Depth Estimation for Semi-Automatic 2D to 3D Conversion

Richard Rzeszutek*
Department of Electrical and Computer Engineering
Ryerson University
Toronto, Canada M5B 2K3
rrzeszut@ee.ryerson.ca

Raymond Phan
Department of Electrical and Computer Engineering
Ryerson University
Toronto, Canada M5B 2K3
rphan@ee.ryerson.ca

Dimitrios Androutsos
Department of Electrical and Computer Engineering
Ryerson University
Toronto, Canada M5B 2K3
dimitri@ee.ryerson.ca

ABSTRACT

The conversion of monoscopic footage into stereoscopic or multiview content is a difficult and time consuming task. A number of semi-automatic methods have been developed to speed up the process and provide some control to the user. However these methods require that the user provide detailed labels indicating the relative depth of objects in the scene. In this paper we present a method to automatically estimate depth in such a way that it is amenable to semi-automatic conversion. The method is designed to simplify the depth labelling task so that the user does not have to provide as many depth labels.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis— *depth cues, motion, stereo*

General Terms

Algorithms

Keywords

2D-to-3D conversion, motion estimation, depth estimation, computer vision

1. INTRODUCTION

Converting existing monoscopic, or "2D", content into stereoscopic, or "3D", content has become an active area of research, driven by the recent interest into 3D media by film studios and display manufacturers. It is currently a manual process, requiring many hours of labour for a normal feature-length film. As such, methods need to be developed to speed up the conversion process by either assisting the animator performing the conversion or through complete automation.

Current methods for 2D to 3D image conversion can be loosely grouped into two categories: semi-automated and automatic. A semi-automated method is one where the user has complete control over the resulting depth map but they must provide some sort of input labelling. There is no assumption about the underlying scene structure, instead it allows the user to indicate which regions of the scene are closer or farther away from the camera. A dense depth map is produced from the user's labels by using the underlying image features, e.g. edges, texture, etc. The resulting map is not metrically correct but perceptually consistent. This is sufficient for most viewers to find the depth effect to be pleasing. Surprisingly, very little research on semi-automated techniques has been performed, and can be beneficial for use in conversion for professional applications. The work by Guttmann *et al* [4], Phan *et al* and Rzeszutek *et al* [7, 9] and Wang *et al* [13] are good examples of semi-automated approaches.

For automated methods, there is little to no human interaction. Rather the user provides the technique with image or video footage and based on information present in the content it estimates the depth of elements in the scene. This problem is a well-studied one in computer vision and a powerful set of tools for solving this is known as Structure from Motion (SfM) has been developed. SfM can determine the three-dimensional structure of the scene, and the location of the camera relative to objects in the scene. This can be used to accurately extract object surfaces from a collection of images or a video sequence [3] or generate high quality depth maps [15]. It has even been used in a semi-automated manner to assist the user with converting 2D movies [14].

The main downside to SfM is that it is computationally intensive. SfM computes the camera locations or absolute point locations while it is often sufficient to determine the depths of objects relative to the *current camera*. There are a number of ways to do this, ranging from producing completely artificial depths inferred from an image [1] to using trained classifiers to predict the depth of a particular pixel [6, 16].

In this paper, we present an automated technique for extracting the relative depths of a scene that can ultimately serve as an input into a semi-automated method. Specifically, it is designed to be the input labelling to the semi-automated method described in [7, 9]. This has two benefits: the user does not need to provide as many labels and it can default to a purely automatic method if it is so desired.

2. METHODOLOGY

Our method focuses on estimating the depth of background elements in video footage. In general, labelling back-

*Corresponding author.

MM'12, October 29–November 2, 2012, Nara, Japan.

ground elements tends to be the most tedious task for semi-automated methods, as these are usually regions containing many objects and can experience complex motion. However, if not converted properly, the background can begin to visually conflict with foreground objects, e.g. background objects may appear *closer* to the camera than the foreground. By estimating background depth, the user can better focus on foreground objects and ensure their depths appear accurate. The method can be decomposed into these steps: 1) Preprocessing, 2) Disparity Estimation, and 3) Map Generation. We assume that the only information available is a frame sequence with the camera experiencing some arbitrary motion. We define this frame sequence as an ordered set $\mathcal{I}$ of N frames such that

$$\mathcal{I} = \{I_0, I_1, \ldots, I_{N-1}\}, \tag{1}$$

where I_k is the frame at index k. At the moment we also assume that the scene is rigid to simplify the development of the method. We will address how to handle non-rigid scenes in Section 4.

