2.1 Depth Estimation of Frames in Image Sequences Using Motion Occlusions

Guillem Palou and Philippe Salembier

Technical University of Catalonia (UPC),
Dept. of Signal Theory and Communications, Barcelona, Spain
{guillem.palou,philippe.salembier}@upc.edu

Abstract. This paper proposes a system to depth order regions of a frame belonging to a monocular image sequence. For a given frame, regions are ordered according to their relative depth using the previous and following frames. The algorithm estimates occluded and disoccluded pixels belonging to the central frame. Afterwards, a Binary Partition Tree (BPT) is constructed to obtain a hierarchical, region based representation of the image. The final depth partition is obtained by means of energy minimization on the BPT. To achieve a global depth ordering from local occlusion cues, a depth order graph is constructed and used to eliminate contradictory local cues. Results of the system are evaluated and compared with state of the art figure/ground labeling systems on several datasets, showing promising results.

1 Introduction

Depth perception in human vision emerges from several depth cues. Normally, humans estimate depth accurately making use of both eyes, inferring (subconsciously) disparity between two views. However, when only one point of view is available, it is also possible to estimate the scene structure to some extent. This is done by the so called monocular depth cues. In static images, T-junctions or convexity cues may be detected in specific image areas and provide depth order information. If a temporal dimension is introduced, motion information can also be used to get depth information. Occlusion of moving objects, size changes or motion parallax are used in the human brain to structure the scene [1].

Nowadays, a strong research activity is focusing on depth maps generation, mainly motivated by the film industry. However, most of the published approaches make use of two (or more) points of view to compute the disparity as it offers a reliable cue for depth estimation [2]. Disparity needs at least two images captured at the same time instant but, sometimes, this requirement cannot be fulfilled. For example, current handheld cameras have only one objective. Moreover, a large amount of material has already been acquired as monocular sequences and needs to be converted. In such cases, depth perception should be inferred only through monocular cues. Although monocular cues are less reliable than stereo cues, humans can do this task with ease.

A. Fusiello et al. (Eds.): ECCV 2012 Ws/Demos, Part III, LNCS 7585, pp. 516–525, 2012.

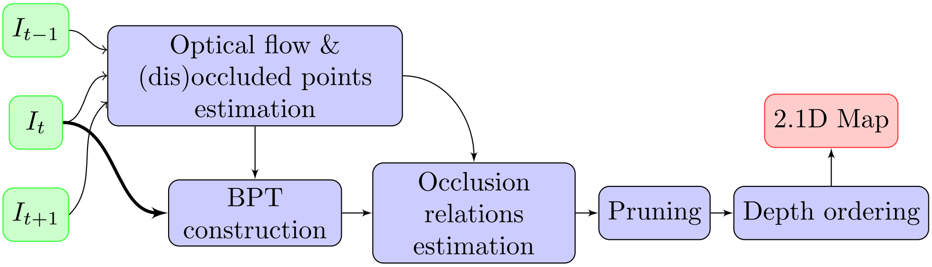
I_{t-1}
I_t
I_{t+1}
Optical flow & (dis)occluded points estimation
BPT construction
Occlusion relations estimation
Pruning
Depth ordering
2.1D Map

Fig. 1. Scheme of the proposed system. From three consecutive frames of a sequence (green blocks), a 2.1D map is estimated (red block)

The 2.1D model is an intermediate state between 2D images and full/absolute 3D maps, representing the image as a partition with its regions ordered by its relative depth. State of the art depth ordering systems on monocular sequences focus on the extraction of foreground regions from the background. Although this may be appropriate for some applications, more information can be extracted from an image sequence. The approach in [3] provides a pseudo-depth estimation to detect occlusion boundaries from optical flow. References [4,5] estimate a layered image representation of the scene. Whereas, references [6,7] attempt to retrieve a full depth map from a monocular image sequence, under some assumptions/restrictions about the scene structure which may not be fulfilled in typical sequences. The work [8] assigns figure/ground (f/g) labels to detected occlusion boundaries. f/g labeling provides a quantitative measure of depth ordering, as it assigns a local depth gradient at each occlusion boundary. Although f/g labeling is an interesting field of study, it does not offer a dense depth representation.

