

# RELATIVE DEPTH FROM MONOCULAR OPTICAL FLOW

Enric Meinhardt-Llopis, Olivier D'Hondt

Gabriele Facciolo, Vicent Caselles

Fundació Barcelona Media

Universitat Pompeu Fabra

## ABSTRACT

We present a method to compute the relative depth of moving objects in video sequences. The method relies on the fact that the boundary between two moving objects follows the movement of the object which is closest to the camera. Thus, the input of the method is a segmentation (to know the boundaries of objects) and an optical flow (to know the movement of the objects). The output of the method is a relative ordering of the neighboring segments. In fact, this output only provides a cue of the desired relative ordering, just like T-junctions typically provide a cue of the relative ordering of the objects around them. These cues can be used later as heuristics or as starting points for higher-level algorithms for image and video-processing.

**Index Terms**— monocular, depth estimation, optical flow, segmentation

## 1. INTRODUCTION

Depth perception from a single image is an easy task for the human visual system: people who have lost an eye can lead a normal life, and everybody can easily reconstruct real-world scenes from a single photograph. According to current theories of vision [1] this is achieved by integrating the information of several *depth cues*. There is a rather large list of cues for depth perception, including perspective, texture gradients, distance fog, focus, *T*-junctions, shading and size. Each of these cues is not sufficient alone, and any single one may lead to incorrect depth perception. However, combining the information from all these cues produces very reliable information. In computer vision, it is easy to obtain information from each of these cues, but difficult to integrate the information from all of them into a single 3D reconstruction.

Depth perception from multiple images adds new cues to this list, thus increasing the reliability of the depth information. The most prominent addition to the list is *parallax*, which can be produced either by binocular perception or by observer movement. A different cue, closely related to the purpose of this work, is *depth from motion*, whereby objects moving towards the observer increase in size, and objects moving away from the observer decrease in size. The brain is very fast and very precise in using this information to compute the crash time of approaching objects. The change

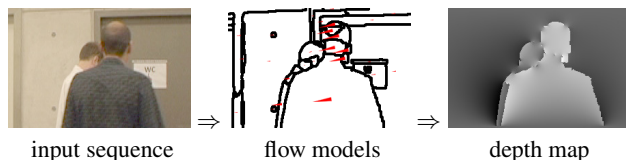
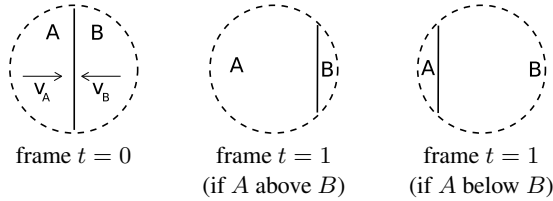


Fig. 1: Our algorithm is the second step in this diagram.

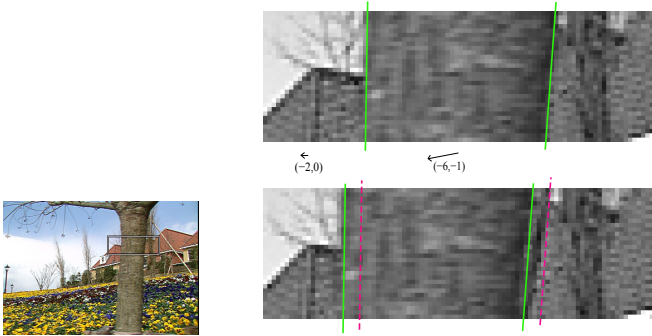
of size can be expressed locally by means of the divergence of the optical flow: the optical flow of an approaching object will have positive divergence, and the optical flow of a distancing object will have negative divergence. This idea has been used successfully for collision avoidance in free-moving robots [2]. Notice that this is a local criterion which works on the interior of objects, and tells how the objects move in the direction of the observer.