2.1 Preprocessing

The sequence preprocessing extracts two sets of data: the forward inter-frame optical flow $\mathcal{O}$ and a per-frame labelling $\mathcal{S}$ obtained from the over-segmentation of each frame in $\mathcal{I}$. As noted by [16], optical flow provides useful information regarding depth in the image even if it does not directly map back to disparity. We use a modified Horn-Schunck variant [11] to obtain the flow fields for the entire sequence. While more advanced methods are available, we have found that this variant provides adequate results without requiring overly long processing times. Similarly, we use a graph-based segmentation method [2] to generate our segmentation labelling, as it is extremely fast and quite reliable.

For the disparity estimation, we do not use the entire flow field. Rather, for each frame k we produce a set of tracks $\mathcal{T}_k$ such that

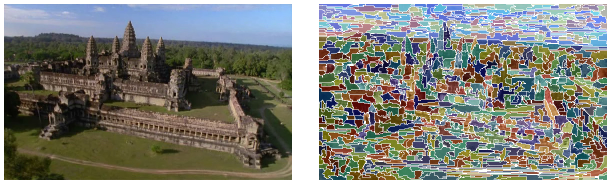
$$\mathcal{T}_k = \{\mathrm{t}_0, \mathrm{t}_1, \ldots, \mathrm{t}_{N_T}\}, \tag{2}$$

where the i-th track t_i is a set of two-dimensional points $\vec{p}_j \in \mathrm{t}_\mathrm{i}$ recording how that particular feature moves over time.

The initial locations of the features t_0 are chosen to be the centroids of the segmented regions in $\mathcal{S}$. This is important as these centroids are located in homogeneous image regions and will be useful when generating the dense depth map later on. The points are tracked using a simplified version of the Particle Video tracker [10], omitting the point insertion and position optimization stages as they are not needed. The relative displacement of the point is far more important than its absolute position in the context of our application. Furthermore, the length of the tracks are relatively short so the cumulative error using only the optical flow is small.

Figure 1 shows how the point locations are generated using a frame from the "Angkor Wat"[1] sequence. The image is segmented using [2], and the centroid locations are used as the initial positions of the tracks. Figure 2 shows the point locations obtained from the region centroids.

[1] `http://www.cad.zju.edu.cn/home/gfzhang/projects/videodepth/more.html`



(a) Original Image (b) Over-segmentation

Figure 1: An example of an over-segmented from the "Angkor Wat" sequence.

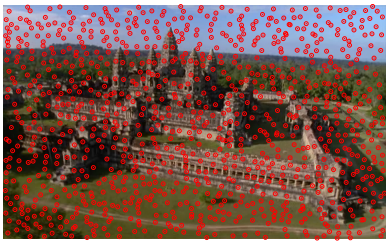


Figure 2: Feature point locations using the region centroids.

2.2 Disparity Estimation

The disparity is estimated by exploiting the epipolar geometry between pairs of frames along a track. For each frame pair, $\mathcal{P}_{k,l} = \{I_k, I_l\}$ where $k < l$, we obtain a set of point disparities by rectifying the two frames. To perform the rectification, we use the polar method described by Pollefeys *et al* [8]. This method is not sensitive to the location of the epipoles, compared to traditional rectification methods based on planar homographies.

Given the fundamental matrix $\mathbf{F}_{k,l}$ between the frame pair the two epipoles, $\vec{e}_k$ and $\vec{e}_l$, are obtained as the left and right null spaces of $\mathbf{F}_{k,l}$. Because all points must map to epipolar lines in the corresponding image [5], the epipoles can act as the origin of a polar coordinate system as points "slide" along epipolar lines. Transforming a point $\vec{p} = (x, y)$ into its polar representation $\vec{p'} = (\rho, \theta)$ in any particular frame can be done by $\rho = \|\vec{p} - \vec{e}\|$, and $\theta = \angle(\vec{p} - \vec{e})$. $\vec{e}$ is the *non-homogeneous* coordinate frame's epipole. Figure 3 provides an example of two sets of feature points after being converted into polar coordinates.

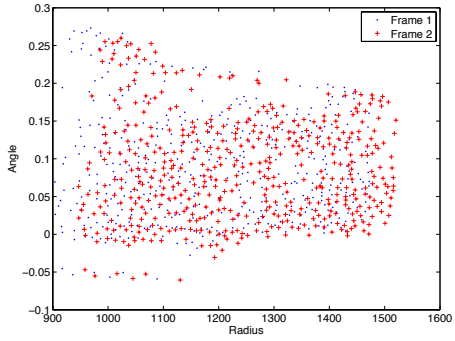


Figure 3: The distribution of feature points in polar coordinates. Under camera motion the distributions will see little change but the average values of ρ and θ will change.

