Region-Based Motion Analysis and 3D Reconstruction for a Translational Video Sequence

Xiaodong Huang and Eric Dubois
School of Information Technology and Engineering (SITE)
University of Ottawa
Ottawa, ON, K1N 6N5 Canada

Abstract

This paper presents a hybrid 1D motion estimation algorithm which combines pixel-based and region-based approaches that can give depth images from translational video sequences with very high quality. Firstly, we combine the motion information estimated by a variational regularization approach and by the Gabor transform through histogram analysis to identify those regions with zero motion (like for the sky). Then another round of region matching is carried out to refine the motion values for the other regions. Our algorithm can detect most of the sky regions segmented by foreground objects with complex geometry while keeping the boundaries of moving objects sharp and clear, which is an very important feature to obtain accurate 3D models. The high quality motion maps/depth images obtained by our algorithm are shown along with 3D reconstructions from novel viewpoints.

1. Introduction

Obtaining 3D models from video sequences (without active range-scanners) is a challenging research area that has received a great deal of study. The estimated 3D models with good quality could be used for many applications including photo-realistic immersive environments, image-based rendering, virtual reality, etc.

Most existing approaches for this research topic developed in recent years, e.g. [5, 10, 1], can be approximately divided into two steps: (1) estimation of depth images or separate 3D models at different positions; (2) combining these separate depth images or 3D models to construct *one* 3D model for the scenes contained in video sequences. For the first step, usually the algorithm of structure from motion is used for monocular video sequences to convert the motion information to depth images for different locations [5]. If stereo video sequences are involved in the first step, then disparity estimation algorithms can be used to obtain scene depth at each location [10]. For the second step, the camera poses and transform matrices among different locations need to be estimated. These are usually carried out by feature matching, and then these transform matrices can be further adjusted with respect to one reference location using some optimization techniques like bundle adjustment. With these optimized camera poses, the separate 3D models at different locations can be transformed to the reference location in order to form one 3D model represented by triangular meshes [5, 10] or 3D point clouds [1]. The two steps do not have to be in this order, the transform matrices for camera movement can be estimated by feature matching first, and then the dense disparity can be further estimated exploiting the epipolar information associated with those transform matrices [1].

For the two steps, the first one for depth image estimation is the most difficult and important one. Because both algorithms – structure from motion and disparity estimation – are matching processes, and matching is an ill-posed inverse problem, the ambiguities contained in these matching processes can bring many errors in the estimated depth images, such as noisy outliers and wrong depth values for an entire large region. Such errors can greatly affect the second step as well as the quality of the final 3D models. Some typical difficulties in depth estimation are the matching for the pixels in slanted surfaces, the matching for sky areas (especially when the sky is segmented by some trees), etc. The methods used in [5, 10, 1] for depth images are mainly correlation-based stereo algorithms, which are not robust under the above mentioned difficulties.

In this paper, we present a more sophisticated algorithm dealing with the above mentioned difficulties for the first step. Our algorithm combines pixel-based and region-based approaches to analyse the 1-D motion information for a translational video sequence and to estimate the depth images at separate locations. A pixel-based approach using the Gabor transform and variational regularization is performed first. Then the region information from the segmentation is

combined with the pixel-based motion estimation results so that a region matching scheme using an affine transform can be applied. A novel contribution contained in our algorithm is a method to analyze the histograms of motion values for the pixels in each region, so that most of the sky regions can be identified and the true motion values for such regions can be determined. The high quality of our results will be shown by both motion maps and 3D reconstructions.

This paper is organized as follows: an overview of our algorithm will be give in section 2; in section 3, the detailed procedures for motion analysis and for obtaining depth images are presented; and in section 4, some results of 3D reconstructions based these depth images are shown, followed by a conclusion in section 5.

2 Algorithm Overview

2.1 Motion Analysis and Depth Images

As shown in Fig. 1, our system starts by filtering the two consecutive images I_t and I_{t+1} from a translational video sequence with a set of Gabor filters. Also I_t is put through a segmentation process using the mean shift algorithm [3] in which each region is formed by grouping pixels with similar color values and is represented by *one* color value for this region. The filtered versions of I_t and I_{t+1} are compared and a coarse 1-D motion map d_G is estimated. Another motion map d_R based on variational regularization using an edge-preserving functional is also estimated iteratively with motion values for each pixels initialized with zero. Then the histograms of motion values from d_G and d_R in each region of I_t (obtained from the segmentation) is compared in order to identify those regions without movements (zero motion). Once such regions with zero motion are identified, the motion values for the other regions of I_t are used to estimate a set of affine transform parameters by least squares, so that the matching relation for the pixels in this region with their corresponding pixels in I_{t+1} can be represented by the resulting affine transform. The affine parameters for each region are further adjusted using a descent-based region matching technique, and these adjusted affine parameters can be used in turn to calculate a more refined motion map. Once we get this final motion map for I_t, the depth image for the location of I_t is obtained using the reciprocal values of the 1-D motion values for each pixel. Because we are dealing with a translational video sequence in this paper, and similar to the relations between disparity and depth for parallel stereo, the 1-D motion value is in a reciprocal relation with the depth value, up to a scale factor.

