Motion segmentation using occlusions

Abhijit S. Ogale, Cornelia Fermüller, Yiannis Aloimonos

Abstract

We examine the key role of occlusions in finding independently moving objects instantaneously in a video obtained by a moving camera with a restricted field of view. In this problem, the image motion is caused by the combined effect of camera motion (egomotion), structure (depth), and the independent motion of scene entities. For a camera with a restricted field of view undergoing a small motion between frames, there exists in general a set of 3D camera motions compatible with the observed flow field even if only a small amount of noise is present, leading to ambiguous 3D motion estimates. If separable sets of solutions exist, motion-based clustering can detect one category of moving objects. Even if a single inseparable set of solutions is found, we show that occlusion information can be used to find ordinal depth, which is critical in identifying a new class of moving objects. In order to find ordinal depth, occlusions must not only be known, but they must also be filled (grouped) with optical flow from neighboring regions. We present a novel algorithm for filling occlusions and deducing ordinal depth under general circumstances. Finally, we describe another category of moving objects which is detected using cardinal comparisons between structure from motion and structure estimates from another source (e.g., stereo).

Index Terms

motion, occlusions, segmentation, ordinal depth, video analysis

A. S. Ogale, C. Fermüller and Y. Aloimonos are with the Center for Automation Research, Dept. of Computer Science, University of Maryland, College Park, MD 20742.

Email: (ogale,fer,yiannis)@cfar.umd.edu

I. Introduction

Motion segmentation is the problem of finding independently moving objects in a video. This process is conceptually simple when the camera is stationary, and a variety of solutions exist under the general heading of background subtraction. However, if the camera itself is moving, a general and robust solution is still elusive, since the image motion is generated by the combined effects of camera motion, structure and the motion of independently moving objects, and isolating these three factors proves to be a difficult task.

Prior research can mostly be classified into two groups: (a) The approaches relying, prior to 3D motion estimation, on 2D motion measurements only [1]–[4]. The limitations of these techniques are well understood. Depth discontinuities and independently moving objects both cause discontinuities in the 2D optical flow, and it is not possible to separate these factors without 3D motion and structure estimation. (b) 3D approaches which identify clusters with consistent 3D motion [5]–[11] using a variety of techniques. Some techniques, such as [12] are based on alternate models of image formation like weak perspective. These additional constraints can be justified for domains such as aerial imagery. In this case, the planarity of the scene allows a registration process [13]–[16], and uncompensated regions correspond to independent movement. This idea has been extended to cope with general scenes by selecting models depending on the scene complexity [17], or by fitting multiple planes using the plane plus parallax constraint [18], [19]. Clearly, improvement in motion detection can be gained using temporal integration. Yet questions related to the integration of 3D motion and scene structure are not yet well understood, as the extension of the rigidity constraint to multiple frames is nontrivial.

Most techniques detect independently moving objects based on the 3D motion estimates, either explicitly or implicitly. Some utilize inconsistencies between egomotion estimates and the observed flow field, while some utilize additional information such as depth from stereo, or partial egomotion from other sensors. Nevertheless, the central problem faced by all motion-based techniques is that, in general, it is extremely difficult to uniquely estimate 3D motion from flow. Several studies have addressed the issue of noise sensitivity in structure from motion. In particular, it is known that for a moving camera with a small field of view observing a

scene with insufficient depth variation, translation and rotation are easily confused [20], [21]. Maybank [22] and Heeger and Jepson [23] have also shown that if the scene is sufficiently nonplanar, then the minima of the cost function resulting from the epipolar constraint lie along a line in the space of translation directions, which passes through the true translation direction and the viewing direction. In [24], an algorithm-independent stability analysis of the structure from motion problem has been carried out.