This work introduces a new cue for depth perception from multiple images, based on occlusions. Unlike depth from motion, this cue provides a local criterion which works on the boundaries between objects, looking how the objects move in the direction perpendicular to the line of sight, and telling how the objects are located in the direction of the observer. The cue is based on the fact that the boundary between two objects moves in the same way as the object which is closer to the observer (because the closest object occludes the other behind that boundary). In terms of divergences, an occlusion boundary produces a band of highly negative divergence around it, and a disocclusion boundary produces a band of highly positive divergence around it. Compared to parallax or depth from motion, the proposed criterion is more general because it does not assume the rigidity of the objects (although our naive implementation does). On the other hand, it only gives a relative ordering of the objects, not a distance.

The goal of this work is to highlight the importance of occlusions and disocclusions in the perception of depth, ignoring all other depth cues. In particular, it does not propose a method to create 3D reconstructions. Our goal is to understand the kind of information that can be extracted from occlusions alone, just like other works focus on information that can be extracted from shading alone [3] or from *T*-junctions alone [4]. There are other works that treat this same problem under different constraints [5, 6, 7], or as a by-product of segmentation or optical-flow algorithms [8, 9, 10].



**Fig. 2:** Local illustration of the criterion, around a point located on the boundary between  $A$  and  $B$ . Region  $A$  moves in the direction  $v_A$ , region  $B$  moves in the direction  $v_B$ . The boundary between the two regions moves in the same direction as the region which is above.



**Fig. 3:** Local illustration of the criterion. Top: first frame, with segmentation boundaries in green. Bottom: second frame, with segmentation boundaries in green. The mean flows of each region are shown. The mean flow of the tree correctly moves the boundaries from one frame to the next. The mean flow of the background moves the boundaries to another location (dotted lines). According to the criterion, both boundaries of the tree support the hypothesis that the tree is above the background.

## 2. PERCEPTUAL PRINCIPLE

Let us assume that we have a perfectly computed dense optical flow and a perfect segmentation of the video frames into objects. In that case, the following criterion provides a relative ordering of neighboring objects: *The boundary between two moving objects in the scene follows the movement of the object which is closest to the camera.* See Figures 2 and 3 for two examples of this criterion. We assume that the criterion is intuitively sound and no further explanation is given beyond these two figures.

Actually, the criterion is not true in full generality. There are some situations where it leads to an incorrect depth ordering. For example, when a sheet of surface slides behind a sharp edge (see Figure 4). For practical purposes, we will ignore these cases. It is up to the user of the method to decide whether these counterexamples are relevant for the intended application.



**Fig. 4:** Counterexample to the criterion: flat flexible object folding behind a corner. In that video sequence, if the optical flow is correctly computed, the criterion gives a wrong relative ordering on the marked areas.

## 3. FLOW AND SEGMENTATION

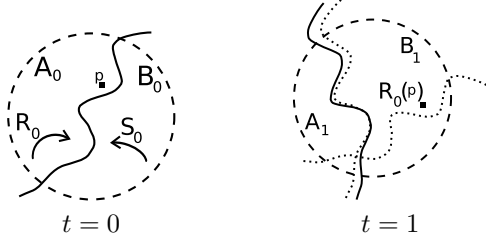
The proposed method requires two ingredients as input, which can be regarded as independent: a spatio-temporal segmentation of the whole video sequence and a dense optical flow of each frame. Although these ingredients are not really independent (for instance the segmentation may use information from the optical flow to enhance its temporal coherence) it is useful to think of them as independent.

For the segmentation, we use the piecewise constant Mumford-Shah model on the whole spatio-temporal color volume to build a multi-scale hierarchical segmentation [11]. We use either classical 3D connectivity or the connectivity induced by an optical flow. The resulting hierarchical segmentation is then pruned by setting manually the desired number of segments, producing an over-segmentation of the video. The point of this over-segmentation is that it must be as coarse as possible without mixing different objects on the same segment. Its segments are spatio-temporal tubes which, when intersected with the frames, produce a segmentation of the video which is spatially coherent. They can be stored in a data structure that provides high level access to these segments [12].

