# Computation of motion and occlusion relation of objects from the motion of their boundaries

**A. Akhriev, A. Bonch-Osmolovsky, and A. Prusakov**
**Institute of Information Technologies, RRC Kurchatov Institute, 1 Kurchatov sq. Moscow, 123182 Russia**
**e-mail: ambo@mail.ru**

## Abstract

An original approach for motion segmentation using the movement of object boundaries is presented. Our treatment is focused on the case of two motions (background and foreground) and just two frames but can be extended to multiple motions and multiple frames. Segmentation results are presented and some shortcomings of the approach based on global motion models are discussed.

**Keywords**: edge point, edge segment, boundary point, boundary, fuzzy clustering, labeling, occlusion.

## Introduction

Segmenting frames into coherently moving regions is an important issue in video sequence analysis. There are two main sources of motion information: the behavior of boundaries and local color and intensity changes between frames (optic flow). In many situations, the behavior of boundaries is sufficient for objects separation, which is readily exemplified by flat-color animations. Natural images may also have low-texture and almost constant color regions, were optic flow data are not to be trusted.

In this paper, we formulate and develop a new "semantic" approach to video sequence segmentation of (splitting frames into moving objects and the background) based on the analysis of boundary motion and occlusion relations between boundaries.

Our approach consists of two steps and uses two successive frames. First, a fuzzy clustering algorithm is applied to reliable edge points (with strong color gradient and good localization) to obtain their tentative fuzzy partitioning into motion groups (in terms of fuzzy membership coefficients) and calculate motion parameters for each group. Next, region boundaries produced by color segmenter are assigned motion labels. Each label comes with a weight that shows how well this boundary is aligned with a similar boundary in the next frame by the motion found for the given cluster. Then, a label relaxation procedure incorporating occlusion relations is applied to classify regions of the first frame either as unambiguously belonging to one motion cluster and, therefore, occlusion layer or ambiguous.

When most of this study was finished, we became aware of ref. [1], where a more or less similar approach is taken. Even so, our work differs from [1] by the choice of Canny-type edges for motion calculation, by the use of fuzzy clustering instead of the EM procedure, by the label relaxation procedure, and in many important implementation details. In a separate communication, we shall report how ambiguity can be reduced and conflicts resolved by a multiframe tracking procedure. To keep this paper short, we had to simplify algorithms and omit many details, but the key points of our method are fully described.

## Calculation of motion

As a first step of motion segmentation of two sequential frames, we perform color segmentation of both frames using an original color segmenter. This segmenter tends to form regions of uniform surface color rather than image color, but that is not very important. The only requirement (in this work) is that object and background regions be never merged. Therefore, oversegmentation is to be preferred to undersegmentation. The result is represented as a graph of adjacency with regions as nodes and their boundaries as edges. We refer to points on these boundaries as *boundary points*, and points yielded by a Canny-type color edge detector (based on ideas outlined in [2–4]) as *edge points*. A chain of pixels extracted by the edge detector is called an *edge segment*.

Edge segments show higher stability between frames than boundary segments and were chosen for motion calculation as described below. All extracted edge segments are smoothed with shrinkage compensation (by a method extending the ideas in [5]) and tangents at every edge point are calculated. Next, "good" edge points of the first frame are selected on the condition that the color gradient strength exceeds markedly its noise level estimated from the image. The second requirement is that good edge points be well localized, as estimated by a special procedure Good edge points are placed in a list.