A good monocular cue to determine a 2.1D map of the scene is motion occlusion. When objects move, background regions (dis)appear, creating occlusions. Humans use these occlusions to detect the relative depth between scene regions. The proposed work assesses the performance of these cues in a fully automated system. To this end, the process is divided as shown in Figure 1 and presented as follows. First, the optical flow is used in Section 2 to introduce motion information for the BPT [9] construction and in Section 3 to estimate (dis)occluded points. Next, to find both occlusion relations and a partition of the current frame, the energy minimization technique described in Section 4 is used. Lastly, the regions of this partition are ordered, generating a 2.1D map. Results compared with [8] are exposed in Section 5.

2 Optical Flow and Image Representation

As shown in Figure 2, to determine the depth order of frame I_t, the previous I_{t-1} and following I_{t+1} frames are used. Forward $\boldsymbol{w}^{t-1,t}$, $\boldsymbol{w}^{t,t+1}$ and backward flows $\boldsymbol{w}^{t,t-1}$, $\boldsymbol{w}^{t+1,t}$ can be estimated using [10]. For two given temporal indices a, b, the optical flow vector $\boldsymbol{w}^{a,b}$ maps each pixel of I_a to one pixel in I_b.

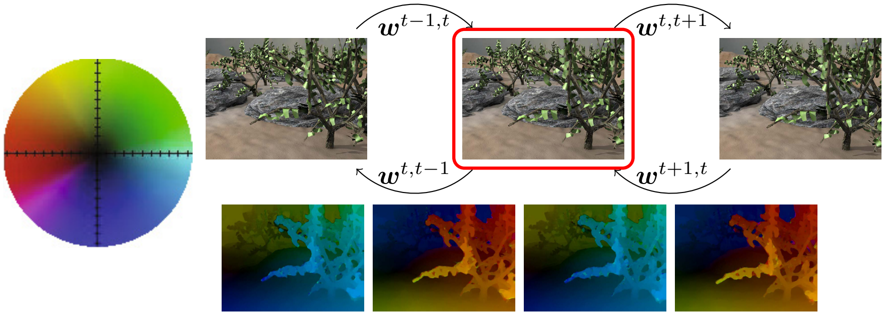

Fig. 2. Left: color code used to represent optical flow values. Three consecutive frames are presented in the top row, I_{t-1} ,I_t in red and I_{t+1}. In the bottom row, from left to right, the $\boldsymbol{w}^{t-1,t}$,$\boldsymbol{w}^{t,t-1}$,$\boldsymbol{w}^{t,t+1}$,$\boldsymbol{w}^{t+1,t}$ flows are shown.

Once the optical flows are computed, a BPT is built [11]. The BPT begins with an initial partition (here a partition where each pixel forms a region). Iteratively, the two most similar neighboring regions according to a predefined distance are merged and the process is repeated until only one region is left. The BPT describes a set of regions organized in a tree structure and this hierarchical structure represents the inclusion relationship between regions. Although the construction process is an active field of study, it is not the main purpose of this paper and we chose the distance defined in [11] to build the BPT: the region distance is defined using color, area, shape and motion information.

3 Motion Occlusions from Optical Flow

When only one point of view is available, humans take profit of monocular depth cues to retrieve the scene structure: motion parallax and motion occlusions. Motion parallax assumes still scenes, and it is able to retrieve the absolute depth. Occlusions may work in dynamic scenes but only offer insights about relative depth. Since motion occlusions appear in more situations and do not make any assumptions, they are selected here. Motion occlusions can be detected with several approaches [12,13]. In this work, however, a different approach is followed as it gave better results in practice.

Using three frames I_{t-1}, I_t, I_{t+1}, it is possible to detect pixels becoming occluded from I_t to I_{t+1} and pixels becoming visible (disoccluded) from I_{t-1} to I_t. To detect motion occlusions, the optical flow between an image pair (I_t, I_q) is used with $q = t \pm 1$. To obtain occluded pixels $q = t + 1$, while disoccluded are obtained when $q = t - 1$.