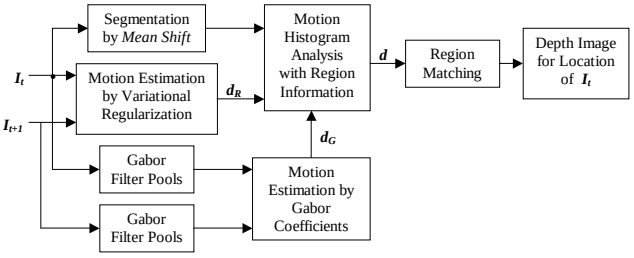


Figure 1. Block architecture for motion analysis

2.2 Construction of 3D Model

After we obtain the depth images for several locations along the 1-D camera trajectory, we will combine some depth images through shifting into one reference location. The shifting is controlled by depth values.

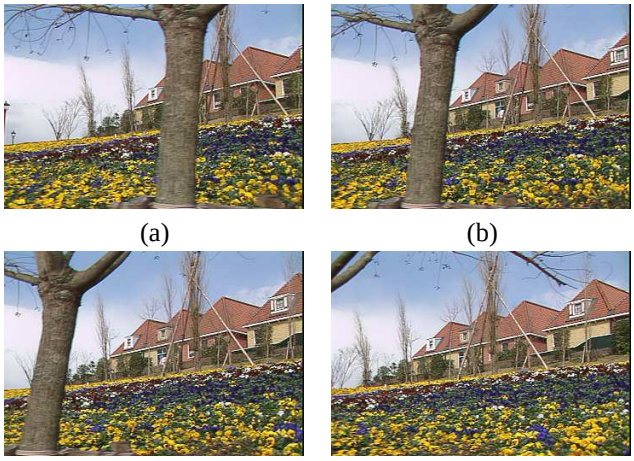
3 1-D Motion Analysis

Among the large amount of literature and algorithms for optical flow and motion estimation, differential techniques [6][8] with variational regularization form a major class. These techniques involve a functional including the displaced frame difference and a smoothing term, and usually descent-based methods are used to minimize the functional by solving its associated Euler-Lagrange equations. Recently, Brox et al. significantly improved this approach by embedding a multiresolution strategy and gradient constancy to a nonlinear objective functional, and obtained the best results until now for some standard test sequences like *Yosemite* [2]. Kim et al. used a similar functional with a modified regularization term and, to handle large motion fields, also used a coarse-to-fine scheme and solve the associated Euler-Lagrange equation using recursive iterations [7].

On another hand, the functionals used in the variational regularization approach usually do not take the occlusion effect into account, i.e., the objective functional that this approach tries to minimize is the displaced frame difference between *all* the pixels of I_t and their corresponding pixels in I_{t+1}. Due to this reason, after iterative calculations to minimize such objective functionals, those background pixels (which should be occluded) along the foreground objects usually have motion values similar to the motion values of those foreground pixels, since the iteration process also try to find a solution for such occluded background pixels. This will bring wrong motion values for such occluded background pixels. For example, in [7], the video sequence *Flower Garden* was used – the scene consists of

a tree in the foreground, and several houses with other trees and shrubs as middle objects, plus the sky as background – and from the result of its motion maps, most of the sky areas are merged with the middle objects and even with the twigs of the foreground tree. Therefore, although the displaced frame difference between I_t and I_{t+1} can be minimized to a small value which is good enough for some other purposes like compression and coding, the motion values estimated by variational regularization approach could not satisfy the purpose of 3D model constructions, since part of or most of the untextured background areas will be merged with the foreground objects, especially when those foreground objects have complex geometries.

