1

Detecting Independent 3D Movement

Abhijit S. Ogale, Cornelia Fermüller, Yiannis Aloimonos

Center for Automation Research, University of Maryland, College Park,
MD 20742, USA
Email: ogale,fer,yiannis@cfar.umd.edu

1.1 Introduction

Motion segmentation is the problem of finding parts of the scene which possess independent 3D motion (such as people, animals or other objects like vehicles). This process is conceptually straightforward if the camera is stationary, and is often referred to as background subtraction. However, if the camera itself is also moving, then the problem becomes more complicated, since the image motion is generated by the combined effects of camera motion, structure and the motion of the independently moving objects. Isolating the contribution of each of these three factors is needed to solve the more general independent motion problem, which involves motion segmentation (finding the moving objects) and also finding their 3D motion. In this chapter, we shall restrict ourselves to the problem of finding moving objects only and not worry about finding their 3D motion. In the beginning, we present our philosophy that visual problems such as motion segmentation are inextricably linked with other problems in vision, and must be approached with a compositional outlook which attempts to solve multiple problems simultaneously. This is followed by a brief review of existing algorithms which detect independently moving objects. The main body of this chapter presents our approach to motion segmentation[1] which classifies moving objects and demonstrates that motion segmentation is compositional and is not about motion alone, but can also utilize information from sources such as occlusions to detect a wider array of moving objects.

[1] This chapter is based on our paper which is due to appear in the IEEE Transactions on Pattern Analysis and Machine Intelligence [1]; portions from [1] have been reprinted with permission (© 2004 IEEE). The support of the National Science Foundation is gratefully acknowledged.

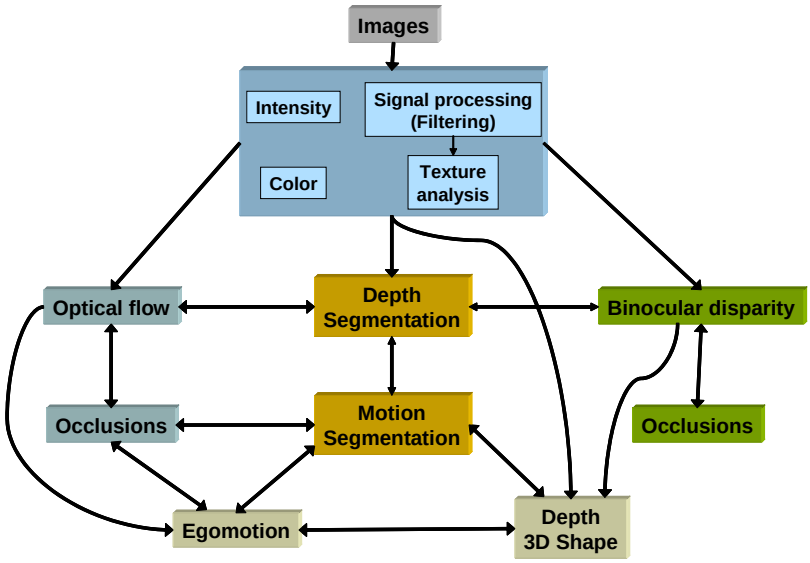

Fig. 1.1. Compositional problems

1.2 A Compositional Viewpoint

Finding the independently moving objects in an image sequence involves solving a host of related problems. In Fig. 1.1, we attempt to give our viewpoint about the relationships between the motion segmentation problem and other problems. At the beginning, the image stream is described in terms of intensity and/or color, and may be subjected to signal processing operations using various localized filters which describe features such as edges or the components of textures using spatial frequency channels. These local measurements are aggregated within a global framework to compute quantities such as optical flow (2D motion) and occlusions. Occlusions are parts of an image frame which disappear in the next frame. If binocular input is present, then disparity measurements and binocular occlusion information can also be computed. Detection of depth edges is affected by evidence from optical flow, binocular disparity and also monocular image measurements such as intensity and texture; in turn, these edges influence the estimation of each of these quantities.

Optical flow is used to compute egomotion (the motion of the camera), detect independently moving objects, and recover the background depth map concurrently. As we shall see, this process is often performed by finding clusters with consistent 3D motion. Later, we also show that occlusions provide information about ordinal scene structure which can be used to find new types of moving objects. The depth map of the scene is influenced by structure estimates from motion, binocular disparity measurements (if present), and

influences from single image measurements such as intensity (i.e., via shape from shading) and texture (i.e., via shape from texture). Overall, the problem of motion segmentation requires a compositional solution which utilizes the relationships between different modules to obtain better solutions.