Flow estimation attempts to find a matching for each pixel between two frames. If a pixel is visible in both frames, the flow estimation is likely to find the true matching. If, however, the pixel becomes (dis)occluded, the matching will not be against its true peer. In the case of occlusion, two pixels p_a and p_b in I_t will be matched with the same pixel p_m in frame I_q:

$$\boldsymbol{p}_a + \boldsymbol{w}^{t,q}(\boldsymbol{p}_a) = \boldsymbol{p}_b + \boldsymbol{w}^{t,q}(\boldsymbol{p}_b) = \boldsymbol{p}_m \tag{1}$$

Equation (1) implicitly tells that either $\boldsymbol{p}_a$ or $\boldsymbol{p}_b$ is occluded. It is likely that the non occluded pixel neighborhood is highly correlated in both frames. Therefore, to decide which one is the occluded pixel, a patch distance is computed:

$$D(\boldsymbol{p}_x, \boldsymbol{p}_m) = \sum_{\boldsymbol{d} \in \Gamma} \left(I_q(\boldsymbol{p}_m + \boldsymbol{d}) - I_t(\boldsymbol{p}_x + \boldsymbol{d}) \right)^2 \tag{2}$$

with $\boldsymbol{p}_x = \boldsymbol{p}_a$ or $\boldsymbol{p}_b$. The pixel with maximum $D(p_x, p_m)$ value is decided to be the occluded pixel. The neighborhood Γ is a 5×5 square window centered at $\boldsymbol{p}_x$ but results are similar with windows of size 3×3 or 7×7.

Occluded and disoccluded pixels may be useful to some extent (e.g. to improve optical flow estimation, [12]). To retrieve a 2.1D map, an (dis)occluded-(dis)occluding relation is needed to create a depth order. (Dis)occluding pixels are pixels in I_t that will be in front of their (dis)occluded peer in I_q. Therefore, using these relations it is possible to order different regions in the frame according to depth. In the proposed system, occlusion relations estimation is postponed until the BPT representation is available, see Section 4.1. The reason to do so is because raw estimated optical flows are not reliable in occluded points. Nevertheless, with the knowledge of region information it is possible to fit optical flow models to regions and provide confident optical flow values even for (dis)occluded points.

4 Depth Order Retrieval

Once the optical flow is estimated and the BPT is constructed, the last step of the system is to retrieve a suitable partition to depth order its regions. There are many ways to obtain a partition from a hierarchical representation [14,15,9]. In this work an energy minimization strategy is proposed. The complete process comprises two energy minimization steps to find the final partition. Since raw optical flows are not reliable at (dis)occluded points, a first step allows us to find a partition P_f where an optical flow model is fitted in each region. When the occlusion relations are estimated, the second step finds a second partition P_d attempting to maintain occluded-occluding pairs in different regions. The final stage of the system relates regions in P_d according to their relative depth.

Obtaining P_f and P_d is performed using the same energy minimization algorithm. For this reason, the general algorithm is presented first in Section 4.1 and then it is particularized for each step in the following subsections.

4.1 General Energy Minimization on BPTs

A partition P, can be represented by a vector $\boldsymbol{x}$ of binary variables $x_i = \{0, 1\}$ with $i = 1..N$, one for each region R_i forming the BPT. If $x_i = 1$, R_i is in the partition, otherwise $x_i = 0$. Although there are a total of 2^N possible vectors,

Algorithm 1. Optimal Partition Selection

```
function OptimalSubTree(Region $R_i$)
    $R_l, R_r \leftarrow$ (LeftChild($R_i$),RightChild($R_i$))
    $(c_i, \boldsymbol{o}_i) \leftarrow (E_r(R_i), R_i)$
    $(\boldsymbol{o}_l, c_l) \leftarrow$ OptimalSubTree($R_l$)
    $(\boldsymbol{o}_r, c_r) \leftarrow$ OptimalSubTree($R_r$)
    if $c_i < c_r + c_l$ then
        OptimalSubTree($R_i$) $\leftarrow (\boldsymbol{o}_i, c_i)$
    else
        OptimalSubTree($R_i$) $\leftarrow (\boldsymbol{o}_l \bigcup \boldsymbol{o}_r, c_l + c_r)$
    end if
end function
```