We will try to solve this problem by comparing the results from the variational regularization approach with the motion estimation results from the Gabor transform and image segmentation. The video sequence that we use is also the *Flower Garden*, which consists of 150 frames. The sequence is taken along a straight line, and is approximately equi-distant for any two consecutive images. The maximum horizontal motion is about 6 pixels/frame. We show the *5th*, *22nd*, *35th* and *65th* images in Fig. 2, in which three of them



(a) (b) (c) (d)

Figure 2. Original images in *Flower Garden*: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

contain the foreground tree and therefore the motion estimation for these images is more difficult than for those without the tree. We will also show the motion estimation results for these images.

3.1 Motion Estimation by Variational Regularization Approach

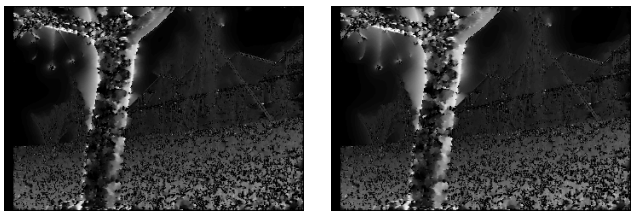
Similar to the general variational regularization approach, our objective functional for 1-D motion estimation also contains a data fidelity term and a regularization term for smoothing control:

$$E(d_R) = \iint [I_t(x,y) - I_{t+1}(x - d_R, y)]^2 \,\mathrm{d}x\,\mathrm{d}y + \lambda \iint \left\{ \frac{1}{(1+I_{t,x}^2)^2} d_{R,x}^2 + \frac{1}{(1+I_{t,y}^2)^2} d_{R,y}^2 \right\} \mathrm{d}x\,\mathrm{d}y \tag{1}$$

where λ is a regularization parameter, $d_{R,x}$ and $d_{R,y}$ are derivatives of $d_R(x,y)$ in x and y directions respectively, and similarly for $I_{t,x}$ and $I_{t,y}$. The minimization of (1) to estimate d_R is carried out by applying a gradient descent method to solve its associated Euler-Lagrange equation with respect to d_R:

$$\frac{\partial d_R}{\partial t} = [I_t(x,y) - I_{t+1}(x - d_R, y)] \times I_{t+1,x}(x - d_R, y) - \lambda \left\{ \frac{\partial}{\partial x} \left[\frac{d_{R,x}}{(1+I_{t,x}^2)^2} \right] + \frac{\partial}{\partial y} \left[\frac{d_{R,y}}{(1+I_{t,y}^2)^2} \right] \right\}. \tag{2}$$

Unlike the coarse-to-fine scheme as in [2] and [7] to prevent the solution to fall into local minima, we just use the original images and d_R are initialized with zero for all pixels. As shown in Fig. 3 in which brighter intensities represent larger motion values (and less depth) and darker intensities represent smaller motion values (and more depth), we can see that with the increase of iteration numbers, d_R



(a) (b)

Figure 3. d_R for $22nd$ image after different numbers of iterations: (a) 2500; (b) 4000.

could reach their true values for those pixels with small movements (like those houses and shrubs), and could not completely reach their true values for the pixels with large movements (like the foreground tree) since they fell into local minima. Most important for 3D reconstruction purpose is that the motion values for those background pixels (sky) leave their true values (zero) and approach the motion values of their foreground objects with the increase of iteration numbers. Therefore, as we stated in the beginning of this section, the objective functional can be further minimized with the increase of iteration numbers, but this does not fit our purpose of 3D model construction. Our solution for this dilema is to use fewer iterations, such that most of the pixels

with small movements can reach their true motion values, and most of the background pixels with zero motion stay where they are; the finding of large motion values for those foreground objects can be left to some other techniques (as we show later).

For the images in Fig. 2, we used 800 iterations for their motion estimation, and the results are shown in Fig. 4. We can see that the values of d_R for most of the background pixels are correct, and the majority part of pixels with small motions also have correct motions values.

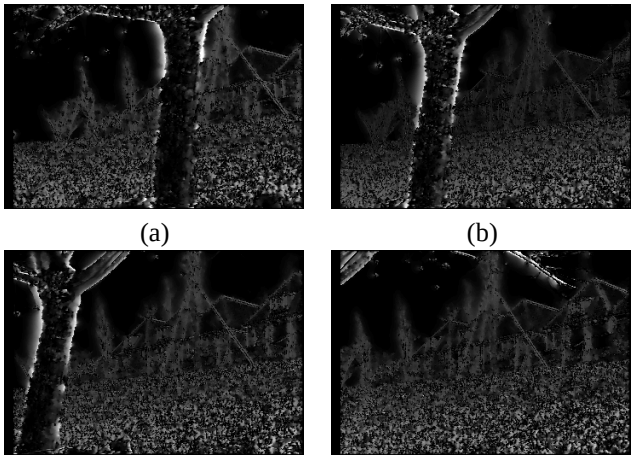


Figure 4. d_R maps after 800 iterations: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

