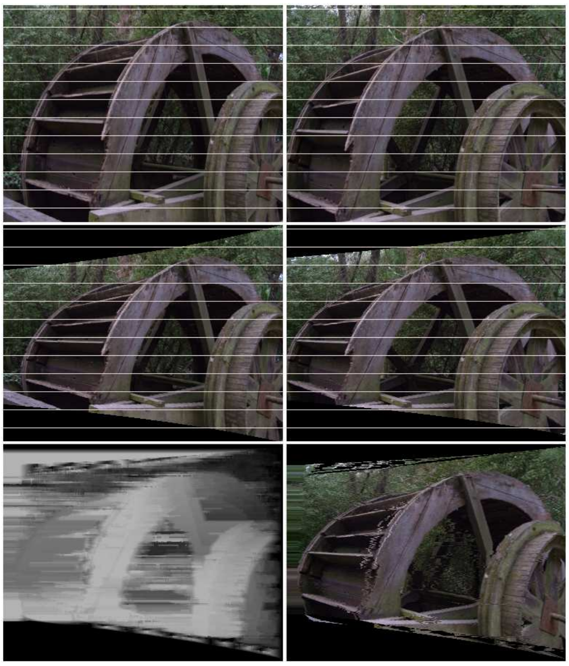

Figure 6: 3D reconstruction for the acquired wheel stereo-pairs using the colour SDPS. Top: initial stereo pair; Center: rectified pair; Bottom: left: colour SDPS depth map, right: 3D visualisation

3D models. Results displayed both show potential and limitations of such approaches as compared with multi-view matching. Currently only a limited set of such applications are available to the general public, and a few known ones have only a very limited number of the processing tools available with our interface. As stereo matching is under constant development, our application is intended to be a peer-sharing platform where new stereo algorithms can be plug-in to investigate and evaluate their performance with a wider number of researchers and at a faster pace than through publications. In such an application, there are many entries where other rectification and stereo matching algorithms and implementations can be plug-in to yield the better overall performance. We are currently implementing a new rectification process based on SIFT features [13] followed by the 7-point sampling and pre-emptive RANSAC for the fundamental matrix estimation.

References

[1] Changming Sun: Fast Stereo Matching Demo. Retrieved October 2008, [On-line]. `http://extra.cmis.csiro.au/IA/changs/stereo`

[2] StereoPhotoViewer Applet. Retrieved October 2008, [On-line]. `www.stereomaker.net/java/spva/stereowe.htm`

[3] StereoPhoto Maker. Retrieved October 2008, [On-line]. `http://www.stereomaker.net/eng/stphmkr/`

[4] Photosynth: Microsoft Live Labs. Retrieved December 2008, [On-line].`http://livelabs.com/photosynth/`

[5] D. Scharstein and R. Szeliski, Middlebury Stereo Vision Page. Retrieved October 2008, [On-line]. `http://vision.middlebury.edu/stereo/`

[6] R. Hartley, In defence of the 8-point algorithm. In *Proc. 5th Int. Conf. on Computer Vision, Cambridge, Massachusetts, USA*, pp. 1064–1070, 1995.

[7] M. A. Fischler and R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. ACM*, Vol.24, pp. 381-395, 1981.

[8] Q. T. Luong and 0. Faugeras, The Fundamental Matrix: Theory, Algorithms, and Stability Analysis, In *Int'l J. Computer Vision*, vol.17, no.1, pp. 43-76, 1996.

[9] J. Shi and C. Tomasi, Good features to track. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR94), Seattle, WA, USA*, pp. 593–600, 1994.

[10] D. Scharstein and R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, Int. J. of Computer Vision, vol. 47, no. 1–3, pp. 7–42, 2002.

[11] G. Gimel'farb, Probabilistic regularisation and symmetry in binocular dynamic programming stereo, Pattern Recognition Letters, Vol. 23, pp. 431–442, 2002.

[12] M. Nguyen, G. Gimel'farb, and P. Delmas. Probabilistic regularisation and symmetry in binocular dynamic programming stereo, Proceedings of the Image and Vision Computing New Zealand Conference (IVCNZ'08), Lincoln University, New Zealand, pp. 104-109, 2008.

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