1.3 Existing Approaches

Prior research can mostly be classified into two groups: (a) The approaches relying, prior to 3D motion estimation, on 2D motion field measurements only [2, 3, 4, 5]. 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 involving 3D motion estimation. (b) Approaches which assume that partial or full information about egomotion is available or can be recovered. Adiv [6] first segments on the basis of optical flow, and then groups the segments by searching for agreeable 3D motion parameters. Zhang et al. [7] utilize rigidity constraints on a sequence of stereo images to find egomotion and moving objects. Thompson and Pong's [8] first method finds inconsistencies between the egomotion and the flow field by using the motion epipolar constraint, while the second method relies on external depth information. Nelson [9] discusses two approaches, the first of which is similar to Thompson and Pong, while the second relies on acceleration detection. Sinclair [10] uses the angular velocity field and the premise that independently moving objects violate the epipolar constraint. Torr and Murray [11] find a set of fundamental matrices to describe the observed correspondences by hypothesizing clusters using robust statistics. Costeira and Kanade [12] use the factorization method along with a feature grouping step (block diagonalization of the shape interaction matrix).

Some techniques, such as [13], which address both 3D motion estimation and moving object detection, are based on alternate models of image formation, such as weak perspective. Such additional constraints can be justified for domains such as aerial imagery. In this case, the planarity of the scene allows a registration process [14, 15, 16, 17], 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 [18], or by fitting multiple planes using the plane plus parallax constraint [19, 20]. The former [19] uses the best of 2D and 3D techniques, progressively increasing the complexity based on the situation. The latter [20] also develops constraints on the structure using three frames. 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. We therefore restrict ourselves to detecting moving objects using two or three frames only.

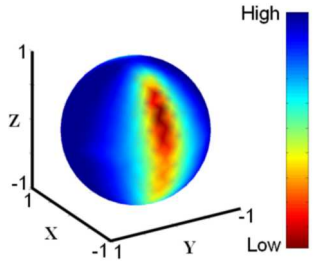

Fig. 1.2. Motion valley (red) visualized as an error surface in the 2D space of directions of translation, represented by the surface of a sphere. The error is found after finding the optimal rotation and structure for each translation direction. (Reproduced from [1] with permission © 2004 IEEE)

1.4 Ambiguity in 3D Motion Estimation

Many techniques detect independently moving objects based on 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 [21, 22]. This can be intuitively understood by examining the differential flow equation:

$$\begin{aligned} u &= \frac{-t_x f + x t_z}{Z} + \alpha \frac{xy}{f} - \beta \left(\frac{x^2}{f} + f \right) + \gamma y, \\ v &= \frac{-t_y f + y t_z}{Z} + \alpha \left(\frac{y^2}{f} + f \right) - \beta \frac{xy}{f} - \gamma x \end{aligned} \tag{1.1}$$

In the above equation, (u, v) is the optical flow, (t_x, t_y, t_z) is the translation, (α, β, γ) is the rotation and $Z(x, y)$ is the depth map. Notice that for a planar scene, up to zeroth order, we have $u \approx -t_x f/Z - \beta f$ and $v \approx -t_y f/Z + \alpha f$. Intuitively, we see how translation along the x-axis t_x can be confused with rotation β along the y-axis, and t_y with α for a small field of view.

Maybank [23] and Heeger and Jepson [24] 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 [25], an algorithm-independent stability analysis of the structure from motion problem has been carried out.

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. 1.2 shows a typical error function obtained using the motion estimation technique of Brodsky et al. [26] plotted on the 2D spherical surface of translational directions. Motion-based clustering can only succeed if a scene entity has a motion which does not lie in the background motion valley. In the following sections, we go beyond motion-based clustering and present a classification of moving objects with algorithms for detecting each class, laying particular emphasis on the role of occlusions.

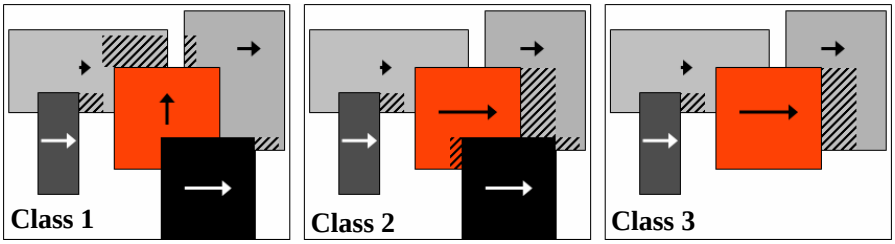

Fig. 1.3. Toy examples of three classes of moving objects. In each case, the independently moving object is red colored. Portions of objects which disappear in the next frame (i.e., occlusions) are shown in a dashed texture. (Reproduced from [1] with permission © 2004 IEEE)

1.5 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. 1.3 shows illustrative examples of the three classes.

1.5.1 Class 1: 3D Motion-Based Clustering

The first column of Fig. 1.3 shows a situation in which the background objects (non-independently moving) are translating horizontally, while the red object is moving vertically. In this scenario, motion-based clustering approaches will be successful, since the motion of the red 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 Sect. 1.7.