3.2 Motion Estimation by Gabor Transform

The method that we used for 1-D motion estimation through the Gabor transform is mainly based on the algorithm in [4], in which a set of quadrature-pair Gabor filters are used. The Gabor functions are Gaussian functions modulated by complex sinusoids. Each quadrature-pair Gabor filter is a set of discretized samples of a Gabor function with different tuning frequencies, and is used for the filtering of the stereo images to obtain the approximate Gabor transform coefficients at those frequencies. Assume that the outputs of k^{th} filter pair are $G^k_{I_t}(x,y)$ and $G^k_{I_{t+1}}(x,y)$ for I_t and I_{t+1} respectively. Then the 1-D motion $\hat{d}_G \in [0, d_{max}]$ for a position (x,y) in I_t is determined as:

$$\hat{d}_G = \arg\min_{d_G} \sum_k \big[|Re\{G^k_{I_t}(x,y)\} - Re\{G^k_{I_{t+1}}(x-d_G,y)\}| + |Im\{G^k_{I_t}(x,y)\} - Im\{G^k_{I_{t+1}}(x-d_G,y)\}| \big] \quad (3)$$

where $Re\{G^k_{I_t}(x,y)\}$ and $Im\{G^k_{I_t}(x,y)\}$ are the real and imaginary parts of $G^k_{I_t}(x,y)$, and similarly for $G^k_{I_{t+1}}(x,y)$.

The reason that we choose the Gabor transform for motion estimation is its robustness in the sense that we do not need to determine any block size as in the cases of block-based stereo algorithms (like correlation-based stereo); the estimation process is done pixel-by-pixel independently of the scene.

We used three central frequencies $\{\pi/16, \pi/8, \pi/4\}$ as the tuning frequencies of the Gabor filters, and each filter is tuned to four directions 0°, 45°, 90° and 135°. The 1-D motion maps d_G estimated by (3) for the four images in Fig. 2 are shown in Fig. 5. We can see that these results are good for pixels with apparent motions. However, for part of the pixels with zero motions but near some middle and foreground objects, their motion results tend to be confused with the motion values for those objects.

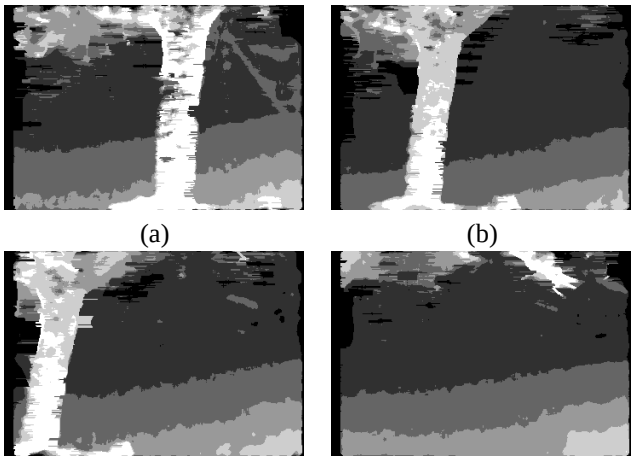


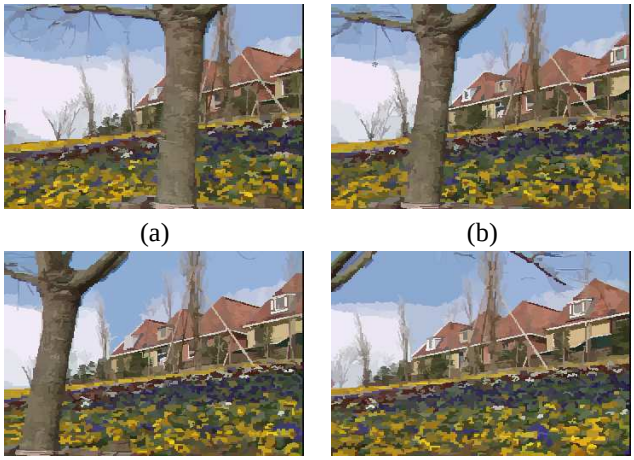
Figure 5. d_G maps from Gabor transform: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

3.3 Region-Based Motion Analysis

Comparing Fig. 4 and Fig. 5, we can find that the results from the two methods are inter-complementary, in which the results from variational regularization approach are good for zero and small motions and the results from the Gabor transform are good for large as well as for small motions. Therefore, we need to complement the two kinds of results from each other and obtain one good motion map for the whole motion range. In order to do that, we need to consider them in groups of connected pixels that fall in the same kind of regions that should have similar motion values. Thus, we need to have region information from segmentation applied to images I_t.