For the optical flow, we settled on the method by Brox et al. [13]. It can be argued that the choice of optical flow method is not critical, because the errors on the flow vectors are smoothed out when we take models of movement for each region. Since our method requires the optical flow to be very precise at the boundaries of objects, we use the pre-computed segmentation to achieve this sharpness: the flow vectors inside each segment are used to build an (affine or projective) model for the movement of this object, and then each flow vector is replaced by the result of the corresponding model. This forces the optical flow to be smooth inside each region, and discontinuous along the boundaries of regions.

## 4. IMPLEMENTATION

After stating the perceptual criterion for monocular depth estimation, we introduce an algorithm that uses this criterion to compute a relative depth ordering on a video. The algorithm assumes that we have already computed the optical flow and



**Fig. 5:** Notation used on the description of the algorithm. The two figures depict the local situation around a boundary separating regions  $A$  and  $B$ . Point  $p$  belongs to region  $A_0$ , but  $R_0(p)$  belongs to  $B_1$ . This means that  $B$  is occluding  $A$ . If  $A$  was above, we should have  $R_0(p)$  belong to  $A_1$ .

the segmentation of the video sequence, and estimated the model of movement for each segment, as explained in Section 3.

The algorithm works by selecting all pairs of neighboring regions, and makes a decision on which region of each pair is above or below the other. Suppose that we have two neighboring regions  $A$  and  $B$ . Let us define the following notation (see also Figure 5):

- $A_i$  is the region  $A$  on frame  $t = i$ , for  $i = 0, 1$
- $B_i$  is the region  $B$  on frame  $t = i$ , for  $i = 0, 1$
- $c_i$  is the boundary between  $A_i$  and  $B_i$ , for  $i = 0, 1$
- $R_0$  is the model of movement between  $A_0$  and  $A_1$
- $S_0$  is the model of movement between  $B_0$  and  $B_1$

Notice that if there are no occlusions and the models of movement are correct, then we have  $R_0(A_0) = A_1$  and  $S_0(B_0) = B_1$ . Thus, since the transformations are continuous, it must be that  $R_0(c_0) = c_1$  and  $S_0(c_0) = c_1$ . This implies that  $S_0 = R_0$ , both movements are the same. When there are occlusions, the movements of  $A$  and  $B$  differ. The criterion introduced in Section 2 states that  $c_1$  is the image of  $c_0$  under the movement of the object which is above. Thus, comparing  $R_0(c_0)$  and  $S_0(c_0)$  to  $c_1$ , we can decide which of  $A$  or  $B$  is above.

There are, in principle, many ways to implement the curve comparison described on the previous paragraph. We propose to compare the displaced regions using Hausdorff distance (area of symmetric difference). Thus, we compare  $R_0(A_0)$  to  $A_1$  and  $S_0(B_0)$  to  $B_1$ . The pair which matches better will correspond to the region which is above. The advantage of this method is that it can be implemented very easily in linear time, by moving each pixel in the video according to the movement of its region, and looking whether it goes to the corresponding region on the next frame, or to a different region. This comparison is illustrated on Figure 5. Here follows the pseudo-code of the algorithm:

**Input:** a spatio-temporal segmentation of a video and a dense optical flow  $F$ .

**Output:** a relative ordering of pairs of neighboring regions of the segmentation.

#### Algorithm:

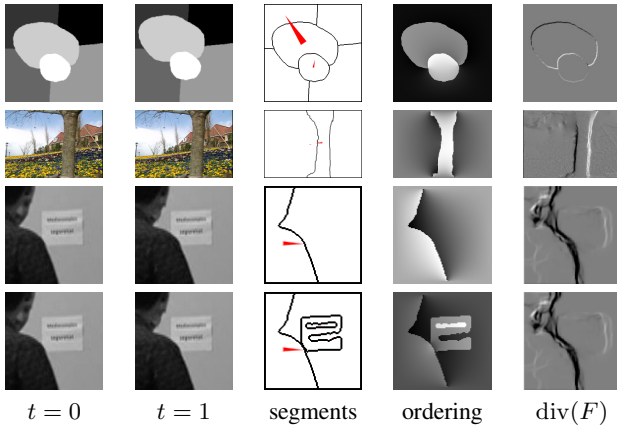
1. **for** each region  $A_t$  on frame  $t$  **do**
2.    $M_{A_t, A_{t+1}} := \text{movement\_model}(A_t, F)$
3.   **for** each pixel  $p$  on frame  $t$  **do**
4.      $A_t := \text{region\_of\_pixel}(p)$
5.      $q := M_{A_t, A_{t+1}}(p)$
6.      $B_{t+1} := \text{region\_of\_pixel}(q)$
7.     **if**  $B \neq A$  **then**
8.       vote +1 that  $B_t$  is above  $A_t$

The algorithm can be interpreted easily: We move each pixel of the video according to the motion model of its region  $A$ . If it falls in a different region  $B$ , that means that  $B$  is occluding  $A$ , and we record this fact. See Figure 5 for a graphical explanation, where  $R_0 = M_{A_0, A_1}$  and  $S_0 = M_{B_0, B_1}$ .

There are some remarks to be done regarding this algorithm. First: The output of the algorithm is a list of votes for each pair of regions, saying which one is above. By setting a threshold on difference of votes (e.g., 1), we obtain the desired partial ordering. Second: As it is stated, the algorithm only finds *occlusions*, but not *disocclusions*. To obtain those, we must run it “backwards in time”. This can be done by using either a bi-directional optical flow or by inverting the movement of each region  $R_{A_{t+1}, A_t} := R_{A_t, A_{t+1}}^{-1}$ . Third: we can easily enforce temporal consistency of the method by propagating the votes along the tubes. The last remark is that the algorithm gives a relative ordering to *every* pair of neighboring segments. In practice, we work with oversegmented videos, where many segments are parts of the same rigid objects. In that case, most of the information given by the algorithm will be neither meaningful nor useful.

For removing this kind of clutter, we propose some heuristics based on the optical flow divergence. As noted on the introduction, the divergence of the optical flow is higher at the occlusion boundaries. This fact is used in two ways. First, as a bias for the voting that the algorithm does at each occluded pixel: the vote of each pixel is weighted by the absolute value of the divergence at this pixel. Second, as a weighting of the whole boundaries between regions: we validate each boundary according to its length and to the mean divergence of its pixels, using the same criterion as in [14] for edge detection. The combination of these two heuristics based on divergences reduces most of the clutter in the output.

The output of the proposed algorithm is a relative ordering of some pairs of neighboring regions. It can be regarded as a set of oriented curves on the image domain, which in turn may be visualized as a gray-level image  $u$  in the following way. Let  $u$  be the solution of Laplace equation  $\Delta u = 0$  on the image domain minus the set of oriented curves. As Neumann boundary conditions along these curves, use the oriented normals to them. The gray-level values of the image  $u$  have no absolute meaning, but the image has discontinuities along the curves, and the highest value corresponds to the closest object.



**Fig. 6:** Analysis of some videos using the proposed method.



**Fig. 7:** Video inpainting with or without using the depth information (D.I.) produced by the proposed algorithm.

## 5. RESULTS AND CONCLUSION

We display the results of our analysis for three sample videos on Figure 6. In each case we show, from left to right: 1,2) Two consecutive video frames of the sequence. 3) A segmentation of the first frame with arrows indicating the movement of each segment. 4) The computed orientation of each boundary. 5) The divergence of the optical flow. The orientations of the boundaries are visualized as follows: the light side of the boundary corresponds to the object which is above, the dark side to the object which is below. The divergence of the flow is displayed in order to realize that the interesting occlusion activity happens at places where  $|\text{div}(F)|$  is high.