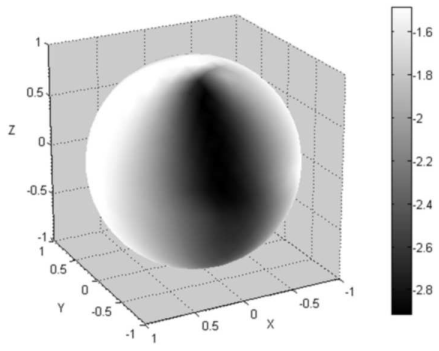

Fig. 1. Motion valley (dark) visualized as an error surface in the 2D space of directions of translation, represented by the surface of a sphere. The error is found after finding the optimal rotation and structure for each translation direction

Thus, given a noisy flow field, any motion estimation technique will yield a region of solutions in the space of translations instead of a unique solution; we refer to this region as the *motion valley.* Each translation direction in the motion valley, along with its best corresponding rotation and structure estimate, will agree with the observed noisy flow field. Fig. 1 shows a typical error function obtained using the method of Brodsky et al. [25]. Motion-based clustering can only succeed if a scene entity has a motion which does not lie in the background motion valley. In this paper, we go beyond motion-based clustering to show that even if motion estimation yields a single valley, ordinal depth from occlusions helps us find a new category of moving objects. We present an algorithm for finding ordinal depth from occlusions using three frames. The problem of finding depth orderings using occlusions has been previously addressed by some studies in the context of finding accurate 2D motion (e.g., [26]–[28]). In the following sections, we present a classification of moving objects and discuss algorithms for detecting each class, with particular emphasis on the role of occlusions.

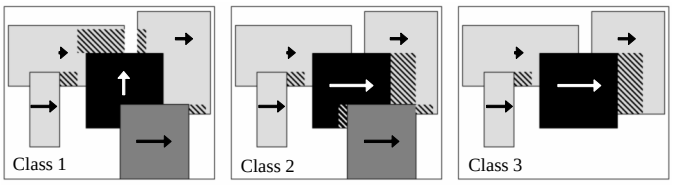

Fig. 2. Toy examples of three classes of moving objects. In each case, the black object is the independently moving object. Portions of objects which disappear in the next frame (i.e., occlusions) are shown in a dashed texture

II. Types of independently moving objects

We now discuss three distinct classes of independently moving objects; the moving objects belonging to *Class 1* can be detected using motion-based clustering, the objects in *Class 2* are detected by detecting conflicts between depth from motion and ordinal depth from occlusions, and objects in *Class 3* are detected by finding conflicts between depth from motion and depth from another source (such as stereo). Any specific case will consist of a combination of objects from these three classes. Fig. 2 shows illustrative examples of the three classes.

A. Class 1: 3D motion-based clustering

The first column of Fig. 2 shows a situation in which the background objects (non independently moving) are translating horizontally, while the black object is moving vertically. In this scenario, motion-based clustering approaches will be successful, since the motion of the black object is not contained in the motion valley of the background. Thus, *Class 1* objects can be detected using motion alone. Our strategy for quickly performing motion-based clustering and detecting *Class 1* objects is discussed in Section III.

B. Class 2: Ordinal depth conflict between occlusions and structure from motion

The second column of Fig. 2 shows a situation in which the background objects are translating horizontally to the right, and the black object also moves towards the right. In this scenario, motion estimation will not be sufficient to detect the independently moving object, since motion estimation yields a single valley of solutions. An additional constraint, which may be termed the *ordinal depth conflict* or the *occlusion-structure from motion (SFM) conflict*, needs to be used to detect the moving object.

Notice the occluded areas in the figure: we can use our knowledge of these occlusions to develop ordinal depth (i.e., *front/back*) relationships between regions of the scene. In this example, the occlusions tell us that the black object is *behind* the dark gray object. However, if we compute structure from motion, since the motion is predominantly a translation, the result would indicate that the black object is in front of the dark gray object (since the black object moves faster). This conflict between ordinal depth from occlusions and structure from motion permits the detection of *Class 2* moving objects. In Section IV, we present a novel algorithm for finding ordinal depth.

C. Class 3: Cardinal depth conflict

The third column of Fig. 2 shows a situation similar to the second column, except that the dark gray object which was in front of the black object has been removed. Due to this situation, the ordinal depth conflict which helped us detect the black object in the earlier scenario is no longer present. In order to detect the moving object in this case, we must employ cardinal comparisons between structure from motion and structure from another source (such as stereo) to identify deviant regions as *Class 3* moving objects. In our experiments, we have used a calibrated stereo pair of cameras to detect objects of *Class 3*. The calibration allows us to compare the depth from motion directly with the depth from stereo up to a scale. We use k-means clustering (with $k = 3$) on the depth ratios to detect the background (the largest cluster). The reason for using $k = 3$ is to allow us to find three groups: the background, pixels with depth ratios greater than the background, and pixels with depth ratios less than the background. Pixels not belonging to the background cluster are the *Class 3* moving objects. At this point, it may be noted that alternative methods exist in the literature (e.g., [6]) for performing motion segmentation on stereo images, which can also be used to detect *Class 3* moving objects.