We applied the mean shift segmentation algorithm [3] to the images I_t, and the segmentation results for the four

images in Fig. 2 are shown in Fig. 6. Each region is indicated by one color value. Comparing Fig. 6 with their original images in Fig. 2, we can find that the mean shift algorithm could not identify some tiny features, which are missing after segmentation (e.g., some twigs on the tree, and part of the shrubs). To alleviate such a problem, we have performed an edge-detection by Canny detector on I_t and on Fig. 2, and then compare the detected edges between the two images to pick out the missing tiny contours. The new segmentation result for the *22nd* image by applying this method is shown in Fig. 7. Although we get most of the missing tiny contours back, this method also introduces some extra contours on some existing regions.



(a) (b) (c) (d)

Figure 6. Segmentation by *Mean Shifts* [3]: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

Once we have the region information, we can compare and analyse the histograms of motion values from d_R and d_G for each region. For example, as shown in Fig. 8 for a sky region in the $5th$ image which is between the upper twigs and the foreground tree, the histogram from d_R is mainly located around zero which is the correct motion value for this region, while the histogram from d_G is spread

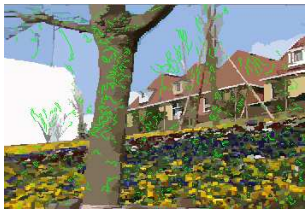
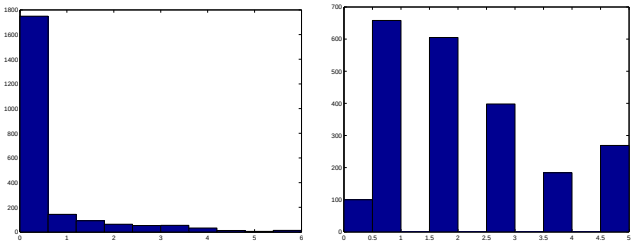


Figure 7. New segmentation of image 22 with tiny contours recovered.

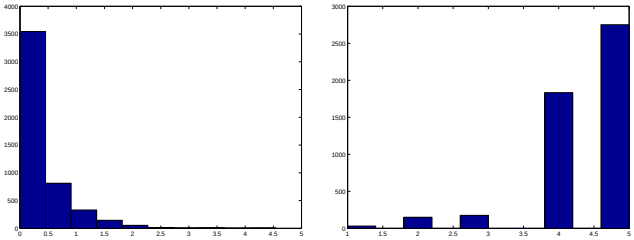
across the whole range. As another example shown in Fig. 9



(a) (b)

Figure 8. Histograms for a sky region in $5th$ image : (a) from d_R; (b) from d_G.

for a region of the foreground tree in the $22nd$ image, we can see that the histogram from d_R is mainly located around zero while the histogram from d_G is mainly located around the highest motion values (which are correct). From our



(a) (b)

Figure 9. Histograms for a foreground tree region in $22nd$ image : (a) from d_R; (b) from d_G.

previous analysis, we already conclude that the motion values from d_R are good for zero and small motions, while the motion values from d_G are good for large motion and the small motion values. Therefore, for the two cases like in Fig. 8 and Fig. 9, we can restore their motion values for that region from d_R (zero motion) and d_G (large motion) respectively according to the above criteria.

The adjusted motion maps d after analysis of the histograms of motion values for each region are shown in Fig. 10. Although we identified most of the sky regions now, most of the other regions with large motions values are still in a coarse stage since the motion values from the Gabor transform are integers (e.g., those slope regions with quantization effects). We still need to further refine those regions by region matching techniques.

