Objects and Occlusion from Motion Labeling

Albert Akhriev[1] Alexander Bonch-Osmolovsky[1] Alexander Prusakov[1] Fedor Chelnokov[2] Pavel Agapov[2]

[1] Institute of Information Technologies, RRC Kurchatov Institute,
1 Kurchatov square, 123182, Moscow, Russia.

[2] Moscow Institute of Physics and Technology,
9, Institutskii per., Dolgoprudny, 141700, Moscow Region, Russia.

semantic.segmentation@coollist.com

ABSTRACT

The problem of segmenting color video sequences is addressed. Boundary motion and occlusion relations expressed by labeling rules are argued to be of key importance for segmentation. Interframe motion of object(s) and background is approximated by a similarity or affine transforms. The computation proceeds in several steps: (1) color gradient edges are fuzzy-clustered according to their motion; (2) boundaries of regions formed by a color segmenter are assigned weights and labeled according to their motion; (3) un-ambiguously moving regions and "conflict" regions are identified by a label relaxation procedure; (4) ambiguities and conflicts are resolved by multiframe analysis.

Keywords

Semantic segmentation, motion fuzzy clustering, motion labeling, occluding boundaries, multiframe tracking.

1. INTRODUCTION

Segmenting frames into coherently moving regions is an important issue in video sequence analysis. There are two main sources of motion information: (1) the behavior of boundaries and (2) the local color and intensity changes between frames (optic flow). In many situations, the behavior of boundaries is sufficient for object separation, which is readily exemplified by flat-color animations. Natural images may also have low-texture and almost constant color regions, where optic flow data may not be trusted.

In this paper, we formulate and develop an original "semantic" approach to video segmentation (splitting frames into moving objects and the background) based on the analysis of boundary motion and occlusion relations among boundaries. It is semantic in the sense that a physically meaningful entity – the occluding boundary – plays a central part. Our research was instigated by the challenge posed by cartoon animations to optic flow-based methods. The distinctive feature of our approach is that it heavily relies on color segmentation to turn any frame into a cartoon. We assume that differently moving objects (and the background) belong to different depth layers and there are no instances of mutual occlusion.

First, a fuzzy clustering algorithm is applied to reliable edge points to obtain their tentative partitioning into motion groups (in terms of fuzzy membership coefficients) and calculate motion parameters for each group. Next, region boundaries produced by the 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. These weights are thresholded and the surviving labels are fed into a label relaxation procedure. This procedure takes into account occlusion relations between regions and their boundaries and is applied to classify frame's regions either as unambiguously belonging to a single motion (depth) layer or ambiguous. The output of boundary motion analysis is complemented by the analysis of how well internal region pixels are recalculated by the obtained motions. Finally, the two labelings are fused and fed into a multiframe region tracking algorithm, which also exploits the occlusion data.

A similar approach to the motion segmentation problem was presented in [Smi00a] and [Smi00b]. How-

Journal of WSCG, Vol.12, No.1-3, ISSN 1213-6972
WSCG'2004, February 2-6, 2003, Plzen, Czech Republic.

ever, as will be clear, the method we use to solve this problem differs in many important respects. Some important details, omitted here for brevity, can be found on our website:
http://www.topazk.ru/ssl/index.htm.

2. CALCULATION OF MOTION

Our motion calculation procedure was described elsewhere [Akh02a], and so only the key points will be briefly outlined. First, an original color segmenter (developed by P. and D. Nikolaevs) is applied to each pair of consecutive video frames. The segmentations in each frame are represented as adjacency graphs with regions as nodes and their *boundaries* as edges. We refer to points on these boundaries as *boundary points*. The chains of pixels extracted by an edge detector (based on ideas outlined in [Can86a], [Sap96a], and [Zen86]).will be called *edge segments*

Because edge segments are easier to rate in terms of strength and localization than boundary segments, the former are used in fuzzy-clustering [Bez81a] by motion, yielding several (2-3, typically) linear transforms $\mathbf{T}_c$ that describe the motion in a scene. Then we calculate the residual d_{ci} for each *i*th boundary segment as a mean distance between its boundary points shifted by *c*th linear transform $\mathbf{T}_c$ and the corresponding boundary points in the next frame. The smaller d_{ci} the better association between the *i*th boundary segment and the *c*th motion cluster.