First, for each edge point of the *first* frame, a *tentative* correspondence with edge segments of the *second* frame is established. For this purpose, a search region around every edge point of the first frame is defined with the radius equal to the maximum expected movement. Edge segments of the second frame falling within this region are considered as candidates. The most suitable segment is found by computing the mean difference of color component gradients at the given edge point of the first frame and points of the tested edge segment in the second frame. The condition for a seg-

ment point to be used in averaging is that its color component gradients (gradR, gradG, gradB) be roughly parallel to the corresponding gradients in the first frame. The calculated differences are compared with each other and with the noise threshold. The difference for the accepted edge segment should be notably smaller than for other candidates. If no segment is accepted or the search region is empty, the point of the first frame is dropped from the list of good points. The tentative correspondence of a given point indicates the edge segment of the second frame to be used in match search in motion calculation. This approach may occasionally fail, e.g., on occluding boundaries but in most cases the matching segment is found correctly. It should stressed that, at this stage, our goal is to estimate the major motions present in the image rather that classify the displacements of individual points.

Fuzzy edge point clustering into motion groups (with concurrent calculation of motion parameters) consists in simultaneous calculation of motions and establishing matches between "good" points of the first frame and points on tentatively selected segments of the second frame to minimize the following expression

$$\delta^2 = \sum_{c=1}^{K} \delta_c^2 = \sum_{c=1}^{K} \sum_{i=1}^{N} u_{ci}^2 d^2(c,i), \qquad (1)$$

where $$d^2(c,i) = \left\| \mathbf{T}_c \mathbf{p}_i - \mathbf{q}_i \right\|^2, \qquad (2)$$

under the well-known conditions on fuzzy membership coefficients [6, 7]. In these equations, $\mathbf{p}_i$ is the *i*th edge point of the first frame, $\mathbf{q}_i$ is the corresponding edge point on the second frame, $\mathbf{T}_c$ is the transformation of the *c*th motion group (cluster), $u_{ci}$ is the membership coefficient of the *i*th point in the *c*th motion group, $d(c,i)$ is the distance between the *i*th point of the first frame shifted by the *c*th motion and the matching point of the second frame, $K$ is the given number of motion groups, $N$ is number of "good" edge points of the first frame, and ||.|| is the norm of 2D vectors. As noted above, the corresponding points in the second frame are sought on the tentatively corresponding edge segments. A similarity or an affine transform can be used as a motion model.

To optimize (1), the P-step of a classical fuzzy-clustering method needs to be modified since the distance (2) cannot be computed directly because the correspondence in (2) is updated at each step. This modification follows the ideas of [8]. For all good points $\mathbf{p}_i$ of the first frame, the *nearest* points $\mathbf{q}_i$ on corresponding edge segments of the second frame are found and the cluster motion parameters $\mathbf{T}_c$ are calculated to minimize the following expression with coefficients $u_{ci}$ fixed,

$$\delta_c^2 = \sum_{i=1}^{N} u_{ci}^2 \left\| \mathbf{T}_c \mathbf{p}_i - \mathbf{q}_i \right\|^2, \quad \forall c \in \{1 \ldots K\}$$

Next, the obtained transform is applied to first frame points to yield their the new positions, and the procedure repeats until reaching a minimum.

The fuzzy-clustering procedure begins with just one cluster. New clusters are successively added from outliers, and this is accompanied by complete reclustering. The process runs until the predetermined number of clusters is reached (2 or 3, typically). The choice of the limiting number of clusters can be left to the program in the spirit of [7, 9].

Now, having motion clusters for edge points, we pass on to motion evaluation of region boundaries produced by the color segmenter. The measure of compatibility of a given boundary with a given motion is defined by

$$d_{ci} = \frac{1}{N} \sum_{k=1}^{N} \min_{\mathbf{q} \in V_{\mathbf{p}}} \left\| \mathbf{T}_c \mathbf{p}_{ik} - \mathbf{q}_{ik} \right\| \qquad (3)$$

where the summation is over all (or decimated) points $\mathbf{p}_{ik}$ of the *i*th boundary, $\mathbf{q}_{ik}$ is the closest boundary point of the second frame with a similar tangent direction found in a small neighborhood $V_{\mathbf{p}}$ of $\mathbf{p}_{ik}$ shifted to the new position by motion $\mathbf{T}_c$, and ||.|| is the norm of 2D vectors. The robustness of (3) can be increased by taking into account a fixed fraction of points (e.g., 70%) with minimal residuals. The compatibility measures calculated for all boundaries and all motions are used as input in the motion labeling procedure.