We assume that the coordinates $(x, y)^T$ of each pixel in a region in I_t is related to its corresponding pixel $(x_{t+1}, y_{t+1})^T$ in I_{t+1} by an affine transform. In the 1-D case, we have:

$$x_{t+1} = a_{11}x + a_{12}y + a_{13}. \quad (4)$$

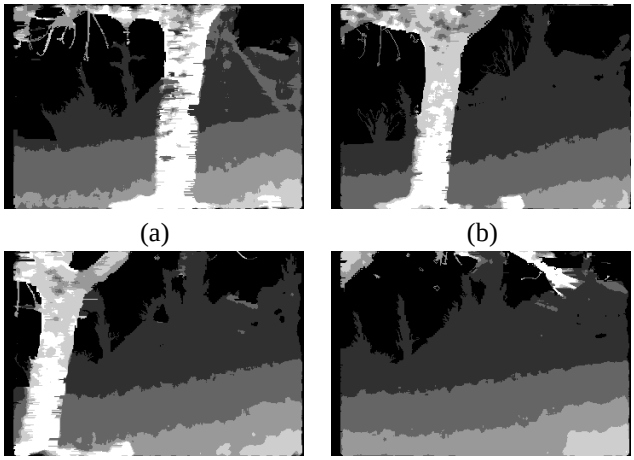


Figure 10. Motion maps after histogram analysis: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

Therefore, the motion value $d(x, y)$ is related to these affine parameters by

$$d(x, y) = x - a_{11}x - a_{12}y - a_{13}. \quad (5)$$

Thus, the estimated $d(x, y)$ for each pixel in one region from the results shown in Fig. 10 can be grouped and used as known variables so that the affine parameters can be estimated from (5). Since each pixel in the region gives a set of equations as in (5), and for most of the cases, the number of pixels in a region is larger than the number of affine parameters (three for 1-D affine transform), the estimation of the three parameters ($a_{11} \sim a_{13}$) can be done by least squares, implemented using singular value decomposition (SVD). Then, once the affine parameters are estimated, a new motion map $d(x, y)$ for each pixel in the region can be in turn calculated by (5).

Then we can go further to achieve a region matching scheme by updating those affine paremeters. The error function that we need to minimize for each region is:

$$E_A = \sum_{(x,y)\in W_i} [I_{t+1}(a_{11}x+a_{12}y+a_{13}, y)-I_t(x, y)]^2 \quad (6)$$

where W_i represents a region. We need to minimize (6) by updating affine parameters $\mathbf{a} = [a_{11}, a_{12}, a_{13}]^T$ iteratively using least squares with Taylor expansion. Assume $\mathbf{X} = [x, y, 1]^T$. Let $\hat{\mathbf{a}}$ be the current estimate of affine parameters, and $\mathbf{a} = \hat{\mathbf{a}} + \Delta\hat{\mathbf{a}}$. Then expand I_{t+1} around the current estimate

$$I_{t+1}(\mathbf{a}^T\mathbf{X}, y) \approx I_{t+1}(\hat{\mathbf{a}}^T\mathbf{X}, y) + \Delta\hat{\mathbf{a}}^T\mathbf{X}I_{t+1,x}(\hat{\mathbf{a}}^T\mathbf{X}, y), \quad (7)$$

where this first order expansion is valid only when $\hat{\mathbf{a}}$ is close to $\mathbf{a}$. This is the reason that we start region matching with the result from pixel-based approach, rather than doing it from the very beginning without pixel-based results. Substituting the above first order expansion into (6), the error function becomes:

$$E_A(\Delta\hat{\mathbf{a}}) = \sum_{(x,y)\in W_i} [\psi^T\Delta\hat{\mathbf{a}} - D]^2 \quad (8)$$

where $\psi = I_{t+1,x}(\hat{\mathbf{a}}^T\mathbf{X}, y)\mathbf{X}$ and $D = I_t(x, y) - I_{t+1}(\hat{\mathbf{a}}^T\mathbf{X}, y)$. The iterative solution of (8) by least squares is:

$$\Delta\hat{\mathbf{a}} = [\sum_{(x,y)\in W_i} \psi\psi^T]^{-1} \sum_{(x,y)\in W_i} D\psi. \quad (9)$$

After this region matching, the final results for the motion maps of Fig. 2 are shown in Fig. 11.

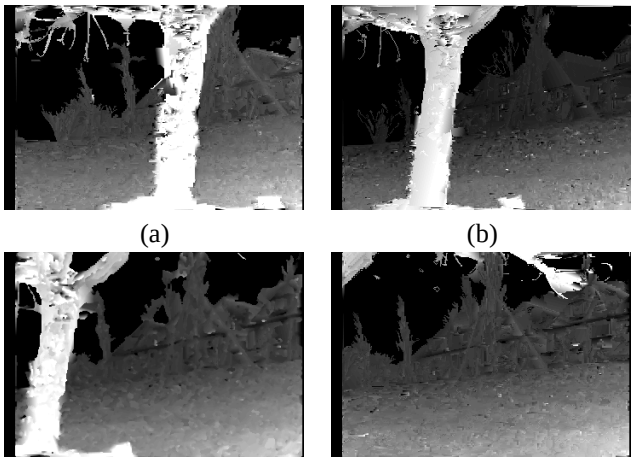


Figure 11. Final Motion maps after region matching: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

Then, for each location, the depth value $z(x, y)$ for a pixel at (x, y) can be obtained as:

$$z(x, y) = \frac{Bf}{d(x, y)} \quad (10)$$

where B is the baseline distance between I_t and I_{t+1}, and f is the focal length.