III. MOTION-BASED CLUSTERING

Motion-based clustering is a chicken-and-egg problem: if we knew the background pixels, we could find the background motion, and vice versa. In the introduction, we have cited several novel

approaches which find motion clusters by concurrently performing segmentation and motion estimation. Here, we present a fast and simple method which consists of two steps.

1) Using phase correlation on two frames in the cartesian representation (to find 2D translation t_x, t_y) and in the logpolar representation (to find scale S and z-rotation γ), we obtain a four-parameter transformation between frames (see [29]). Phase correlation can be thought of as a voting approach [30], and hence we find empirically that these four parameters depend primarily on the background motion even in the presence of moving objects. This assumption is true as long as the background edges dominate the edges on the moving objects. This four-parameter transform predicts a flow direction at every point in the image. We select a set of points S in the image whose true flow direction lies within an angle of η_1 degrees about the direction predicted by phase correlation or its exact opposite direction (we use $\eta_1 = 45^\circ$) .
2) Optical flow values at the points in set S are used to estimate the background motion valley using the 3D motion estimation technique of Brodsky et al. [25]. Since all points in the valley predict similar flows on the image (which is why the valley exists in the first place), we can pick any solution in the valley and compare the reprojected flow with the true flow. Regions where the two flows are not within η_2 degrees of each other are considered to be *Class 1* independently moving objects (we use $\eta_2 = 45^\circ$) .

This procedure allows us to find the background and *Class 1* moving objects without iterative processes. The voting nature of phase correlation helps us to get around the chicken-and-egg aspect of the problem. To find optical flow, we can use any algorithm which finds dense flow (e.g., [31], [32]; we use the former). Although we have not used occlusions here, it is worthwhile to note that occlusions can be used to reduce the size of the motion valley.

IV. Ordinal depth from occlusion filling using three frames

A. *Why occlusions must be* filled*?*

Given two frames from a video, occlusions are points in one frame which have no corresponding point in the other frame. However, merely knowing the occluded regions is not sufficient to

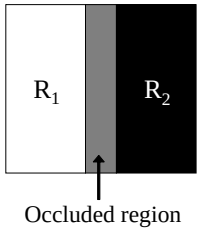

Fig. 3. If the occluded region belongs to R_1, then R_1 is behind R_2 and vice-versa

deduce ordinal depth. In Fig. 3, we show a situation where an occluded region O is surrounded by two regions R_1 and R_2 which are visible in both frames.

If the occluded region O belongs to region R_1, then we know that R_1 must be behind R_2, and vice versa.

This statement is extremely significant, since it holds true even when the camera undergoes general motion, and even when we have independently moving objects in the scene! Thus, we need to know *'who occluded what'* as opposed to merely knowing *'what was occluded'*. Since optical flow estimation provides us with a segmentation of the scene (regions of continuous flow), we now have to *assign flows to the occluded regions*, and *merge* them with existing segments to find ordinal depth.

B. Occlusion filling (rigid scene, no independently moving objects)

In the absence of independently moving objects, knowledge of the focus of expansion (FOE) or contraction (FOC) can be used to fill occluded regions. Since camera rotation does not cause occlusions [33], knowing the FOE is enough. In the simplest case, shown in Fig. 4a, where the camera translates to the right, if object A is in front of object B then object A moves more to the left than B, causing a part of B on the left of A to become occluded. Thus, if the camera translates to the right, occluded parts in the first frame always belong to segments on their left. For general egomotion: First, draw a line L from the FOE/FOC to an occluded pixel O. Then: (A) If we have an *FOE* (see Fig. 4b), the flow at O is obtained using the flow at the nearest visible pixel P on this line L, such that O lies between P and the FOE. (B) if we have an *FOC* (see Fig. 4c), then fill in with the nearest pixel Q on line L, such that Q lies between O and the FOC.