# Motion labeling

The motions of objects and their boundaries are supposed to satisfy two constraints: (1) objects with similar motions belong to one depth layer; (2) in each layer, the interframe motion can be reasonably well approximated by the adopted model, e.g., the similarity transformation. Let us also assume that the scene is composed of just two depth layers (an object and the background) and that none of color-segmented regions includes both background and foregrounds pixels. The overall goal is to assign motion and, thereby, depth labels to all regions in agreement with motions (labels) of their boundaries and with minimal ambiguity.

Let us assume that label *1* is always assigned to the dominant motion associated with the background. This is important for the labeling procedure. In the motion calculation described above, the first cluster most often corresponds to the background motion. Nevertheless, to ensure correct motion ordering, a validation procedure has been developed but is not described here for brevity. Label *2* is the label of the foreground motion.

The plan is to assign to each boundary a set of labels with weights ($m$, $w_m$) rating the compatibility between the actual boundary motion and that of cluster $m$. Next, the obtained label sets will be purged based on the outlined motion assumptions and labels will be propagated from boundaries to regions. The purging (relaxation) procedure is based on the three following rules and their implications:

(1) the boundary of two regions is part of and moves together with the region closest to the viewer;

(2) all common boundaries of any regions with labels $m_1$ and $m_2$ must be given the same label;
(3) a region is never assigned a label other than that of its boundaries.
Rules (1) and (2) are generally similar to those formulated in [1] and are physical by their nature. In other words, it can be claimed that any depth and motion assignment consistent with these rules is physically realizable. Rule (3), however, is introduced merely to eliminate the otherwise irreducible ambiguity between object parts and holes.

Label relaxation can be accomplished by different mechanisms. In [1], the net solution probability was maximized by simulated annealing. Our approach differs in that its priority is to identify unambiguously labeled regions.

Let us introduce the normalized coefficient of compatibility of the *i*th boundary with the *c*th motion by

$$w_{ci} = \frac{(1/d_{ci})}{\sum_{k=1}^{2}(1/d_{ki})}.$$

For each boundary *i*, only such labels *c* are retained for which $w_{ci} > 0.33$. Unfortunately, in real videos, many boundaries, especially short ones, may carry both labels (Fig. 1). The surviving labels for each boundary are entered in the list of valid labels $G_i$. The purpose of our algorithm is not to find all legal labelings (i.e., satisfying rules (1)–(3)), which might involve intractable combinatorics, but rather to produce, for each region, a list of labels such that every label on this list participates in some legal region labeling. The computation proceeds in three steps:
*Step1:* Scan all regions and find those with at least one boundary such that its $G_i$ consists of a single label *1*, and assign label *1* to these regions.
*Step2:* Scan all regions and find those with at least one boundary such that its $G_i$ consists of a single label *2*. If this region shares this boundary with a region that got label *1* in step 1, then this region is assigned label *2*.
*Step3:* All regions that were not labeled in steps 1 and 2 get both labels.

This algorithm is readily extended to the case of more than two labels and can be proved to solve the stated problem whenever a legal labeling exists. There is also a fast algorithm not described here to check the existence of at least one legal labeling. As a byproduct, this algorithm identifies the so-called conflict regions, which arise when the occluding object is partly hidden by the background, a non-trivial case of mutual occlusion violating our basic assumptions. The problem can be resolved by introducing a temporary label *3* as a label for occluding part of the background and by modifying the labeling algorithm.