1.5.2 Class 2: Ordinal Depth Conflict between Occlusions and Structure from Motion

The second column of Fig. 1.3 shows a situation in which the background objects are translating horizontally to the right, and the red 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 red object is *behind* the black object. However, if we compute structure from motion, since the motion is predominantly a translation, the result would indicate that the red object is in front of the black object (since the red object moves faster). This conflict between ordinal depth from occlusions and structure from motion permits the detection of *Class 2* moving objects. In Sect. 1.8, we present a novel algorithm for finding ordinal depth.

1.5.3 Class 3: Cardinal Depth Conflict

The third column of Fig. 1.3 shows a situation similar to the second column, except that the black object which was in front of the red object has been removed. Due to this situation, the ordinal depth conflict which helped us detect the red 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 ratio greater than the background, and pixels with depth ratio 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., [7]) for performing motion segmentation on stereo images, which can also be used to detect *Class 3* moving objects.

1.6 Phase Correlation

Before we move on to our approach for motion-based clustering, let us explain a simple technique which allows us to recover a four-parameter transformation between a pair of images using phase correlation (see [27]). We use this technique for initializing background motion estimation, and it may also be used for stabilizing a jittery video.

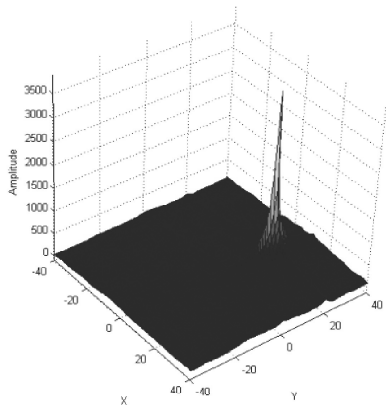

Fig. 1.4. An example of a peak generated by the phase correlation method

1.6.1 Basic Process

Consider an image which is moving in its own plane, i.e., every point on the image has the same flow. Thus, if the image $I_2(x, y)$ is such a translated version of the original image $I_1(x, y)$, then we can use phase correlation to recover the translation in the following manner.

If $I_2(x, y) = I_1(x + t_x, y + t_y)$, then their Fourier transforms are related by:

$$f_2(\omega_x, \omega_y) = f_1(\omega_x, \omega_y) e^{-i(\omega_x t_x + \omega_y t_y)} \tag{1.2}$$

The phase correlation $PC(x, y)$ of the two images is then given by:

$$PC(x, y) = \mathcal{F}^{-1} \left[\frac{f_1^* \cdot f_2}{|f_1^* \cdot f_2|} \right] = \mathcal{F}^{-1} \left[e^{-i(\omega_x t_x + \omega_y t_y)} \right] \tag{1.3}$$

$$PC(x, y) = \delta(x - t_x, y - t_y) \tag{1.4}$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform, * denotes the complex conjugate and δ is the delta function. Thus, if we use phase correlation, we can

recover this global image translation (t_x, t_y) since we get a peak at this position (see example in Fig. 1.4).

1.6.2 Log-Polar Coordinates

The log-polar coordinate system (ρ, α) is related to the cartesian coordinates (x, y) by the following transformation:

$$x = e^{\rho} cos(\alpha) \tag{1.5}$$
$$y = e^{\rho} sin(\alpha) \tag{1.6}$$

If the image is transformed into the log-polar coordinate system (see Fig. 1.5), then changes in scale and rotation about the image center in the cartesian coordinates are transformed into translations in the log-polar coordinates. Hence, if we perform the phase correlation procedure mentioned above in the log-polar domain, we can also recover the scale change s, and a rotation about the center γ, between two images.

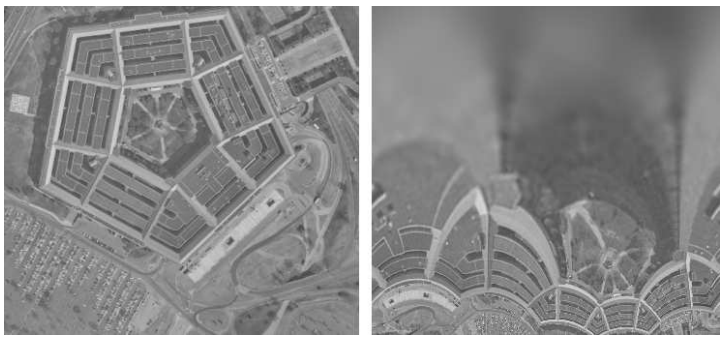

Fig. 1.5. An image in cartesian coordinates (left) and its log-polar representation (right)

1.6.3 Four-Parameter Estimation

Given two images which are related by 2D translation, rotation about the center and scaling, we can perform phase correlation in both the cartesian and log-polar domains to compute a four-parameter transformation T between the two images:

$$T = \begin{bmatrix} s \cdot cos(\gamma) & s \cdot sin(\gamma) & t_x \\ -s \cdot sin(\gamma) & s \cdot cos(\gamma) & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1.7}$$

In practice, initializing this process is tricky, since dominant 2D translation will cause problems in the log-polar phase correlation by introducing many large additional peaks, and, similarly, dominant scaling and rotation will cause problems in the cartesian phase correlation.