only a reduced subset may represent a partition, as shown in Figure 3. A given vector $\boldsymbol{x}$ is a valid vector if one, and only one, region in every BPT branch has $x_i = 1$. A branch is the sequence of regions from a leaf to the root of the tree. Intuitively speaking, if a region R_i is forming the partition P ($x_i = 1$), no other region R_j enclosed or enclosing R_i may have $x_j = 1$. This can be expressed as a linear constraint $\boldsymbol{A}$ on the vector $\boldsymbol{x}$. $\boldsymbol{A}$ is provided for the case in Figure 3:

$$\boldsymbol{A}\boldsymbol{x} = \mathbf{1} \qquad \begin{pmatrix} 1\,0\,0\,0\,1\,0\,1 \\ 0\,1\,0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,0\,1\,1 \end{pmatrix} \boldsymbol{x} = \mathbf{1} \tag{3}$$

Where $\mathbf{1}$ is a vector containing all ones. The proposed optimization scheme finds a partition that minimizes energy functions of the type:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} E(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}} \sum_{R_i \in BPT} E_r(R_i) x_i \tag{4}$$

$$s.t. \qquad \boldsymbol{A}\boldsymbol{x} = \mathbf{1} \qquad x_i = \{0, 1\} \tag{5}$$

where $E_r(R_i)$ is a function that depends only of the internal characteristics of the region (mean color or shape, for example). If that is the case, Algorithm 1 uses dynamic programming (Viterbi like) to find the optimal $\boldsymbol{x}^*$.

Fitting the Flows and Finding Occlusion Relations. As stated in Section 3, the algorithm [10] does not provide reliable flow values at (dis)occluded points. Therefore, to be able to determine consistent occlusion relations, the flow in non-occluded areas is extrapolated to these points by finding a partition P_f and estimating a parametric projective model [16] in each region. The set of regions that best fits to these models is computed using Algorithm 1 with $E_r(R_i)$:

$$E_r(R_i) = \sum_{q=t\pm 1} \sum_{x,y \in R_i} \left| \boldsymbol{w}^{t,q}(x,y) - \widetilde{\boldsymbol{w}}^{t,q}_{R_i}(x,y) \right| + \lambda_f \tag{6}$$

Valid $\boldsymbol{x} = (x_1, \ldots, x_7)^T$:

$\boldsymbol{x}_1 = (1,1,1,1,0,0,0)^T$
$\boldsymbol{x}_2 = (0,0,1,1,1,0,0)^T$
$\boldsymbol{x}_3 = (1,1,0,0,0,1,0)^T$
$\boldsymbol{x}_4 = (0,0,0,0,1,1,0)^T$
$\boldsymbol{x}_5 = (0,0,0,0,0,0,1)^T$

Not valid:

$\boldsymbol{x}_I = (1,1,0,0,1,0,0)^T$

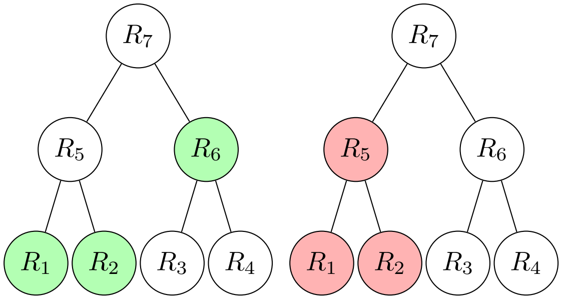

Fig. 3. Right: list of all the possible prunings and the invalid pruning of the rightmost figure. Center: Small BPT with green nodes marked forming the pruning $\boldsymbol{x}_3$. Right: Same BPT, but the marked nodes form an invalid pruning $\boldsymbol{x}_I$.

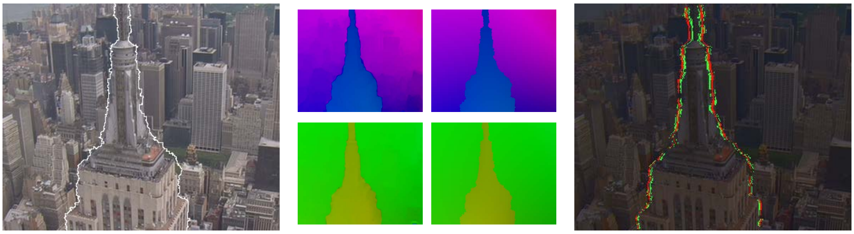

Fig. 4. From left to right. Keyframe with the region borders overlaid in white. Forward and backward estimated flows (top, bottom respectively) and modeled flows. Keyframe with occluded (red) and occluding (green) pixels overlaid.