At this point, the main problem of the results is *clutter*, due to the use of too fine segmentations. If we fine-tune by hand the parameters of the segmentations to avoid over-segmentation, we can suppress most of the clutter, with some effort. The weighting by the divergences explained above can then be used to weight the importance of each boundary, in order to sort them by their meaningfulness.

Regarding the applications, the main role of the proposed method is to be integrated into a larger depth estimation framework, which uses other cues besides this one. Otherwise, there are some particular problems where this method provides applicable results. For example, we have applied this criterion to the problem of video inpainting (Figure 7), in order to avoid getting information from an object that occludes the inpainting region.

**Acknowledgements:** This work was partially funded by Mediapro through the Spanish project CENIT-2007-1012 i3media and by the Centro para el Desarrollo Tecnológico Industrial (CDTI), within the Ingenio 2010 initiative. O. D’Hondt acknowledges support from the Torres Quevedo program of the MICINN (Spain). V. Caselles, G. Facciolo and E. Meinhardt also acknowledge partial support by MICINN project, reference MTM2009-08171, by GRC reference 2009 SGR 773. V. Caselles acknowledges partial support by “ICREA Acadèmia” prize for excellence in research, the last two funded by the Generalitat de Catalunya.

## 6. REFERENCES

- [1] D. Marr, *Vision*, Henry Holt and Co., 1982.
- [2] R.C. Nelson and J. Aloimonos, “Obstacle avoidance using flow field divergence,” *IEEE Trans. Pattern Anal. Machine Intelligence*, vol. 11, no. 10, pp. 1102–1106, 2002.
- [3] B.K.P. Horn, “Obtaining shape from shading information,” in *Shape from shading*. MIT Press, 1989, pp. 123–171.
- [4] M. Dimiccoli, J.M. Morel, and P. Salembier, “Monocular Depth by Nonlinear Diffusion,” in *Comp. Vis., Graph, Image Process.* IEEE, 2008, pp. 95–102.
- [5] G. Adiv, “Determining three-dimensional motion and structure from optical flow generated by several moving objects,” *IEEE Trans. Pattern Anal. Machine Intelligence*, , no. 4, pp. 384–401, 1985.
- [6] D. Koller, J. Weber, and J. Malik, “Robust multiple car tracking with occlusion reasoning,” *Computer Vision ECCV’94*, pp. 189–196, 1994.
- [7] A. Schodl and I. Essa, “Depth layers from occlusions,” in *Computer Vision and Pattern Recognition, 2001*, 2005, vol. 1.
- [8] A.S. Ogale et al., “Motion segmentation using occlusions,” *IEEE Trans. Pattern Anal. Machine Intelligence*, pp. 988–992, 2005.
- [9] D. Sun, E.B. Sudderth, and M.J. Black, “Layered Image Motion with Explicit Occlusions, Temporal Consistency, and Depth Ordering,” 2010.
- [10] ME Sargin, L. Bertelli, BS Manjunath, and K. Rose, “Probabilistic occlusion boundary detection on spatio-temporal lattices,” in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2010, pp. 560–567.
- [11] G. Koepfler, C. Lopez, and J.-M. Morel, “A Multiscale Algorithm for Image Segmentation by Variational Method,” *SIAM J. Numer. Anal.*, vol. 31, no. 1, pp. 282–299, 1994.
- [12] C.C. Dorea, M. Pardàs, and F. Marques, “Trajectory tree as an object-oriented hierarchical representation for video,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 4, pp. 547–560, 2009.
- [13] T. Brox, A. Bruhn, N. Papenbergh, and J. Weickert, “High accuracy optical flow estimation based on a theory for warping,” *Computer Vision-ECCV 2004*, pp. 25–36, 2004.
- [14] A. Desolneux, L. Moisan, and J.-M. Morel, “Edge Detection by Helmholtz Principle,” *J. Math. Imaging Vis.*, vol. 14, no. 3, pp. 271–284, 2001.