3. MOTION AND DEPTH LABELING

The motions of objects and their boundaries are supposed to satisfy two constraints: (1) objects with similar motions belong to the same depth layer; (2) in each layer, the interframe motion can be reasonably well approximated by the adopted model, e.g., the similarity transform. 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 depth labels to all regions in agreement with motions (labels) of their boundaries and with minimal ambiguity.

In what follows, we shall focus on the practically important case of two motions. Although we have investigated the case of multiple motions, the lack of space does not allow us to consider this case in depth.

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 (see [Akh02a]), the first cluster most often corresponds to the background motion. To ensure correct motion ordering, a validation procedure was developed (section 5). Label **2** is then 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*. The obtained label sets will then be purged and the labels will be propagated from boundaries to regions. The purging (relaxation) procedure is based on the three following rules and their implications:

Rule (1): the boundary of two regions is part of and moves together with the region closest to the viewer;
Rule (2): all common boundaries of any two adjacent regions with labels m_i and m_j must be labeled by either m_i or m_j;
Rule (3): a region is never to be assigned a label other than that of its boundaries.

Rule (1) is identical to that formulated in [Smi00a] and is *physical* by its nature. Rule (2) simply states that motions are uniquely associated with depth layers. Rule (3) is introduced merely to eliminate the otherwise irreducible ambiguity between object parts and holes. It can be shown that any depth assignment and motion labeling of regions (and, thereby, boundaries) consistent with these rules is physically realizable. Such labelings will be called *legal labelings*.

The label relaxation can be accomplished by different methods. In [Smi00a] and [Smi00b] the overall solution probability was maximized by simulated annealing. In our approach, *the priority is to identify unambiguously labeled regions rather than find the most probable labeling solution*.

Define the normalized weight as the relative compatibility of the *i*th boundary with the *c*th motion

$$w_{ci} = \left(1/d_{ci}\right)\Big/\sum_{k=1}^{2}\left(1/d_{ki}\right), \qquad (1)$$

where the residuals d_{ki} were introduced in section 2. For each boundary *i*, only such labels *c* with $w_{ci} > 0.33$ are retained. This threshold can be adjusted for better results. Unfortunately, in real videos, many boundaries, especially short ones, will carry both labels (Fig. 1). The remaining labels for each boundary are entered in a list of valid labels L_i. Our purpose is *not* to find all legal (i.e., satisfying Rules (1)–(3)) or most probable labelings, 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 labeling. This problem can be solved by an efficient sequential algorithm. The algorithm can be generalized to handle the case of more than two labels (motion clusters)..

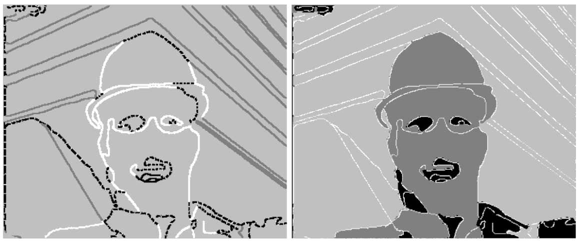

Figure 1. *Left.* **Each calculated motion is applied to all region boundaries in the 1st frame. A boundary is assigned a label *c* if, when displaced by the *c*th motion, it aligns reasonably well with a suitable boundary in the 2nd frame. Boundaries uniquely marked by label 2 are shown in white and by label 1 in gray. Black lines are ambiguous boundaries carrying both labels.** *Right.* **Region labels derived from motion labels assigned to boundaries of constant-color regions. A region can have either one label (the same for any legal labeling) or two labels (depending on the labeling, is attributed to the object or the background). The regions with a unique background label 1 are shown in light gray. Dark gray regions belong to the object and have a single label 2. Black regions have both labels.**

The *labeling algorithm* involves no iterations and is accomplished in three steps.

Step 1: Scan all regions and find those with at least one boundary such that its label list L_i = {**1**} contains a single label **1**, and assign it to these regions.
Step 2: Scan all regions and find those with at least one boundary such that its label list L_i = {**2**} 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.**
Step 3: All regions not labeled in the two preceding steps are assigned both labels.

This algorithm can be proved to solve the stated problem whenever a legal labeling exists. The proof (omitted for brevity) is based on the following three statements

1. A region given a single label after *Steps 1* and *2* cannot carry any other label in any legal labeling.
2. If, on *Step 2,* some region gets label **2** but on *Step 1* it was already given label **1,** then no legal labeling exists.
3. Every label assigned to a region in *Step 3* serves as this region's label in at least one legal labeling.