It is not uncommon that some regions get both labels *1* and *2* and remain ambiguous (Fig. 2) due to local similarity of different motions, faults of color segmenter, and the presence of many short boundaries. This ambiguity can be effectively resolved in real videos by testing the applicability of the obtained motions to inner pixels of regions in the original (unsegmented) image. Let region *A* have both labels and consider three successive frames of a video sequence. Nearly all points of the middle frame are visible in at least one of the adjacent frames. Therefore, an occlusion-tolerant interframe difference at point **p** with respect to motion $T_c$ can be defined as

$$D(\mathbf{p}, c) = \min\left(\left\| \mathbf{C}_{t+1}(\mathbf{T}_c\mathbf{p}) - \mathbf{C}_t(\mathbf{p}) \right\|, \left\| \mathbf{C}_{t-1}(\mathbf{T}_c^{-1}\mathbf{p}) - \mathbf{C}_t(\mathbf{p}) \right\|\right)$$

where *c* is the motion label, $\mathbf{C}_t(\mathbf{p})$ is the color of pixel **p** in frame *t*, and $\|.\|$ is the norm in the color space. By comparing the medians of $D(\mathbf{p}, c)$ computed for all internal pixels of *A* for both motions ($c=1, 2$), one of these labels may be selected (Fig. 3). It is reasonable apply this filter in conjunction with tracking over multiple frames.

## Results and future work

The outlined approach in many practical situations is able to segment a scene into two motion groups with an acceptable degree of ambiguity. Figures 1–3 show the segmentation results for a couple of frames from the standard *Foreman* sequence. These frames roughly satisfy the affine motion assumption. The blame for residual ambiguity can be partly attributed to the color segmenter, but mostly has to do with more general motion segmentation problems:
1. Objects quite often fail to move as prescribed by a simple (e.g., affine) model. The use of a "flexible" does not solve the problem either, because motion-based segmentation and object segmentation are not the same.
2. Motion of large objects masks that of small objects.
3. Instances of self-occlusion and mutual occlusion limit the applicability of the layers concept and labeling methods.

A more physically sound approach, in our view, would be to couple segmentation with the detection of occluding boundaries and exploit motion data to this end. This can be accomplished by the following means:
(1) estimating local motions rather than global ones and expanding the solution to the entire frame;
(2) detecting occluding boundaries by analyzing the discontinuities of the optic flow and the motion at T-junctions.
(3) tracking motion and regions over multiple frames;
(4) developing more sophisticated labeling techniques to deal with mutual occlusion.

This work was partially funded by a contract with the Samsung Advanced Institute of Technology.

## References

1. Smith, P., Drummond, T., and Cipolla, R. Segmentation of multiple motions by edge tracking between two

frames. In M. Mirmehdi and B. Thomas (eds.), *Proc. 11th British Machine Vision Conference*, vol. 1, pp. 342–351, Bristol, September 2000.
2. Zenzo, S.D. A note on the gradient of a multi-image. *CVGIP*, vol. 33, pp. 116–125, 1986.
3. Canny, J. A computational approach to edge detection. *IEEE Trans., PAMI*, vol. 8, pp. 679–698, 1986.
4. Sapiro, G. and Ringach, D.L. Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Trans., Image Processing*, vol. 5, pp.1582–1586, 1996.
5. Lowe, D.G. Organization of smooth image curves at multiple scales. *IJCV*, vol. 3, pp.119–130, 1989.
6. Bezdek J.C. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, New York, 1981.
7. Gath, I. and Geva, A.B. Unsupervised optimal fuzzy clustering. *IEEE Trans., PAMI*, vol. 11, pp. 773–781, 1989.
8. Zhang, Z. Iterative point matching for registration of free-form curves and surfaces. *IJCV*, vol. 13, pp. 119–152, 1994.
9. Frigui, H. and Krishnapuram, R. A robust competitive clustering algorithm with application in computer vision. *IEEE Trans., PAMI*, vol. 21, pp. 450–464, 1999.