The disparity of a point $\Delta_{k,l}(\vec{p})$ can be found by

$$\Delta_{k,l}(\vec{p}) = |\rho_l - \rho_k|. \quad (3)$$

We take the absolute value of the disparity as the sign is usually an indicator of the direction of the camera motion. Because ρ indicates the distance from the epipole, this effectively encodes the point's distance from the camera. Also, θ encodes the rotation around the epipole, but it can be ignored when determining disparity.

Zhang *et al* [16] argue that signed disparity can be used to infer whether the object is in front of or behind the display. However, this will cause issues for objects moving at different speeds and directions in a frame. A well-designed depth-based image rendering (DBIR) system will be able to properly project objects to be in front or behind of the display, and is not required to be modelled during the estimation stage. For a static scene, the magnitude of the disparity is directly related to an object's distance from the camera and is an important result.

Due to small baselines and tracking errors, the estimate of $\mathbf{F}$ can be extremely noisy and the disparity between any two frames will vary wildly. To deal with this, we use "parallel chaining" [12] that was initially developed for projective factorization. The chaining procedure involves determining the projective depths between first frame and all other frames, producing the sequence $(1,2),(1,3),(1,4),\ldots$ and so on. For our application, we produce a parallel chain of disparities for frame k rather than just a single disparity between the frame and another frame. This chain is defined as a set of disparities $\mathfrak{D}_k = \{\Delta_{k,k+1}, \Delta_{k,k+2}, \ldots, \Delta_{k,k+N_T-1}\}$. In order to compare disparities in the chain we scale $\Delta_{k,l}$ so that it is on $[0,1]$.

Under pure translation the disparities between two images will always vary by a scaling factor. After rescaling, the disparities in the parallel chain should be very similar. This does not hold true for other motions but in those instances the tracks terminate quickly and will approximate a translation. However, if the tracking is not perfect then the disparities will not vary by a scaling even under pure translation. This can be seen in Figure 4. It represents the parallel chain obtained for the first frame in the Angkor Wat sequence.

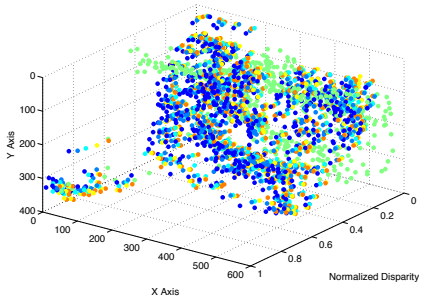


Figure 4: Overlaid disparities estimates for each frame pair in the parallel chain. A normalized depth of 1 represents objects close to the camera while a normalized depth of 0 represents those far away from the camera.

The point cloud is "fuzzy" because of errors in tracking and errors in estimating $\mathbf{F}_{k,l}$. But the chain represents multiple disparity estimates for the same point over multiple frames and *on average* the disparities estimates should be correct. Therefore we assume that the disparity of frame k is the average of the disparities in $\mathfrak{D}_k$. The noticeably cleaner result can be seen in Figure 5.

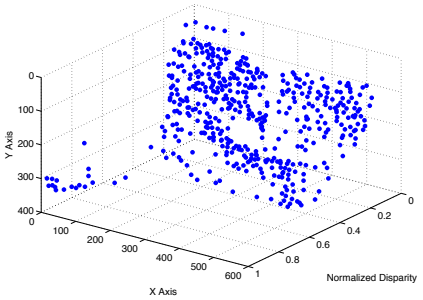


Figure 5: Disparity estimate after averaging the results of the parallel chain for the first frame of the Angkor Wat sequence.

2.3 Map Generation

We generate a dense map in one of two ways. The first method is to simply assign disparity values to the segmentation of the frame (Figure 1b). This produces an incomplete, but relatively dense, disparity map that can be used to allow the user to quickly preview the results. This is shown in Figure 6.

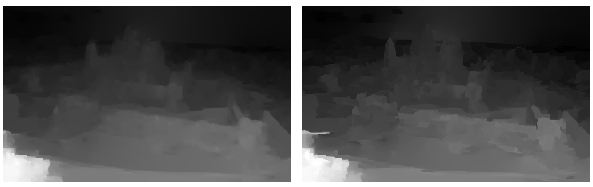
Figure 6: Fast disparity map produced by colouring in the labels shown in Figure 1b.

To produce a complete dense map, we utilize the disparity values to produce a seed labelling for [7]. The seed labels can be produced by either assigning single pixels the disparity value (a sparse labelling) or using the incomplete map in Figure 6. The difference between using the two label styles can be seen in Figure 7. The primary difference is that the sparse labelling tends to look more natural for single frames though can result in "depth leakage" [9] if volumetric processing is used.