Recall that label **1** corresponds to the background and label **2** to the occluding object. If so, Rule (1) implies that, in a legal labeling, a boundary label must equal the larger label of the two adjacent regions.

In our implementation of the algorithm, if any conflict region (labeled **1** in *Step 1* and **2** in *Step* 2) is found, the process does not terminate and both labels are assigned to this region. This is justified because the boundary motion may have been inaccurately computed, especially for short boundaries, and the threshold for label selection in (1) may be not right. Despite the absence of a legal labeling, this method was found to often lead to reasonable results. It should be noted that the number of conflict regions can be used as a test of correct depth ordering, i.e., that layer 2 indeed occludes layer 1. Fig.1 shows an example of an initial boundary labeling and the resulting region labeling.

4. REGION MOTION ANALYSIS

It often happens that some regions get both labels {**1**,**2**}. The ambiguity is generally caused by (a) local similarity of different motions, (b) faults of the color segmenter; and (c) the presence of short boundaries whose motion is hard to reliably determine. This ambiguity can be effectively resolved in video sequences with sufficient texture by applying the obtained motions to the inner pixels of regions in the original (unsegmented) image. In this section we consider region-based analysis in the case of just two motions.

Let region *R* be assigned both labels on the preceding step, and consider three successive frames of a sequence. It can be claimed that nearly all points of the middle frame will be visible in one of the adjacent frames. An occlusion-tolerant interframe difference with respect to the motion $\mathbf{T}_c$ can be defined as

$$D_{min}(\mathbf{p},c)=\min\Big(\left\|\mathbf{C}_{t+1}(\mathbf{T}_c\mathbf{p})-\mathbf{C}_t(\mathbf{p})\right\|,\ \left\|\mathbf{C}_{t-1}(\mathbf{T}_c^{-1}\mathbf{p})-\mathbf{C}_t(\mathbf{p})\right\|\Big) \qquad (2)$$

where *c* is the motion label, $\mathbf{C}_t(\mathbf{p})$ is the color of a pixel **p** in frame F_t, and $\|.\|$ is the norm in the color space. For region *R* and motion *c*, one can define an occlusion-tolerant measure of interframe color difference

$$D_{med}(R,c)=\operatorname*{median}_{\mathbf{p}\in R}\left(D_{min}(\mathbf{p},c)\right). \qquad (3)$$

Our plan is to generate region-based motion labels for the selected region *R* based on D_{med}. The weights are calculated similarly to those in the boundary label assignment (1)

$$w_{cR}=\left(1/D_{med}(R,c)\right)\Big/\sum\nolimits_{k=1}^{2}\left(1/D_{med}(R,k)\right). \qquad (4)$$

Here, w_{cR} is a measure of compatibility between the motion of region *R* and the *c*th motion group. For each ambiguous region *R,* only those labels *c* are retained for which w_{cR} exceeds a predefined threshold (0.43, in our experiments). A region gets both labels, if both values D_{med} in (4) are smaller than a fixed multiple of the noise level.

The labels produced by this region-based procedure are fused with those yielded by the boundary motion analysis using a straightforward logic: *defined* + *defined* = *defined*; *defined* + *ambiguous* = *defined*; *defined* + *contradiction* = *ambiguous* and fed as input to the multiframe tracking algorithm.

5. RECOVERING LAYER ORDERING

Given two motions $\mathbf{T}_1$, $\mathbf{T}_2$, and two successive frames F_t, F_{t+1}, we now describe how to determine which motion is a foreground one (occluding), and which motion is a background (occluded) one.

In the first method, both orderings variants were tried: regions undergoing motion $\mathbf{T}_1$ occlude those undergoing motion $\mathbf{T}_2$ and vice versa. The correct ordering was found as that leading to a legal labeling with the highest possible threshold on the weights w_{ci} in (1). In general, the larger threshold, the less uncertainty of label assignment. This method was implemented and tested along with some modifications but turned out to be not very stable.