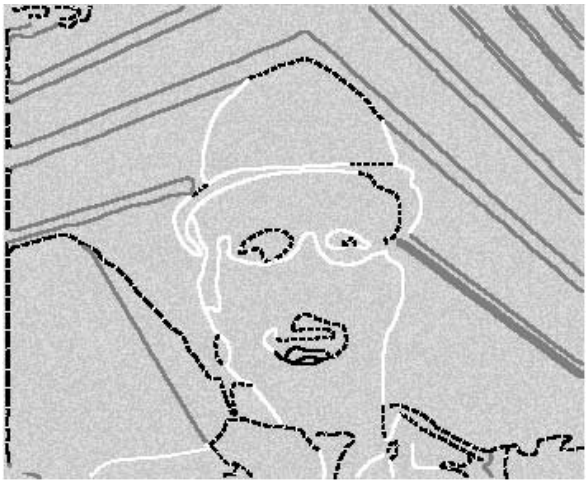

Fig.1. Each calculated motion is applied to every region boundary in the first frame. A boundary is assigned a label *L* if, being displaced by the *L*th motion, it aligns well with a suitable boundary in the second frame. Boundaries uniquely marked by label 2 are shown in white and by label 1 in gray. Broken lines are (ambiguous) boundaries carrying both labels.

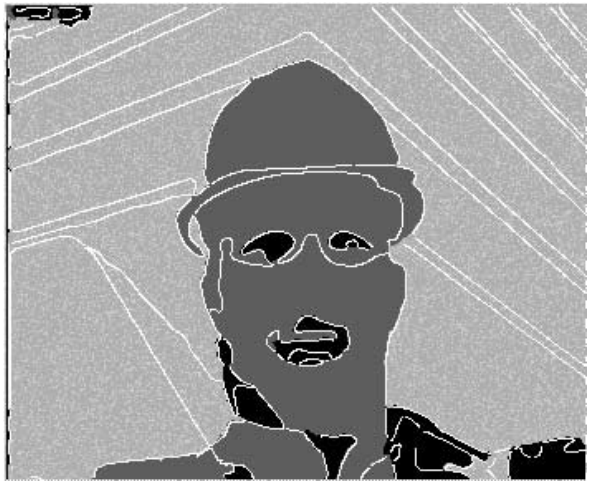

Fig.2. Region labels derived from motion labels assigned to boundaries of "unicolor" regions. A region can have either one label (the same for any legal labeling) or two labels (depending on the labeling, is associated with to either the object or the background). The regions with a unique background label *1* are shows in light gray. Dark gray regions belong to the object and carry label *2* alone. Black regions have both labels.

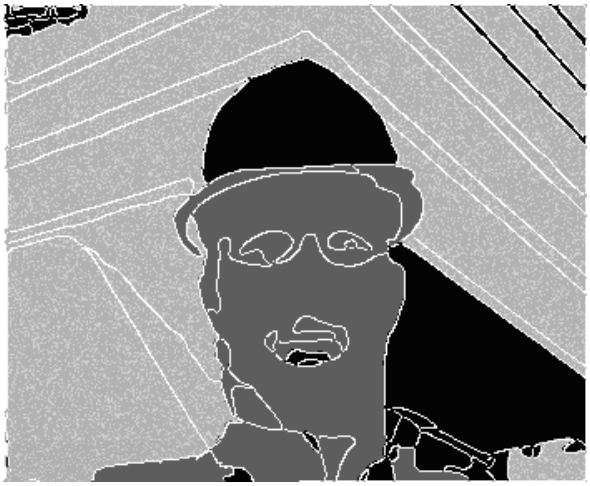

Fig. 3. Two motions are found for each pair of three sequential frames. Each region *A* of the center frame is tested as to what motion better "recalculates" its pixel values (see text for details). Light-gray regions are those marked by label *1*; dark-gray regions are marked by label *2*; and black regions are those where this mechanism of label selection fails to produce conclusive results. It should be noted that labelings here and in Fig. 2 are complimentary and, when combined, produce a nearly perfect segmentation.