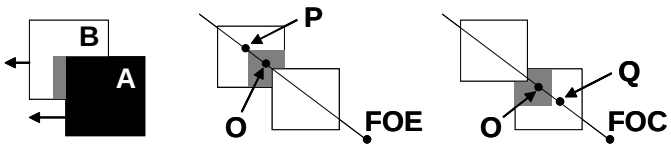

Fig. 4. Occlusion Filling: from left (a) to (c). Gray regions indicate occlusions (portions which disappear in the next frame)

C. Generalized occlusion filling (in the presence of moving objects)

In the presence of moving objects, even the knowledge of the FOE provides little assistance for filling occlusions, since the occlusions no longer obey the criteria presented above; a more general strategy must be devised. The simplest idea which comes to mind is the following: if an occluded region O lies between regions R_1 and R_2, then we can decide how to fill O based on its *similarity* with R_1 and R_2. However, *similarity* is an ill-defined notion in general, since it may mean similarity of gray value, color, texture or some other feature. Instead, we present below a novel and robust strategy utilizing optical flow alone (see Fig. 5) for filling occlusions in the general case using three frames instead of two.

Given three consecutive frames F_1, F_2, F_3 , we use an optical flow algorithm which finds dense flow and occlusions (e.g., [31], [32]; we use the former) to compute the following:

1) Using F_1 and F_2, we find flow $\vec{u}_{12}$ from frame F_1 to F_2, and the reverse flow $\vec{u}_{21}$ from frame F_2 to F_1. The algorithm also gives us occlusions O_{12} which are regions of frame F_1 which are not visible in frame F_2. Similarly, we also have O_{21}.
2) Using frames F_2 and F_3, we find $\vec{u}_{23}$ and $\vec{u}_{32}$, and O_{23} and O_{32}.

Our objective is to fill the occlusions O_{21} and O_{23} in frame F_2 to deduce the ordinal depth. The idea is simple: *O_{23} denotes areas of F_2 which have no correspondence in F_3. However, these areas were visible in both F_1 and F_2; hence in $\vec{u}_{21}$ these areas have already been grouped with their neighboring regions. Therefore, we can use the segmentation of flow $\vec{u}_{21}$ to fill the occluded areas O_{23} in the flow field $\vec{u}_{23}$*. Similarly, we can use the segmentation of $\vec{u}_{23}$ to fill the occluded areas O_{21} in the flow field $\vec{u}_{21}$. After filling, deducing ordinal depth is straightforward: if an occlusion is bounded by R_1 and R_2, and if R_1 was used to fill it, then R_1 is below R_2. This method is able to fill the occlusions and find the ordinal depth in a robust fashion.