The second method directly weighs layer orderings. First, let us introduce the notation. Let C and N be the sets of points in the current, F_t, and the next frame, F_{t+1}, respectively., and let $\mathbf{T}_1$, $\mathbf{T}_2$ be the calculated global motions in the scene. Denote by C_1 and C_2 the sets of points in the current frame having a match in the next frame under the respective motion: $C_k = \{\mathbf{p} \in C : I_2(\mathbf{T}_k(\mathbf{p})) \approx I_1(\mathbf{p})\}$, k = 1, 2, where I_1, I_2 are the intensity (or colors) values at points of the current and the next frames, respectively. The points in the current frame that are not visible in the next frame make up the set $H_C = C \setminus (C_1 \cup C_2)$. The points of H_C that do move out of the frame will unambiguously belong to the background and will be occluded in frame F_{t+1}. Denote by $N_1 = \mathbf{T}_1(C_1)$ and $N_2 = \mathbf{T}_2(C_2)$ the images of sets C_1 and C_2 in the next frame and their complement be $H_N = N \setminus (N_1 \cup N_2)$. Therefore, H_N is the set of points invisible in the current frame and visible on the next one.

Let $\mathbf{p} \in H_C$. If $\mathbf{T}_1(\mathbf{p}) \in N_2$ then the image of point $\mathbf{p}$ by the 1st motion could be occluded by a point undergoing the 2nd motion. Analogously, if $\mathbf{T}_2(\mathbf{p}) \in N_1$ then the image of point $\mathbf{p}$ by the 2nd motion could be occluded by a point undergoing the 1st motion. Fig.2 illustrates the situation.

The correct layer ordering could be inferred from the analysis of what happens with points of H_C and H_N. We count the number m_1 of points $\mathbf{p} \in H_C$ such that the difference $\left\| I_1(\mathbf{T}_2^{-1}(\mathbf{T}_1(\mathbf{p}))) - I_2(\mathbf{T}_1(\mathbf{p})) \right\|$ is small (i.e. $\mathbf{T}_1(\mathbf{p}) \in N_2$) and the number m_2 of points for which the difference $\left\| I_1(\mathbf{T}_1^{-1}(\mathbf{T}_2(\mathbf{p}))) - I_2(\mathbf{T}_2(\mathbf{p})) \right\|$ is small (i.e. $\mathbf{T}_2(\mathbf{p}) \in N_1$). If $m_1 > m_2$, then the 2nd motion is the foreground (occluding) one and vice versa. To increase the number of points participating in the analysis, we interchange frames F_t, F_{t+1} and consider the reversed motions, i.e. points $\mathbf{p} \in H_N$ are used and those $\mathbf{T}_1^{-1}(\mathbf{p}) \in C_2$ and $\mathbf{T}_2^{-1}(\mathbf{p}) \in C_1$ are tested. Naturally, only the points that remain inside the frame under both motions are counted.

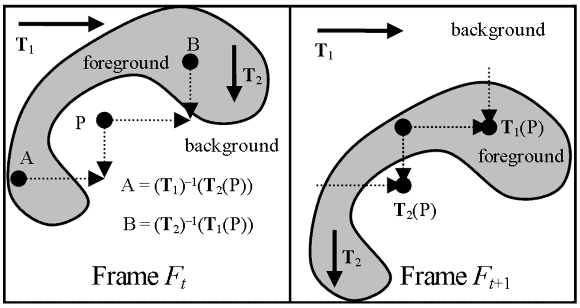

Figure 2. The case of two motions. Background point *P* in the first frame is occluded in the next frame. Analyzing the relations of *P* with points *A*, *B*, $\mathbf{T}_2$(*P*), and $\mathbf{T}_1$(*P*), we can conclude which motion of the two is the foreground (occluding) one.

When does the second method work? The overbalance in favor of a particular motion results from the "hiding" background points ($\mathbf{p} \in H_C$) such that, e.g., $\mathbf{T}_1(\mathbf{p}) \in N_2$, but $\mathbf{T}_2(\mathbf{p}) \notin N_1$. The latter means that $\mathbf{T}_2(\mathbf{p}) \in H_N$, because $\mathbf{T}_2(\mathbf{p}) \notin N_2$ since $\mathbf{p} \in H_C$. In the given example, the outcome of the method is determined by those points of the current frame F_t that are "hidden" by the 1st motion and mapped on the set of newly opened points in the next frame F_{t+1} by the 2nd motion.

We omit for brevity the description of fairly straightforward statistical tests used to evaluate the differences and classify points belonging to the sets C_1, C_2, N_1, or N_2.

It is no surprise that, when the motions are small or almost similar, or the object boundary is noisy and there is little color contrast between the object and the background, the second method might not produce a conclusive layer ordering. Better results can be obtained by a combination of both described methods. As an alternate, we are currently investigating a multiframe ordering validation technique, but this is a topic of a separate paper.