4 3D Reconstruction Using the Estimated Motion Maps

We show in this section some 3D reconstructions based on the motion or depth images we obtained. We set up the 3D models in OpenGL using 3D point arrays, and using *orthographic* projection mode for the rendering of novel

views. The most suitable way in doing this is to use triangular mesh arrays. However, for a scene with complex geometry like *Flower Garden*, object separation or 3D segmentation has to be done first so that only 3D points on the same object surfaces are connected by triangular meshes, otherwise different objects like the foreground tree and middle objects (houses) would be connected when the sideviews are rendered. Since this topic belongs to another important issue which deals with the combining of different depth images together, we only use point arrays for our 3D models because the main focus of this paper is on how to obtain depth images with high quality.

We first show in Fig. 12 some separate reconstructions based on each depth image on the four locations of Fig. 11 respectively. Fig. 12(a) is rendered by rotating about 10° around y-axis (vertical axis) from the original viewpoint to the right. Fig. 12(b) is rendered by rotating about 20° around y-axis to the left, then rotating up 15° around x-axis (horizontal axis). Fig. 12(c)(d) are rendered by rotating about 10° around y-axis to the right, then rotating up 10° and 5° around x-axis respectively. From these reconstructions, we can see that the sky has more shifting than the foreground scenes since it has the largest depth, and the occlusion from the foreground trees and shrubs on the sky can be clearly seen (black areas). Also, the linear variation of the depth values for the slanted slope surface can also be seen, especially from Fig. 12(d). All these facts indicate that the complex geometric structures detected by our algorithm are largely correct.

Then we try to combine the two depth images ($5th$ and $65th$) and their textures together with $5th$ image as reference location. Since we assume that there is only translational shifting for the whole video sequence, the homogeneous transformation between the two locations is represented by only one parameter K for the horizontal translation. Therefore, to shift the pixels (x, y) of $65th$ image with depth value $z(x, y)$ to their corresponding image coordinates (x_0, y_0) in the $5th$ image location, the x_0-components can be calculate as

$$x_0 = x + \frac{K}{z} \tag{11}$$

and $y_0 = y$, where K is a constant determined by the baseline distance and the focal length (we used $K = 60$ for the $5th$ and $65th$ images). We show in Fig. 13 the novel views after combining those two depth images without the foreground tree (disregard those pixels with lowest $z(x, y)$ values) in order to clearly show the fusion of the two images for those middle objects. From Fig. 13(a), we can see that the occluded areas in the $5th$ image (the areas behind the foreground tree) is recovered after combining the $65th$ image, and the missing parts on the right side of the $5th$ image and on the left side of the $65th$ image are also filled into one image. A little amount of discrepancy on the right side of

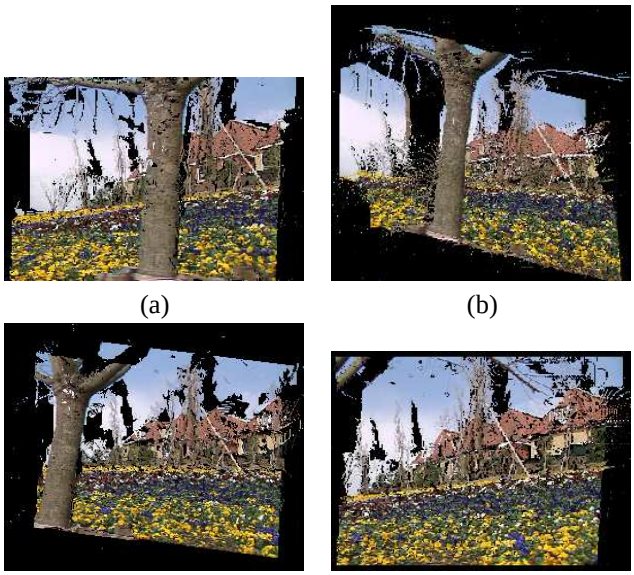
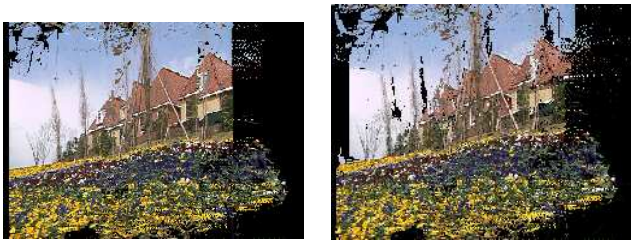