To address this, we first perform phase correlation in both the cartesian and log-polar representations on the original images. Then, for each of the results, we find the ratio of the magnitude of the tallest peak to the overall median peak amplitude. If this ratio is greater for the cartesian computation, it means that translation is dominant over scaling and rotation, and must be removed first. Then we can estimate scaling and rotation again on the corrected images. Similarly, if the ratio is greater for the log-polar computation, we perform the correction the other way around. This process can be iterated a few times until the transformations converge.

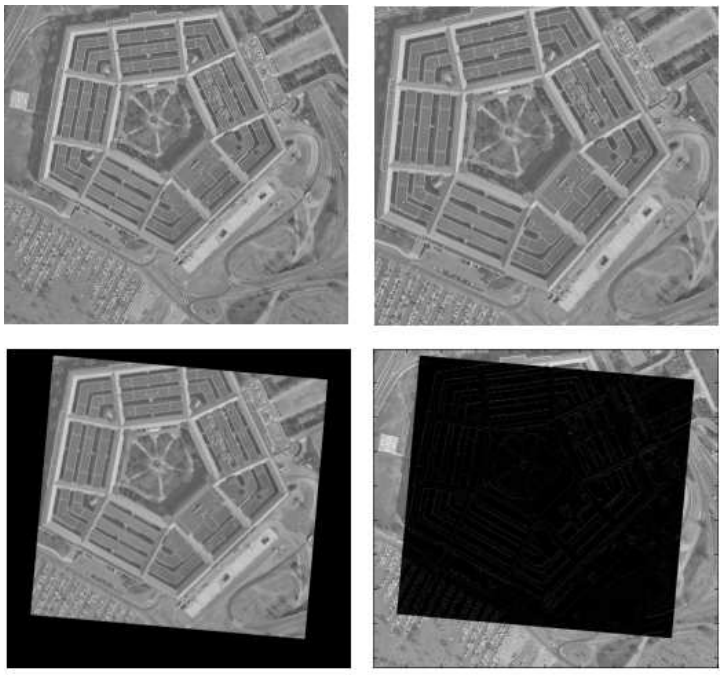

Fig. 1.6. Top row shows two input images I_1 and I_2. Image I_2 was created from I_1 by rotating by 5 degrees, scaling by a factor of 1.2, and translating by $(-10, 20)$ pixels. Bottom row: The left image shows image I_2' obtained by unwarping I_2 using the results of the phase correlation. The right-hand side shows the absolute intensity difference between I_1 and the unwarped image I_2' to reveal the accuracy of the registration. Notice that the difference is nearly zero in the area of overlap

1.6.4 Results

Fig. 1.6 shows the results on a pair of test images which are related by significantly large values of translation, rotation and scaling. These results can be improved to subpixel accuracy by using the method of Foroosh et al. [28]. We have applied the method mentioned above to video sequences and have achieved good stabilization over long durations, even in the presence of independently moving objects.

1.7 Motion-Based Clustering

Motion-based clustering is in itself a difficult problem, since the process of finding the background motion and finding the independently moving clusters has to be performed concurrently. The problem is a chicken-and-egg problem: if we knew the background pixels, we could find the background motion, and vice versa. In Sect. 1.3, 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.

The first step consists of using phase correlation on two frames in the cartesian representation (to find 2D translation t_x, t_y), and in the log-polar representation (to find scale S and z-rotation γ); we obtain a four-parameter transformation between frames (see the previous section). Phase correlation can be thought of as a voting approach [29], 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 $\eta_1 = 45°$) .

In the second step, 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. [26]. 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 $\eta_2 = 45°$) .

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., [30, 31]; 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.

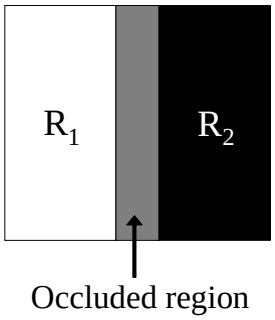

Fig. 1.7. If the occluded region belongs to R_1, then R_1 is behind R_2, and vice versa. (Reproduced from [1] with permission © 2004 IEEE)

1.8 Ordinal Depth from Occlusion Filling Using Three Frames

1.8.1 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 deduce ordinal depth. In Fig. 1.7, 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.

1.8.2 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 [32], knowing the FOE is enough. In the simplest case, shown in Fig. 1.8a, 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. 1.8b), 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. 1.8c), then fill in with the nearest pixel Q on line L, such that Q lies between O and the FOC.