6. MULTIFRAME TRACKING

All previously described methods analyze a pair of successive frames for two motions. As the result, some regions remain ambiguous. Fortunately, it often happens that ambiguous regions in one frame correspond to unambiguously labeled regions in other frames. The ambiguity of sequence segmentation could be greatly reduced or even eliminated by coordinating the solutions over several successive frames.

In image sequences with smooth enough motion, the region area does not normally change much. Therefore, simple transformation models (Euclidian, similarity, or affine) are good candidates for tracking.

Consider a region A in frame F_t with an area S_A, a region B in frame F_{t+1} with an area S_B, and a motion transformation $\mathbf{T}_c$.. The *overlapping ratio* of the two regions A and B under the motion c is defined by

$$\eta^c_{AB} = S_{A\cap B} / \min\left(S_A, S_B\right). \tag{5}$$

The *nesting ratio* of the two regions A and B under the motion c is defined by

$$\rho^c_{AB} = S_{A\cap B} / \max\left(S_A, S_B\right). \tag{6}$$

Definition 6.1. Regions A and B are called *corresponding* under the cth motion, if the overlapping ratio exceeds some predefined threshold.

There are no theoretical grounds for the choice of the threshold, mentioned in the above definition, apart from experimental observations. It can be as high as 0.8, if one wishes to preclude wrong correspondence, or as low as 0.1, if a procedure is available to detect and undo wrong matches. The splitting algorithm (see below) might be useful for automatic thresholding.

For simplicity, in this section we again consider the case of two motions, and assume that objects labeled by **2** occlude objects labeled by **1**. As previously described, all possible occlusion orders are tested and the best one is adopted. The following rule reconciles labels in successive frames. If, under motion **2**, a region A marked by **2** in frame F_t corresponds to a region B in frame F_{t+1}, then B cannot be labeled by **1** (provided we have sufficient confidence in this correspondence relation, despite the fact that the color segmentation could be unstable). Indeed, since motion **2** is the foreground one, background cannot occlude an object undergoing this motion. Based on this rule, the region labels can be modified:

(1) If the label list of a region A consists of a single label **1** and A corresponds under motion **2** to a region B having two labels, and B does not correspond to any other region labeled by **2** alone, then label **2** can be removed from the B's label list.

(2) If the label list of a region A consists of a single label **2** and A corresponds under motion **2** to a region B having two labels, and B does not correspond to any other region labeled by **1** alone, then label **1** can be dropped from the B's label list.

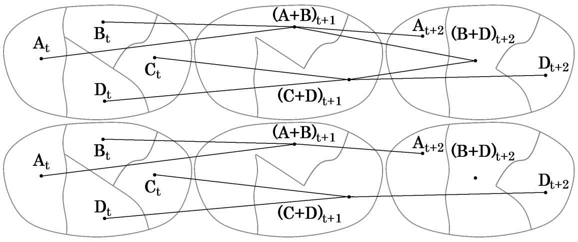

Figure 3. Regions can be combined into a collection by tracing over multiple frames. An object occurring in three successive frames is shown. Its regions can be combined either in the single collection (top diagram) or in the two separate collections (bottom diagram) depending on the threshold.

To extend region tracing from just a couple of frames and regions to the entire video sequence and groups of ambiguous regions, we need the notion of a *collection* of regions:

Definition 6.2. A *collection* **C** is a set of regions in several frames such that any two its regions $A\in \mathbf{C}$ and $B\in \mathbf{C}$ can be connected by a chain of regions belonging to **C** such that any two adjacent regions in this chain are in correspondence under the foreground motion.

The adjacency in Definition 6.2 does not imply geometrical proximity but a correspondence between regions of successive frames. Fig.3 explains the concept of a collection. Suppose that a moving foreground object consists of four regions A, B, C and D in the frame F_t. In the next two frames, F_{t+1} and F_{t+2}, the object is randomly split into differently shaped regions because color segmentation instability. If the overlapping ratio threshold (5) is small enough, then all regions could be combined into a single collection, Fig.3, top diagram. If the threshold is large enough, then we obtain two separate collections $\{A_t, B_t, (A+B)_{t+1}, A_{t+2}\}$, $\{C_t, D_t, (C+D)_{t+1}, C_{t+2}\}$, Fig.3, bottom diagram.

Definition 6.3. A collection is called *exhaustive,* if no new region can be included without violating the collection definition.