The modeled flow $\widetilde{\boldsymbol{w}}^{t,q}_{R_i}$ is estimated by robust regression [17] for each region R_i. The constant $\lambda_f = 4 \times 10^3$ is used to prevent oversegmentation and was found experimentally and proved not to be crucial in the overall system performance.

Occlusion relations estimation. With the partition P_f and a flow model available for each region, occlusion relations can be reliably estimated. The (dis)occluding pixel is the forward mapping from I_t to I_q using $\widetilde{\boldsymbol{w}}^{t,q}_{R_i}$, back-projected to image I_t using $\boldsymbol{w}^{q,t}$. $q = t-1, t+1$ for disocclusions and occlusions relations respectively:

$$\boldsymbol{p}_o = \boldsymbol{p}_u + \widetilde{\boldsymbol{w}}^{t,q}_{R_i}(\boldsymbol{p_u}) + \boldsymbol{w}^{q,t}(\boldsymbol{p}_u + \widetilde{\boldsymbol{w}}^{t,q}_{R_i}(\boldsymbol{p_u})) \tag{7}$$

with $p_u \in R_i$. Flow fitting and occlusion relations are shown in Figure 4.

Finding the Final Depth Regions. Once the motion flows are modeled for each region of P_f, occlusion relations can be estimated using (7). Since each relation comprises two different pixels $(\boldsymbol{p}_u, \boldsymbol{p}_o)$, we can use the region information in the BPT to propagate these relations to obtain occlusion relations between

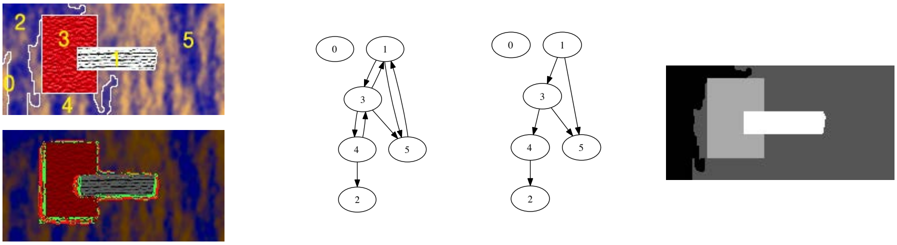

Fig. 5. Depth ordering example. From left to right, top to bottom. Final depth partition with region number. Estimated occluded points in red and occluding points in green. Initial graph. Final graph where cycles have been removed. Depth order image (the brighter the region, the closer).

regions. Therefore, given a partition P, if $\boldsymbol{p}_u$ is in a region R_i and $\boldsymbol{p}_o$ is in a different region R_j, we can conclude that R_j is in front of R_i. However, if both pixels belong to the same region, no relation can be established, since a region cannot be in front of itself. As a result, the pruning idea is to obtain a partition P_d (generally different from P_f) from the BPT which maintains occluded-occluding pairs in different regions, keeping a simple partition. The Algorithm 1 is also used with energy function:

$$E_r(R_i) = \sum_{(\boldsymbol{p}_u,\boldsymbol{p}_o)\in R_i} \frac{1}{N_o} + \lambda_o \tag{8}$$

Where N_o is the total number of estimated occlusion relations. Equation (8) establishes a compromise between the number of occlusion relations kept and the simplicity (the number of regions) of the partition. To avoid oversegmented solutions for P_d, $\lambda_o = 4 \times 10^{-3}$ is introduced.

4.2 Depth Ordering

Once the final partition D is obtained with the energy defined in (8), a graph $G = (V, E)$ is constructed to allow a global reasoning about local depth cues to be done and in particular to deal with conflicting depth information. The vertices V represent the regions of D and the edges E represent occlusion relations between regions. An edge $e_i = (v_a, v_b, p_i)$ goes from node v_a to node v_b if there are occlusion relations between region R_a and region R_b. The weight $p_i = N_{ab}/N_o$ where N_{ab} is the number of occlusion relations between both regions.