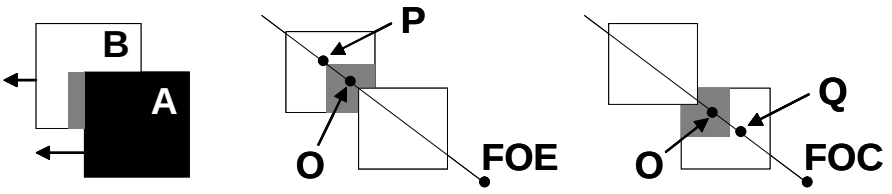

Fig. 1.8. Occlusion filling: from left (a) to (c). Gray regions indicate occlusions (portions which disappear in the next frame) (Reproduced from [1] with permission)

1.8.3 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. Using such measures of image similarity creates many failure modes. We present below a novel and robust strategy utilizing optical flow alone (see Fig. 1.9) for filling occlusions in the general case using three frames instead of two.

Given three consecutive frames F_1, F_2, F_3 of a video sequence, we use an optical flow algorithm which finds dense flow and occlusions (e.g., [30, 31]; we use the former) to compute the following:

1. Using F_1 and F_2, we find flow $\mathbf{u}_{12}$ from frame F_1 to F_2, and the reverse flow $\mathbf{u}_{21}$ 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 $\mathbf{u}_{23}$ and $\mathbf{u}_{32}$, 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*

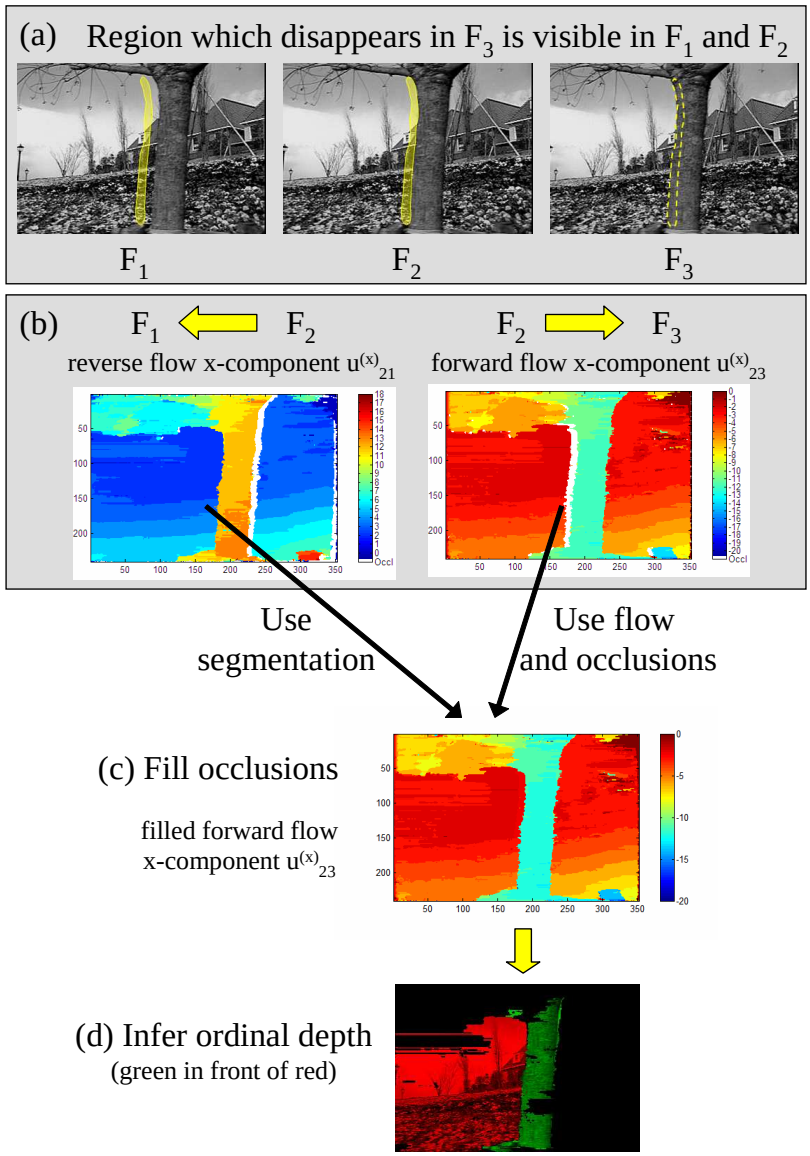

Fig. 1.9. Generalized occlusion filling and ordinal depth estimation. (a) Three frames of a video sequence. The yellow region which is visible in F_1 and F_2 disappears behind the tree in F_3. (b) Forward and reverse flow (only the x-components are shown). Occlusions are colored white. (c) Occlusions in $\mathbf{u}_{23}$ are filled using the segmentation of $\mathbf{u}_{21}$. Note that the white areas have disappeared. (d) Deduce ordinal depth relation. In a similar manner, we can also fill occlusions in $\mathbf{u}_{21}$ using the segmentation of $\mathbf{u}_{23}$ to deduce ordinal depth relations for the right side of the tree. (Reproduced from [1] with permission © 2004 IEEE)