We generate all possible exhaustive collections of corresponding regions, and it can be readily seen that each region in each frame will participates in one and only one collection (indeed, all regions in a collection correspond under the foreground motion, so, if any region participates in several collections, then the collections could be united). If, apart from ambiguous regions, a collection also happens to include a region with a definite label, then all the regions of the collection will get this label. In this way, a definite label can be propagated through the video sequence. The larger the collection, the higher the chances are that at least one unambiguously marked region will get into it. This is the reason to keep the overlapping ratio threshold (5) as small as possible.

In collections, labels are propagated from definite regions to ambiguous ones, but profuse collections may contain regions with inconsistent labels. Such collections must be divided into smaller ones. Following algorithm splits all inconsistent collections.

1. Get an inconsistent collection that contains regions with definite label **1** and definite label **2**.
2. Create the list L_c of all pairs of corresponding regions sorted in the ascending order by their nesting ratios (6). Pick up the pair with the least nesting ratio (current correspondence).
3. Remove the current correspondence, and see what happens:
 (a) The collection failed to break apart and still contains the same regions; then get the next correspondence in the list L_c to remove and **goto** step 3;
 (b) The collection was broken into two parts such that all regions in the first part are marked by both labels. The removal of this link did not eliminate the inconsistency in the second sub-collection and failed to reduce the ambiguity in the first one. Then restore the connection, get the next correspondence in the list L_c to remove and **goto** step 3;
 (c) The collection was broken into two parts but any or both sub-collections remain(s) inconsistent. Then, for the inconsistent sub-collection(s), recursively repeat this algorithm from the start;
 (d) Assign the definite labels to all regions of newly created sub-collections. **Break** from the loop.

It is clear now why the two coefficients (5) and (6) were introduced. The overlapping ratio characterizes the degree of matching between the smaller region of a pair and the one it corresponds to, whereas the nesting ratio evaluates the degree of matching between the larger region and the second member of the pair. For example, if a large region in the current frame breaks into n smaller parts in the next frame, then the overlapping ratios of all the corresponding pairs will be close to 1. But small regions will probably find false matches, producing inconsistent collections; hence some links must be broken. It is not surprising that we use the overlapping ratio to construct collections and the nesting ratio to divide inconsistent ones.

7. RESULTS AND DISCUSSION

The approach described in this paper is unique (or not very common) in its concurrent and equally important use of the boundary and region motion data. The first component – the computation of region motion from boundary motion – allows it to easily handle images with low-texture regions and cartoon animations, which are intractable for optic-flow-based segmentation algorithms. At the same time, ambiguous, but sufficiently textured, regions will be assigned to the obtained global motion and depth layer by the second, region-based, component.

The combined segmentation results are illustrated in Fig.1 for the case of two frames and in Figs.4–6 for multiframe sequences. One can see how different segmentation mechanisms complement each other.

The motivation and the problem formulation underlying our approach and also the rough partition of the problem into separate blocks (boundary motion analysis, clustering, labeling, label relaxation, and multiframe tracking) turned out to be similar to those in [Smi00a] and [Smi00b]. Like the authors of these papers, we believe that occlusions need not be regarded as an interfering factor and should, instead, be relied upon in image and sequence segmentation.

There are some minor distinctions between our approach to motion grouping and that described in [Smi00a] and [Smi00b]. For example, we use the fuzzy-clustering algorithm to find edge motions [Akh02a] while Smith *et al.* use the EM algorithm. The former, in our opinion, is more mathematically transparent and allows simpler introduction of a new motion cluster (group). We also would like to draw attention to the original and very effective occlusion-tolerant method to test the applicability of a given motion to internal region pixels, based on the analysis of three successive frames.

Our approach to assigning regions to depth layers and motion groups differs essentially from that in [Smi00a], [Smi00b]. As previously explained, we *do not try to find the most probable segmentation solution*, particularly because the probability maximum can be quite flat due to the presence of many ambiguous regions. Instead, we seek to truthfully reveal the degree of uncertainty inherent in the problem in hand and, given image data, try to reduce it by additional means. We do not seek to obtain a single solution when it does not exist in principle. Accordingly, we make no attempt to exploit to the outmost the meager distinctions that might exist in the weights of boundary and region labels. In our approach, a region either has a motion label in its list or not, and we add up boundary labels relatively liberally.