3. RESULTS

In this section we present our results for the Angkor Wat sequence against a "ground truth". We use the results of [15] as their method provides high quality depth estimates for rigid scenes. The dense maps were produced by using the incomplete depth maps from Section 2.3 as seeds to [7] with the depth priors disabled. The results are in Figure 8.

While our results do not match those of [15], the results are comparable. Our method is significantly simpler to implement but it is still able to generate depth estimates that reasonably approximate the ground truth.



(a) Sparse Seeds (b) Semi-dense Seeds

Figure 7: A comparison between using sparse seeds and using the incomplete depth representation.

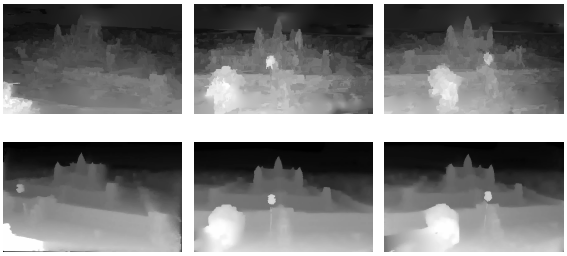


Figure 8: Comparison of results for the Angkor Wat sequence. The top row are the results of our method while the bottom row are the ground truths from [15].

4. CONCLUSION AND FUTURE WORK

In this paper we have presented a method for automated depth estimation method that can be easily integrated with a semi-automated pipeline. The depth estimation method works by using the epipolar relationship between image pairs. We minimize the overall error by averaging the disparity estimates across multiple image pairs. The resulting depth values can be sent as input into a semi-automated depth generation system so that the user can add their own labels as needs.

We are actively investigating methods for dealing with non-rigid scenes. At the moment, our method will produce incorrect depths for scenes where objects move independently of the camera and the user will be required to clean these up. However, if the user provides their own labels for the various foreground objects, these can be easily excluded during the depth estimation. For a more automated approach, we can use the inliers returned from the estimation of **F**, normally done with a robust method such as RANSAC, so that any feature points *not* corresponding to this model to be rejected.

5. REFERENCES

[1] L. J. Angot, W.-J. Huang, and K.-C. Liu. A 2D to 3D Video and Image Conversion Technique based on a Bilteral Filter. *Proc. SPIE Electronic Imaging - Three-Dimensional Image Processing (3DIP) and Applications*, 2010.

[2] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004. 10.1023/B:VISI.0000022288.19776.77.

[3] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. Seitz. Multi-view stereo for community photo collections. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1 –8, oct. 2007.

[4] M. Guttmann, L. Wolf, and D. Cohen-Or. Semi-automatic Stereo Extraction from Video Footage. *Proc. IEEE Intl. Conf. on Computer Vision (ICCV)*, 2009.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[6] J. Konrad, G. Brown, M. Wang, P. Ishwar, C. Wu, and D. Mukherjee. Automatic 2d-to-3d image conversion using 3d examples from the internet. volume 8288, page 82880F. SPIE, 2012.

[7] R. Phan, R. Rzeszutek, and D. Androutsos. Semi-automatic 2d to 3d image conversion using scale-space random walks and a graph cuts based depth prior. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 865 –868, sept. 2011.

[8] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *Int. J. Comput. Vision*, 59(3):207–232, Sept. 2004.

[9] R. Rzeszutek, R. Phan, and D. Androutsos. Semi-automatic synthetic depth map generation for video using random walks. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1 –6, july 2011.

[10] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision*, 80:72–91, 2008. 10.1007/s11263-008-0136-6.

[11] D. Sun, S. Roth, and M. Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432 –2439, june 2010.

[12] B. Triggs. Factorization methods for projective structure and motion. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 845 –851, jun 1996.

[13] O. Wang, M. Lang, M. Frei, A. Hornung, A. Smolic, and M. Gross. Stereobrush: interactive 2d to 3d conversion using discontinuous warps. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '11, pages 47–54, New York, NY, USA, 2011. ACM.

[14] B. Ward, S. B. Kang, and E. Bennett. Depth director: A system for adding depth to movies. *Computer Graphics and Applications, IEEE*, 31(1):36 –48, jan.-feb. 2011.

[15] G. Zhang, J. Jia, T.-T. Wong, and H. Bao. Consistent depth maps recovery from a video sequence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(6):974 –988, June 2009.

[16] Z. Zhang, Y. Wang, T. Jiang, and W. Gao. Stereoscopic learning for disparity estimation. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pages 365 –368, may 2011.