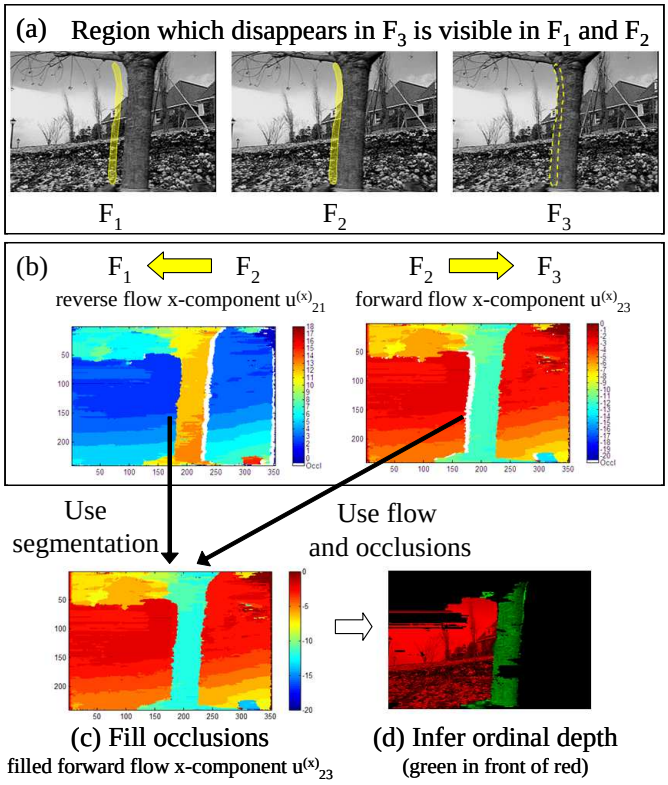

Fig. 5. Generalized occlusion filling and ordinal depth estimation. (a) Three frames of a video sequence. The yellow region which is visible in F_1 and F_2 disappears behind the tree in F_3. (b) Forward and reverse flow (only the x-components are shown). Occlusions are colored white. (c) Occlusions in $\vec{u}_{23}$ are filled using the segmentation of $\vec{u}_{21}$. Note that the white areas have disappeared. (d) Deduce ordinal depth relation. In a similar manner, we can also fill occlusions in $\vec{u}_{21}$ using the segmentation of $\vec{u}_{23}$ to deduce ordinal depth relations for the right side of the tree

V. Algorithm Summary

1) Input video sequence: $V = (F_1, F_2, ..., F_n)$
2) For each $F_i \in V$ do
   a) find forward $\vec{u}_{i,i+1}$ and reverse $\vec{u}_{i,i-1}$ flows with occlusions $O_{i,i+1}$ and $O_{i,i-1}$
   b) select a set S of pixels using phase correlation between F_i and F_{i+1}
   c) find background motion valley using the flows for pixels in S
   d) detect *Class 1* moving objects and background B_1
   e) find ordinal depth relations using results of step (a)
   f) for pixels in B_1, detect *Class 2* moving objects, and new background B_2
   g) if depth from stereo is available, detect *Class 3* objects present in B_2

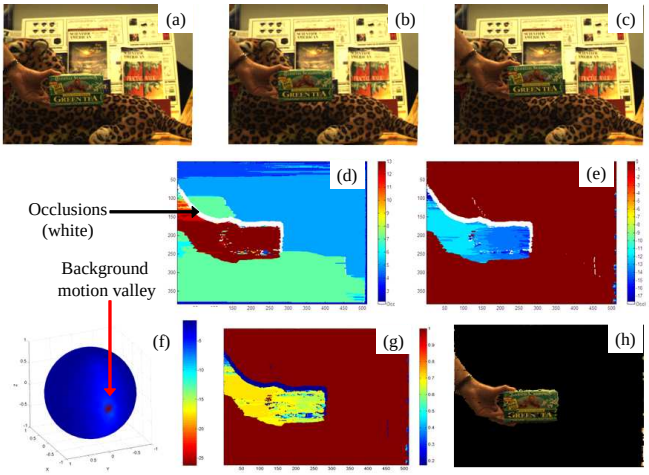

Fig. 6. Class 1: (a,b,c) show three frames of the *teabox* sequence. (d,e) show X and Y components of the optical flow using frames (b) and (c). Occlusions are colored white. (f) shows the computed motion valley for the background. (g) shows the cosine of the angular error between the reprojected flow (using the background motion) and the true flow. (h) shows the detected *Class 1* moving object

VI. Experiments

Fig. 6 shows a situation in which the background is translating horizontally, while a *teabox* is moved vertically. In this scenario, since the *teabox* is not contained in the motion valley of the background, it is detected as a *Class 1* moving object.

Fig. 7 shows three frames of a video in which the camera translates horizontally, while a *coffee mug* is moved vertically upward, and a *red Santa Claus toy* is moved horizontally parallel to the background motion. The *coffee mug* is detected as a *Class 1* moving object, while the *red toy* is detected as a *Class 2* moving object using the conflict between ordinal depth from occlusions and structure from motion. A handwaving analysis indicates that since the *red toy* is moving faster than the foreground boxes, structure from motion (since the motion is predominantly a translation) naturally suggests that the *red toy* is in front of the two boxes. But the occlusions clearly indicate that the *toy* is behind the two boxes, thereby generating a conflict.