Figure 12. Separate reconstruction for different locations: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

Fig. 13(a) can be seen for the right-most house. The reason for this discrepancy is due to the fact that the motion of the *Flower Garden* sequence is not strictly horizontal. There are small vertical displacements between any two consecutive images, and from the $5th$ to the $65th$ image, these vertical displacements accumulate to a visible amount. Therefore, to construct the 3D model for the whole scene, we need to extend our motion estimation algorithm to 2D if we want to accurately combine the depth images for the whole sequence, rather than assuming pure translational motion and using (11) only. Also, in Fig. 13(b) we can see those occlusion areas on the background sky after a small rotation from the original viewpoint.

Finally we show in Fig. 14 the full fusion of the $5th$ and $65th$ images through their depth images with the $5th$ image as reference location. We can see that the tree branches on the middle top portion of the $65th$ image have moved to the very top-right part of the fused images without background, since these tree branches have lowest depth values in the $65th$ image (or largest motion values) and, while seeing from the location of the $5th$ image, the background for these tree branches should come from those images after the $65th$ image which we did not put into the fusion process. Also, from Fig. 14(b), we can see that there are no occlusion effects from the foreground tree after a small rotation from the original viewpoint, because those occluded areas in the $5th$ image are fused by the $65th$ image.



(a) (b)

Figure 13. The fusion of $5th$ and $65th$ images without the foreground tree (with $5th$ image as reference location): (a) direct reconstruction; (b) with rotation.

(a) (b)

Figure 14. Full fusion of $5th$ and $65th$ images (with $5th$ image as reference location): (a) direct reconstruction; (b) with rotation.

5 Conclusion and Future Work

We developed a hybrid 1D motion estimation algorithm which combines pixel-based and region-based approaches that can give depth images from translational video sequences with very high quality. The novelty of our algorithm lies in the fact that it provides a robust method to solve some long standing problems in structure-from-motion, like the identification of untextured areas (like sky) as background to some foreground objects with complex geometry, and keeping the boundaries of moving objects sharp and clear. These problems cannot be solved by either pixel-based or region-based approaches separately. Also, our algorithm can be extended to both 2D motion estimation and to disparity estimation problems.

The next step of our work will be to extend our algorithm to 2D cases, so that one 3D model with high quality can be obtained for the whole scene by combining accurate local 3D models through tracking the feature points or contours in the sequence to estimate the necessary homogeneous transformations. Also, for the combining of different depth images, we plan to use some more complex optimization algorithm like the bundle adjustment algorithm [11] to fuse all local 3D models with more accuracy, and use triangular meshes to represent the final 3D model efficiently and to eliminate outliers [9].

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada under the research network *LORNET*.

References

[1] C. Baker, C. Debrunner, and M. Whitehorn. 3D model generation using unconstrained motion of a hand-held video camera. *Proc. SPIE Conf. on Three-Dimensional Image Capture and Applications*, 6056, 2006.

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *Proc. European Conf. Computer Vision*, 4:25–36, 2004.

[3] D. Comaniciu and P. Meer. Mean-shift: a robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[4] D. Fleet. Disparity from local weighted phase-correlation. *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, pages 48–56, 1994.

[5] F. Galpin and L. Morin. Sliding adjustment for 3D video representation. *EURASIP Journal on Applied Signal Processing*, 10:1088–1101, 2002.

[6] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[7] J. Kim and T. Sikora. Hybrid recursive energy-based method for robust optical flow on large motion fields. *Proc. IEEE Int. Conf. Image Processing*, 1:129–132, 2005.

[8] B. Lucas and T. Kanade. An iterative image registration technique with application to stereo vision. *Proc. DARPA Image Understanding Workshop*, pages 121–130, 1981.

[9] G. Roth and E. Wibowo. An efficient volumetric method for building closed triangular meshes from 3-d image and point data. *Proc. Graphics Interface (GI)*, pages 173–180, 1997.

[10] S. Se and P. Jasiobedzki. Instant scene modeler for crime scene reconstruction. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 3:123–130, 2005.

[11] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. *Proc. Int. Workshop on Vision Algorithms: Theory and Practice*, pages 298–372, 1999.