On the other hand, a tentative comparison of motion clustering in [Smi00b] with our results indicates that, on the whole, all reasonable methods of boundary motion evaluation based on a parametric global model yield comparable results. The intrinsic shortcoming of such methods is that essentially different motions are hard to distinguish in certain circumstances. For example, pure rotation and pure translation may locally give rise to similar velocity fields. So, even having two well-defined motion groups, we may fail to reliably refer a given boundary to just one group. Other sources of uncertainty are fairly common boundary shapes – straight lines and circle arcs, which can be almost equally well transformed by different motions.

The algorithms we use to derive lists of admissible edge labels make it possible to analyze the internal computational structure of the problem in hand and split it into separate tasks. It was found that, in the case of two global parametric motions, the set of admissible labels of a region is determined solely by boundary label list of this and the immediately adjacent regions. Therefore, the label tags of other boundaries are not important for labeling the given region. This is not exactly true for the case of three or more motions, but even in this case, as shown by experiments, the "long range" interaction is fairly limited. The latter circumstance makes the labeling problem computationally tractable. The fact that the problem of sequence segmentation from boundary motion could be posed in terms of the graph labeling theory is very fortunate because now it can be treated in conjunction with the problem of finding occlusion boundaries in a single image by the T-junction analysis.

Normally the results are more accurate (and less ambiguous) when a longer image sequence is processed. Our method of region tracking takes into account the obtained depth order and is based on the analysis of what we call multiframe region collections. As a quantitative measure, it uses the overlapping and nesting ratios, which look simplistic. Nevertheless, as shown by our experiments, the procedure works much better than could be expected.

We are not fully satisfied by our determination of the occlusion order. Because we compute motions for each two (or three) successive frames, we have to associate these motions across adjacent frame pairs. The comparison of the number of conflict and ambiguous regions seems to work but not as well as one might wish. We are currently analyzing the applicability of other methods, and have some encouraging results, but these are not incorporated in this work.

Our approach is intrinsically coupled with color segmentation, which can generate various grades of segmentation. The problem is that one never knows if the given segmentation level indeed separates the foreground from the background, i.e. there are no regions that significantly span both the object and the background. We are currently exploring the possibility to use simultaneously several segmentation levels. The popular expectation that the use of global parametric motions may eventually lead to the general solution of the semantic segmentation problem, in our opinion, is not well justified for several reasons:

1. Semantic objects, and especially their parts, quite often fail to move as prescribed by a motion model. The use of more "flexible" models does not solve the problem either, because motion-based segmentation and object segmentation are not basically the same.
2. Motion of large objects is very likely to mask that of small objects in *any* motion clustering procedure.
3. Instances of self-occlusion and mutual occlusion limit the applicability of the layers concept and, therefore, labeling methods.

A more physically sound approach, in our view, would be to couple the segmentation with the analysis of occluding boundaries, making their detection, both static and dynamic, the central point of the method. This can be accomplished by the following means:

1. Estimating local motions and occlusion rather than global ones and expanding the solution over a frame.
2. Detecting occluding boundaries by analyzing the motion of T-junctions and optic flow discontinuities.
3. Integrating motion-based occlusion data with occlusion data derived from T-junction analysis in a single image (static labeling).

8. REFERENCES

[Smi00a] Smith P., Drummond, T., Cipolla, R. Segmentation of Multiple Motions by Edge Tracking between Two Frames. Proc. 11th British Mach. Vis. Conf. (BMVC), vol. I Bristol (2000), pp.342–351.

[Smi00b] Smith P., Drummond, T., Cipolla, R. Motion Segmentation by Tracking Edge Information over Multiple Frames. Proc. 6th European Conf. Comp. Vis. (ECCV), vol. II, Dublin (2000), pp.396-410.

[Akh02a] Akhriev A., Bonch-Osmolovsky A., Prusakov A.: Computation of Motion and Occlusion Relation of Objects from the Motion of Their Boundaries. Proc. Int. Conf. CVPRIP, vol. I Durham (2002), pp.789-792.

[Can86a] Canny, J.: A Computational Approach to Edge Detection. IEEE PAMI 8, pp.679–698, 1986.

[Sap96a] Sapiro, G. and Ringach, D.L.: Anisotropic Diffusion of Multivalued Images with Applications to Color Filtering. IEEE Image Processing 5 pp.1582–1586, 1996.

[Zen86a] Zenzo, S.D.: A note on the gradient of a multiimage. CVGIP 33, pp.116–125, 1986.

[Bez81a] Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1981).