Finally, Fig. 8 shows a situation in which the background is translating horizontally to the right, and the leopard is dragged horizontally towards the right. In this case, a single motion valley is found, the depth estimates are all positive, and no ordinal depth conflicts are present. (Although this case shows the simplest situation, we can also imagine the same situation as Fig. 7, with

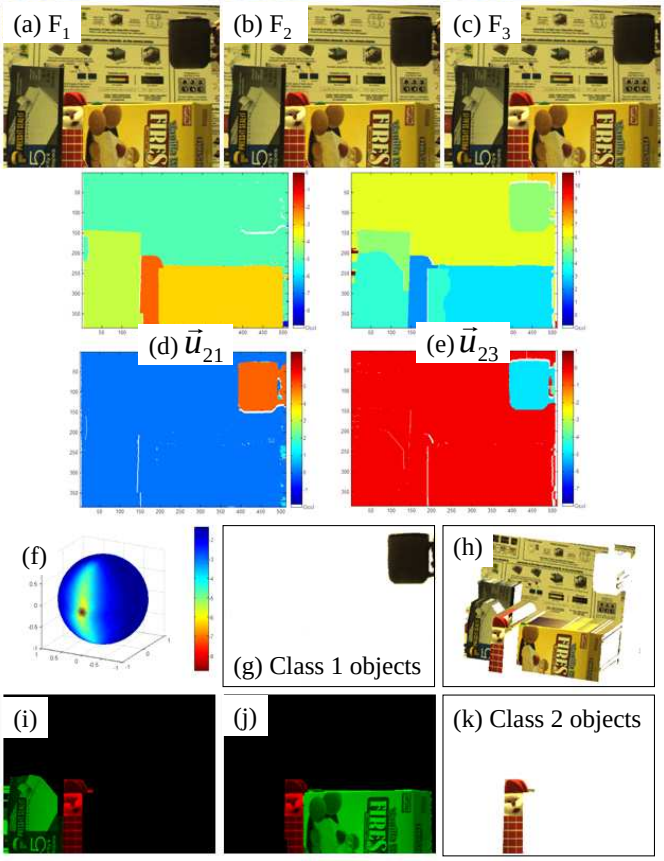

Fig. 7. (a,b,c) show three frames F_1, F_2, F_3 of the *santa-coffee* sequence. The camera translates horizontally to the left, hence the scene moves to the right. The *coffee mug* is lifted up, and the red toy *santa* is pulled by a person (not seen) to the right. (d) and (e) show optical flow $\vec{u}_{21}$ from frame F_2 to F_1, and $\vec{u}_{23}$ from frame F_2 to F_3 respectively. Note that each flow is shown as two images, with the X-component image above the Y-component image. Occlusions are colored white. (f) shows the estimated background motion. (g) shows the *coffee mug* detected as a *Class 1* object. (h) shows the computed structure from motion (SFM) for the background. Note that the toy *santa* appears *in front* of the two boxes. (i) and (j) show two ordinal depth relations obtained from occlusions which tell us that the santa (marked in red) *is behind* the boxes (marked in green). (k) shows the toy *santa* detected as a *Class 2* moving object using the ordinal depth conflict

the exception that the *red toy* does not move fast enough so as to appear in front of the two boxes and generate an occlusion-motion conflict.) In this case, depth information from stereo (obtained using a calibrated stereo pair of cameras) was compared with depth information from motion. k-means clustering (with $k = 3$) of the depth ratios was used to detect the background (the largest cluster). The pixels which did not belong to the background cluster are labeled as *Class 3* moving objects. To find optical flow, we have used the algorithm discussed in [31]; some of our code for finding optical flow may be found at *www.cs.umd.edu/users/ogale*.