To be able to determine a depth order between regions, G should be acyclic. To this purpose, the algorithm defined in [11] is used. It iteratively finds low confident occlusion relations and breaks cycles. Once all cycles have been removed in G, a topological partial sort [18] is applied and each region is assigned a depth order. Regions which have no depth relation, are assigned the depth of their most similar adjacent region according to the distance in the BPT construction. The complete process is illustrated in Figure 5 with a simple example.

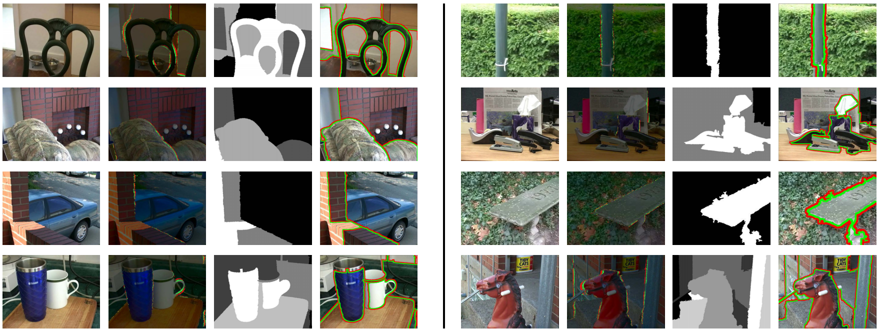

Fig. 6. Results on the CMU dataset. From left to right, for the two columns. 1) Keyframe image and 2) image with occlusion relations (green occluding, red occluded). 3) estimated depth partition, with white regions meaning closer and black meaning further. 4) Figure/ground assignment on contours with green and red overlaid marking figure and ground regions, respectively.

Table 1. Our method vs. [8] on the percentage of correct f/g assignments

Dataset	CMU	BDS
[8]	83.8 %	68.6 %
Proposed system	88.0 %	92.5 %

5 Results

The evaluation of the system is performed at keyframes of several sequences, comparing the assigned f/g contours against the ground-truth assignments. When two depth planes meet, the part of the contour belonging to the closest region is assigned figure, or ground otherwise, see Figure 6. The datasets are the Carneige Mellon Dataset (CMU) [19] and the Berkeley Dataset (BDS) [8]. Results contain sequences with ground-truth data (30 for the CMU, 42 for the BDS). Table 1 shows the percentage of correct f/g assignments on detected contours.

It can be seen in Table 1 that the proposed system outperforms the one presented in [8], showing that motion occlusions are a reliable cue for depth ordering. Results of depth ordering can be seen in Figures 6 and 7, showing that motion occlusions may work over a variety of situations: static scenes, moving foregrounds, moving background or even multiple moving objects.

It is interesting to take a closer look at results in Figure 6 and the occlusion-relation estimation, shown in the second column for each image. In spite of the simplicity of the optical flow estimation algorithm, occlusion points were reliably estimated. Modeling the flows with a projective model provides reliable flow information and proved to be a crucial step for occlusion relations estimation.

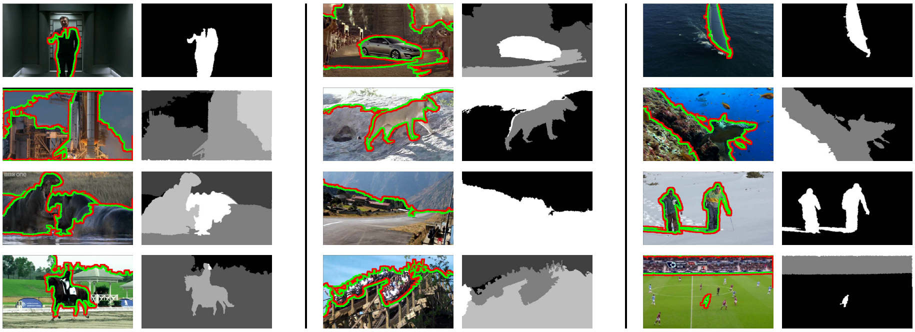

Fig. 7. Results on some of the sequences of the BDS dataset. For each column, the right image corresponds to the keyframe with figure/ground assignments on contours overlaid. The left image correspond to the final depth ordered partition.