no correspondence in F_3. However, these areas were visible in both F_1 and F_2; hence in $\mathbf{u}_{21}$ these areas have already been grouped with their neighboring regions. Therefore, we can use the segmentation of flow $\mathbf{u}_{21}$ to fill the occluded areas O_{23} in the flow field $\mathbf{u}_{23}$. Similarly, we can use the segmentation of $\mathbf{u}_{23}$ to fill the occluded areas O_{21} in the flow field $\mathbf{u}_{21}$. 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 the find ordinal depth in a robust fashion.

1.9 Algorithm summary

1. Input video sequence: $V = (F_1, F_2, ..., F_n)$
2. For each $F_i \in V$ do
   a) find forward $\mathbf{u}_{i,i+1}$ and reverse $\mathbf{u}_{i,i-1}$ 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

1.10 Experiments

Fig. 1.10 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. 1.11 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. 1.12 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 in Fig. 1.11, with 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.

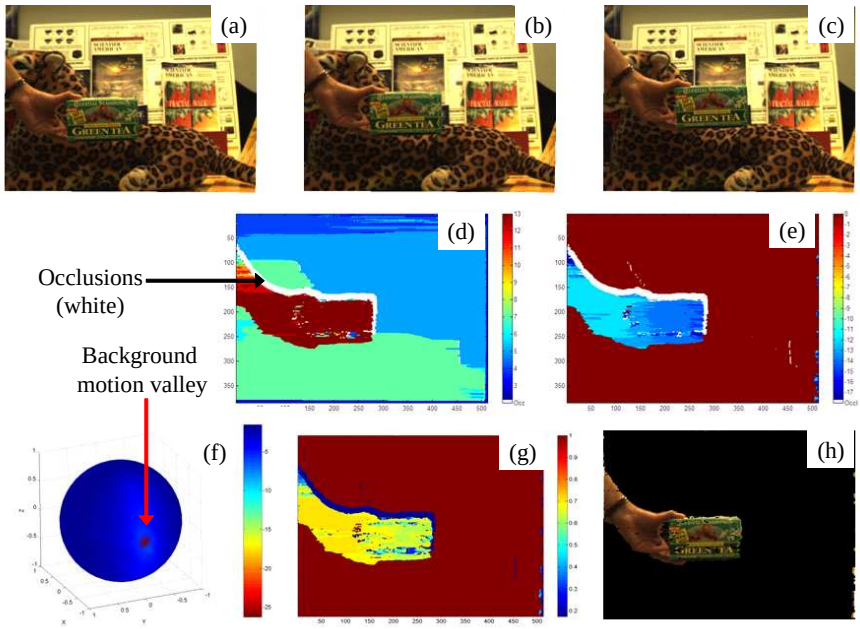

Fig. 1.10. Class 1: (a,b,c) show three frames of the *teabox* sequence. (d,e) show X and Y components of the optical flow using frames (b) and (c). Occlusions are colored white. (f) shows the computed motion valley for the background. (g) shows the cosine of the angular error between the reprojected flow (using the background motion) and the true flow. (h) shows the detected *Class 1* moving object (Reproduced from [1] with permission © 2004 IEEE)

1.11 Conclusion

In this chapter, we have discussed the motion segmentation problem within a compositional framework. Moving objects were classified into three classes,