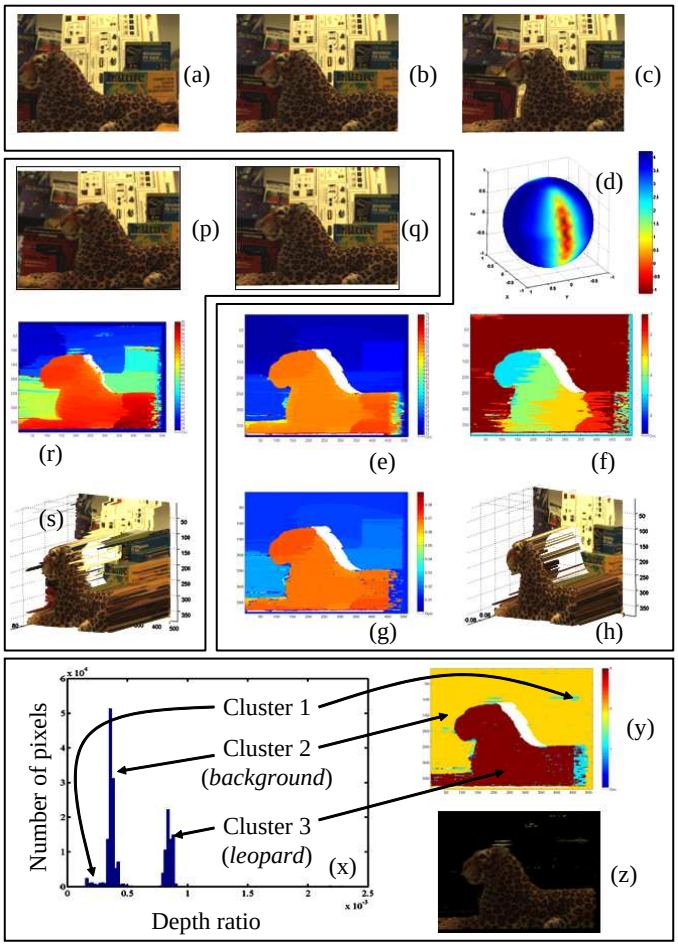

Fig. 8. Class 3: (a,b,c) show three frames F_1, F_2, F_3 of the *leopardB* sequence. (d) shows the computed motion valley. (e,f) show X and Y components of the flow $\vec{u}_{23}$ between F_2 and F_3. White regions denote occlusions. (g) shows inverse depth from motion. (h) shows 3D structure from motion. (p,q) show rectified stereo pair of images. (q) is the same as (b). (r) shows inverse depth from stereo. (s) shows 3D structure from stereo. Compare (s) with (h) to see how the background objects appear closer to the leopard in (s) than in (h). (x) shows the histogram of depth ratios and clusters detected by k-means ($k = 3$). (y) shows cluster labels: cluster 2 (yellow) is the background, cluster 3 (red) is the leopard, cluster 1 (light blue) is mostly due to errors in the disparity and flow. (z) shows the moving objects of *Class 3* (clusters other than 2)

VII. Conclusion

This paper classifies moving objects into three classes, and discusses constraints for detecting each class of objects: *Class 1* is detected using motion alone, *Class 2* is detected using conflicts between ordinal depth from occlusions and depth from motion, while *Class 3* requires cardinal comparisons between depth from motion and depth from another source. The key contribution of this paper is the detection of *Class* 2 moving objects using the ordinal depth conflict. In this regard, we have presented a novel method using three frames for performing occlusion filling in the presence of moving objects to deduce ordinal depth relations. This tool is of general use

in video processing and can be used in a variety of applications including video compression.

VIII. Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive suggestions for improving this manuscript. The support of the National Science Foundation and ARDA is gratefully acknowledged.

References

[1] M. Bober and J. Kittler, "Robust motion analysis," in *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, 1994, pp. 947–952.

[2] P. J. Burt, J. R. Bergen, R. Hingorani, R. Kolczynski, W. A. Lee, A. Leung, J. Lubin, and H. Shvaytser, "Object tracking with a moving camera," in *Proc. IEEE Workshop on Visual Motion*, 1989, pp. 2–12.

[3] J.-M. Odobez and P. Bouthemy, "MRF-based motion segmentation exploiting a 2D motion model and robust estimation," in *Proc. Int'l Conf. Image Processing*, vol. III, 1995, pp. 628–631.

[4] Y. Weiss, "Smoothness in layers: Motion segmentation using nonparametric mixture estimation." in *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, 1997, pp. 520–526.

[5] G. Adiv, "Determining 3D motion and structure from optical flow generated by several moving objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 7, pp. 384–401, 1985.

[6] Z. Zhang, O. D. Faugeras, and N. Ayache, "Analysis of a sequence of stereo scenes containing multiple moving objects using rigidity constraints," in *Proc. Second Int'l Conf. Comp. Vision*, 1988, pp. 177–186.

[7] W. B. Thompson and T.-C. Pong, "Detecting moving objects," *Int'l J. of Comp. Vision*, vol. 4, pp. 39–57, 1990.

[8] R. C. Nelson, "Qualitative detection of motion by a moving observer," *Int'l J. of Comp. Vision*, vol. 7, pp. 33–46, 1991.