6 Conclusions

In this work, a system inferring the relative depth order of the different regions of a frame relying only on motion occlusion has been described. Combining a variational approach for optical flow estimation and a region based representation of the image we have developed a reliable system to detect occlusion relations and to create depth ordered partitions using only these depth cues. Comparison with the state of the art shows that motion occlusions are very reliable cues. The presented approach, although using only motion information to detect boundaries, achieves better results on f/g assignment than [8] which is considered as the state of the art in f/g assignment.

There are many possible extensions to the proposed system. First, to provide more occlusion information on a given frame, a bigger temporal window could be used to retrieve motion occlusions. Second, we can take profit of other monocular depth cues, such as T-junctions and convexity to help on motionless depth relations. Although state of the art results on these cues [11] show that they are less reliable than motion occlusions, they could be a good complement to the system. We believe also that occlusions caused by motions can be propagated throughout the sequence to infer a consistent depth ordering across multiple frames. Since results on single frames are promising, sequence depth ordering seems plausible.

References

1. Ono, M.E., Rivest, J., Ono, H.: Depth perception as a function of motion parallax and absolute-distance information. Journal of Experimental Psychology: Human Perception and Performance 12, 331–337 (1986)
2. Qian, N., Qian, D.N.: Binocular disparity and the perception of depth. Neuron 18, 359–368 (1997)

3. He, X., Yuille, A.: Occlusion Boundary Detection Using Pseudo-depth. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 539–552. Springer, Heidelberg (2010)
4. Turetken, E., Alatan, A.A.: Temporally consistent layer depth ordering via pixel voting for pseudo 3D representation. In: Proc. 3DTV Conf.: The True Vision - Capture, Transmission and Display of 3D Video, pp. 1–4 (2009)
5. Chang, J.Y., Cheng, C.C., Chien, S.Y., Chen, L.G.: Relative depth layer extraction for monoscopic video by use of multidimensional filter. In: Proc. IEEE Int. Multimedia and Expo. Conf., 221–224 (2006)
6. Li, P., Farin, D., Gunnewiek, R.K., de With, P.H.N.: On creating depth maps from monoscopic video using structure from motion. In: 27th Symposium on Information Theory in the Benelux, pp. 508–515 (2006)
7. Zhang, G., Jia, J., Wong, T.T., Bao, H.: Consistent depth maps recovery from a video sequence. IEEE Trans. Pattern Anal. Mach. Intell. 31, 974–988 (2009)
8. Sundberg, P., Brox, T., Maire, M., Arbelaez, P., Malik, J.: Occlusion boundary detection and figure/ground assignment from optical flow. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO (2011)
9. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE Trans. on Image Processing 9, 561–576 (2000)
10. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: Pajdla, T., Matas, J. (eds.) ECCV 2004, Part IV. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
11. Palou, G., Salembier, P.: From local occlusion cues to global depth estimation. In: IEEE Int. Conf. on Acoustics Speech and Signal Processing, Kyoto, Japan (2012)
12. Sun, D., Sudderth, E., Black, M.J.: Layered Image Motion with Explicit Occlusions, Temporal Consistency, and Depth Ordering. In: Lafferty, J.D., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A. (eds.) Advances in Neural Information Processing Systems, vol. 23. Curran Associates, Inc. (2010)
13. Alvarez, L., Deriche, R., Papadopoulo, T., Sánchez, J.: Symmetrical dense optical flow estimation with occlusions detection. Int. Journal of Computer Vision 75, 371–385 (2007), doi:10.1007/s11263-007-0041-4
14. Calderero, F., Marques, F.: Region merging techniques using information theory statistical measures. IEEE Trans. on Image Processing 19, 1567–1586 (2010)
15. Vilaplana, V., Marques, F., Salembier, P.: Binary partition trees for object detection. IEEE Trans. on Image Processing 17, 2201–2216 (2008)
16. Kanatani, K.: Transformation of optical flow by camera rotation. IEEE Trans. on Pattern Analysis and Machine Intelligence 10, 131–143 (1988)
17. Andersen, R.: Modern methods for robust regression. Quantitative applications in the social sciences, vol. 152. Sage Publications (2008)
18. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd revised edn. The MIT Press (2001)
19. Stein, A.: Occlusion Boundaries: Low-Level Detection to High-Level Reasoning. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (2008)