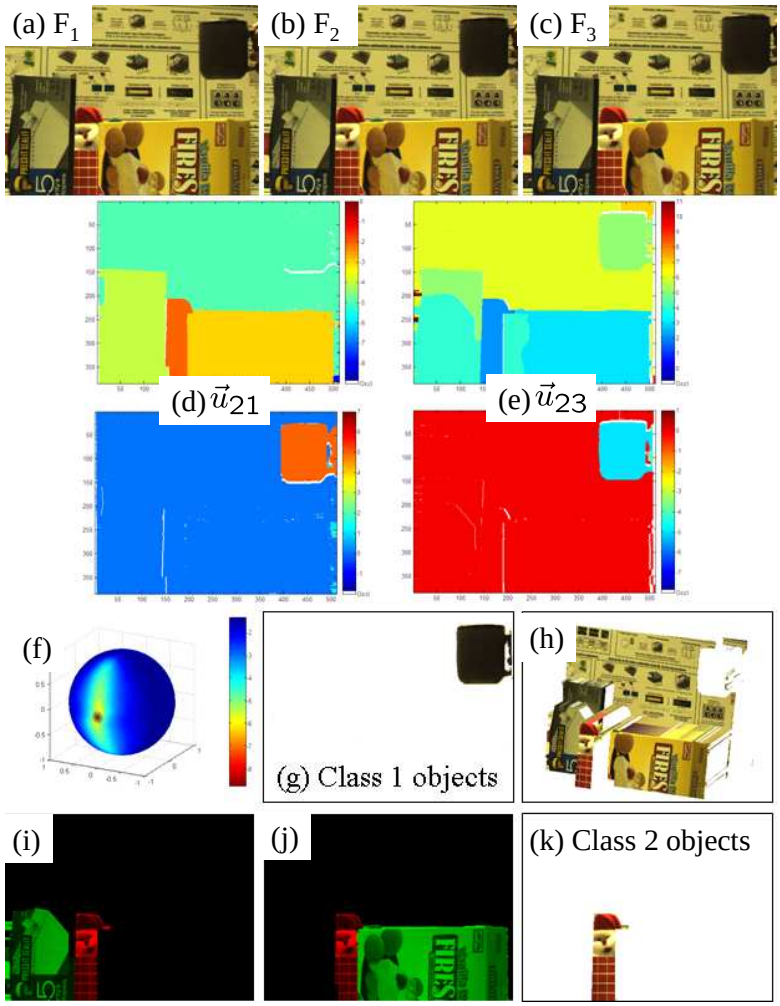

Fig. 1.11. (a,b,c) show three frames F_1, F_2, F_3 of the *santa-coffee* sequence. The camera translates horizontally to the left, hence the scene moves to the right. The *coffee mug* is lifted up, and the red toy *santa* is pulled by a person (not seen) to the right. (d) and (e) show optical flow $\mathbf{u}_{21}$ from frame F_2 to F_1, and $\mathbf{u}_{23}$ 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 (Reproduced from [1] with permission © 2004 IEEE)

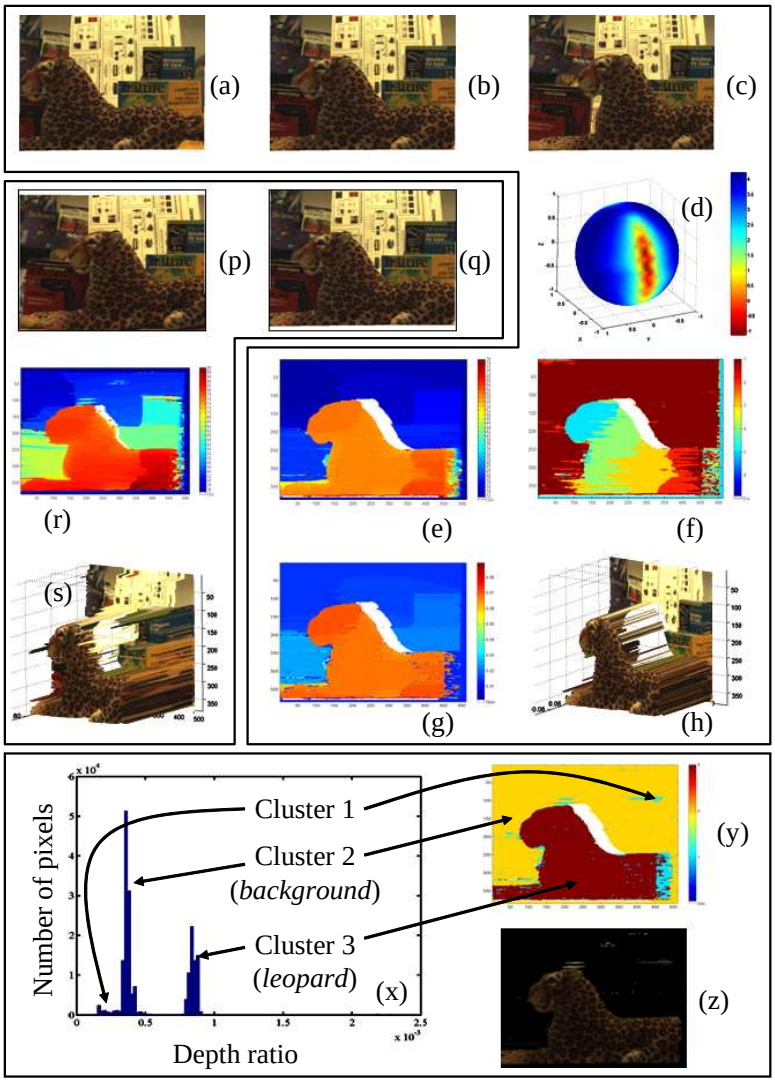

Fig. 1.12. Class 3: (a,b,c) show three frames F_1, F_2, F_3 of the *leopardB* sequence. (d) shows the computed motion valley. (e,f) show X and Y components of the flow $\mathbf{u}_{23}$ between F_2 and F_3. White regions denote occlusions. (g) shows inverse depth from motion. (h) shows 3D structure from 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) (Reproduced from [1] with permission © 2004 IEEE)

and constraints for detecting each class of objects were presented: *Class 1* was detected using motion alone, *Class 2* was detected using conflicts between ordinal depth from occlusions and depth from motion, while *Class 3* required cardinal comparisons between depth from motion and depth from another source.