[9] D. Sinclair, "Motion segmentation and local structure," in *Proc. Fourth Int'l Conf. Comp. Vision*, 1993, pp. 366–373.

[10] P. H. S. Torr and D. W. Murray, “Stochastic motion clustering,” in *Proc. Third European Conf. Comp. Vision*. Springer, 1994, pp. 328–337.

[11] J. Costeira and T. Kanade, “A multi-body factorization method for motion analysis,” in *Proc. Int’l Conf. Comp. Vision*, 1995, pp. 1071–1076.

[12] J. Weber and J. Malik, “Rigid body segmentation and shape description from dense optical flow under weak perspective,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 2, 1997.

[13] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, “Bundle adjustment - a modern synthesis,” in *Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer, 2000.

[14] S. Ayer, P. Schroeter, and J. Bigün, “Segmentation of moving objects by robust motion parameter estimation over multiple frames,” in *Proc. Third European Conf. Comp. Vision*. Springer, 1994, pp. 316–327.

[15] C. S. Wiles and M. Brady, “Closing the loop on multiple motions,” in *Proc. Fifth Int’l Conf. Comp. Vision*, 1995, pp. 308–313.

[16] Q. F. Zheng and R. Chellappa, “Motion detection in image sequences acquired from a moving platform,” in *Proc. IEEE Int’l Conf. Acoustics, Speech, and Signal Processing*, 1993, pp. 201–204.

[17] P. H. S. Torr, “Geometric motion segmentation and model selection,” in *Philosophical Transactions of the Royal Society A*, J. Lasenby, A. Zisserman, R. Cipolla, and H. Longuet-Higgins, Eds., 1998, pp. 1321–1340.

[18] M. Irani and P. Anandan, “A unified approach to moving object detection in 2D and 3D scenes,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 577–589, 1998.

[19] H. S. Sawhney, Y. Guo, and R. Kumar, “Independent motion detection in 3D scenes,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 1191–1199, 2000.

[20] G. Adiv, “Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 477–489, 1989.

[21] K. Daniilidis and M. E. Spetsakis, “Understanding noise sensitivity in structure from

motion," in *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, ser. Advances in Comp. Vision, Y. Aloimonos, Ed. Lawrence Erlbaum Associates, 1997, ch. 4.

[22] S. J. Maybank, "A theoretical study of optical flow," Ph.D. dissertation, University of London, 1987.

[23] D. J. Heeger and A. D. Jepson, "Subspace methods for recovering rigid motion I: Algorithm and implementation," *Int'l J. of Comp. Vision*, vol. 7, pp. 95–117, 1992.

[24] C. Fermüller and Y. Aloimonos, "Observability of 3D motion," *Int'l J. of Comp. Vision*, vol. 37, pp. 43–63, 2000.

[25] T. Brodský, C. Fermüller, and Y. Aloimonos, "Structure from motion: beyond the epipolar constraint," *Int'l J. of Comp. Vision*, vol. 37, pp. 231–258, 2000.

[26] T. Darrell and D. Fleet, "Second-order method for occlusion relationships in motion layers," MIT Media Lab, Tech. Rep. 314, 1995.

[27] L. Bergen and F. Meyer, "Motion segmenation and depth ordering based on morphological segmentation," in *European Conf. Comp. Vision*, 1998, pp. 531–547.

[28] D. Tweed and A. Calway, "Integrated segmentation and depth ordering of motion layers in image sequences," in *British Machine Vision Conf.*, 2000.

[29] B. S. Reddy and B. N. Chatterji, "An FFT-based technique for translation, rotation and scale-invariant image registration," *IEEE Trans. Image Processing*, vol. 5, no. 8, pp. 1266–1271, August 1996.

[30] D. J. Fleet, "Disparity from local weighted phase-correlation," *Int'l Conf. Systems, Man and Cybernetics*, pp. 48–56, October 1994.

[31] A. S. Ogale, "The compositional character of visual correspondence," Ph.D. dissertation, University of Maryland, College Park, August 2004.

[32] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Proc. Int'l Conf. Comp. Vision*, vol. 2, 2001, pp. 508–515.

[33] C. Silva and J. Santos-Victor, "Motion from occlusions," *Robotics and Autonomous Systems*, vol. 35, no. 3-4, pp. 153–162, June 2001.