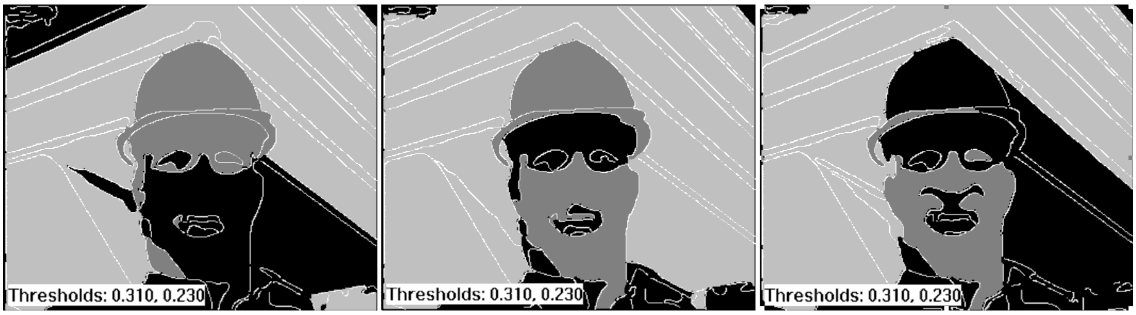

Figure 4. Region motion and order computed from boundary motion for three successive frames. Light gray regions are those unambiguously assigned to the background and dark gray regions are unambiguous object regions. Black regions are ambiguous and, depending on the actual labeling, can be either background or foreground. As explained in text, conflict regions are also treated as ambiguous.

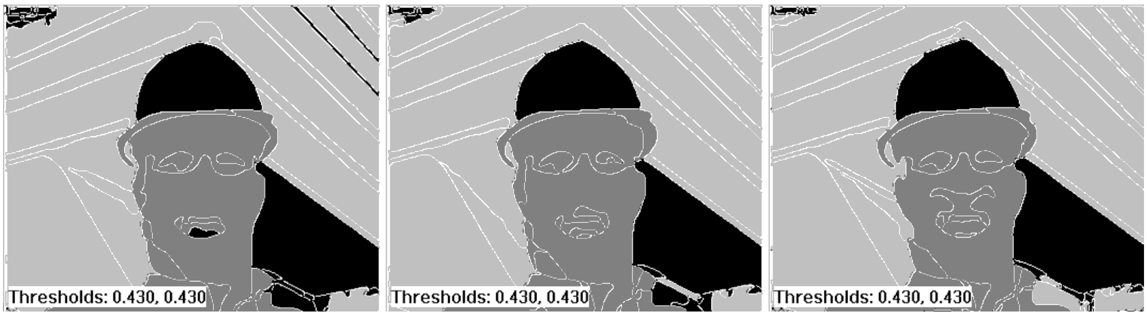

Figure 5. Region assignment to depth layers based on applying the obtained boundary motion to region interior. Compare this to Fig. 4. Light gray regions are those unambiguously assigned to the background; dark gray regions are unambiguous object regions. Black regions remain ambiguous (none of the motions is much better than the other).

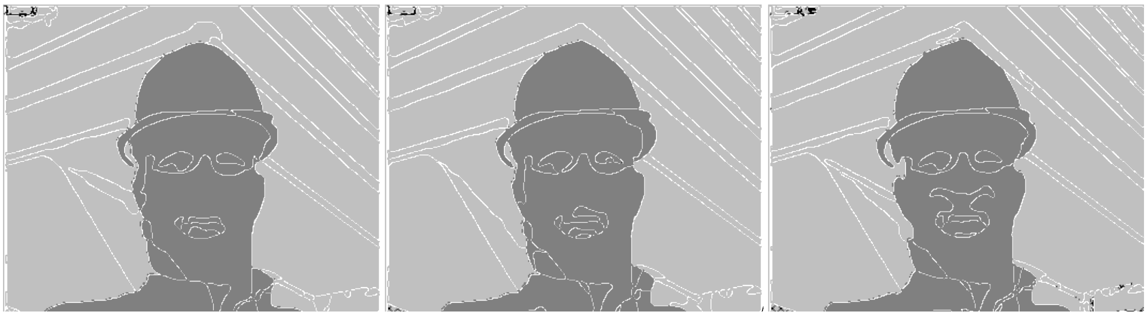

Figure 6. The final multiframe segmentation obtained by the region tracking procedure that takes into account all segmentation results from the previous stages (Figs. 4 and 5). Light gray regions are assigned to the background and dark gray regions belong to the object. Black regions still remain ambiguous.