References

1. A.S. Ogale, C. Fermüller and Y. Aloimonos, Motion segmentation using occlusions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, in press.
2. M. Bober and J. Kittler, Robust motion analysis, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 947–952, 1994.
3. 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*, 2–12, 1989.
4. J.-M. Odobez and P. Bouthemy, MRF-based motion segmentation exploiting a 2D motion model and robust estimation, in *Proc. International Conference on Image Processing*, III:628–631, 1995.
5. Y. Weiss, Smoothness in layers: Motion segmentation using nonparametric mixture estimation, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 520–526, 1997.
6. G. Adiv, Determining 3D motion and structure from optical flow generated by several moving objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:384–401, 1985.
7. 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 International Conference on Computer Vision*, 177–186, 1988.
8. W.B. Thompson and T.-C. Pong, Detecting moving objects, *International Journal of Computer Vision*, 4:39–57, 1990.
9. R.C. Nelson, Qualitative detection of motion by a moving observer, *International Journal of Computer Vision*, 7:33–46, 1991.
10. D. Sinclair, Motion segmentation and local structure, in *Proc. Fourth International Conference on Computer Vision*, 366–373, 1993.
11. P.H.S. Torr and D.W. Murray, Stochastic motion clustering, in *Proc. Third European Conference on Computer Vision.* Springer, 328–337, 1994.
12. J. Costeira and T. Kanade, A multi-body factorization method for motion analysis, in *Proc. International Conference on Computer Vision*, 1071–1076, 1995.
13. J. Weber and J. Malik, Rigid body segmentation and shape description from dense optical flow under weak perspective, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):139–143, 1997.
14. Q.F. Zheng and R. Chellappa, Motion detection in image sequences acquired from a moving platform, in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 201–204, 1993.
15. S. Ayer, P. Schroeter, and J. Bigün, Segmentation of moving objects by robust motion parameter estimation over multiple frames, in *Proc. Third European Conference on Computer Vision.* Springer, 316–327, 1994

16. C.S. Wiles and M. Brady, Closing the loop on multiple motions, in *Proc. Fifth International Conference on Computer Vision*, 308–313, 1995.
17. 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.
18. 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., 1321–1340, 1998.
19. M. Irani and P. Anandan, A unified approach to moving object detection in 2D and 3D scenes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:577–589, 1998.
20. H. Sawhney, Y. Guo, and R. Kumar, "Independent motion detection in 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1191–1199, 2000.
21. G. Adiv, Inherent ambiguities in recovering 3D motion and structure from a noisy flow field, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:477–489, 1989.
22. K. Daniilidis and M.E. Spetsakis, Understanding noise sensitivity in structure from motion, in *Visual Navigation: from Biological Systems to Unmanned Ground Vehicles*, Series on Advances in Computer Vision, Y. Aloimonos, Ed., Lawrence Erlbaum Associates, ch. 4, 1997.
23. S.J. Maybank, A theoretical study of optical flow, Ph.D. dissertation, University of London, 1987.
24. D.J. Heeger and A.D. Jepson, Subspace methods for recovering rigid motion I: Algorithm and implementation, *International Journal of Computer Vision*, 7:95–117, 1992.
25. C. Fermüller and Y. Aloimonos, Observability of 3D motion, *International Journal of Computer Vision*, 37:43–63, 2000.
26. T. Brodský, C. Fermüller, and Y. Aloimonos, Structure from motion: beyond the epipolar constraint, *International Journal of Computer Vision*, 37:231–258, 2000.
27. B.S. Reddy and B. Chatterji, "An FFT-based technique for translation, rotation and scale-invariant image registration," *IEEE Transactions on Image Processing*, 5(8):1266–1271, August 1996.
28. H. Foroosh, J. Zerubia, and M. Berthod, Extension of phase correlation to subpixel registration, *IEEE Transactions on Image Processing*, 11(3):188–200, March 2002.
29. D. Fleet, Disparity from local weighted phase-correlation, *IEEE International Conference on SMC*, 48–56, October 1994.
30. A.S. Ogale, The compositional character of visual correspondence, Ph.D. dissertation, University of Maryland, College Park, USA, *www.cfar.umd.edu/users/ogale/thesis/thesis.html*, August 2004.
31. V. Kolmogorov and R. Zabih, Computing visual correspondence with occlusions using graph cuts, in *Proc. International Conference on Computer Vision*, 2:508–515, 2001.
32. C. Silva and J. Santos-Victor, Motion from occlusions, *Robotics and Autonomous Systems*, 35(3–4):153–162, June 2001.