Contour Flow: Middle-Level Motion Estimation by Combining Motion Segmentation and Contour Alignment

Huijun Di, Qingxuan Shi, Feng Lv, Ming Qin, Yao Lu
Beijing Key Laboratory of Intelligent Information Technology
School of Computer Science, Beijing Institute of Technology
{ajon, shiqingxuan, lvfeng, qinming, vis_yl}@bit.edu.cn

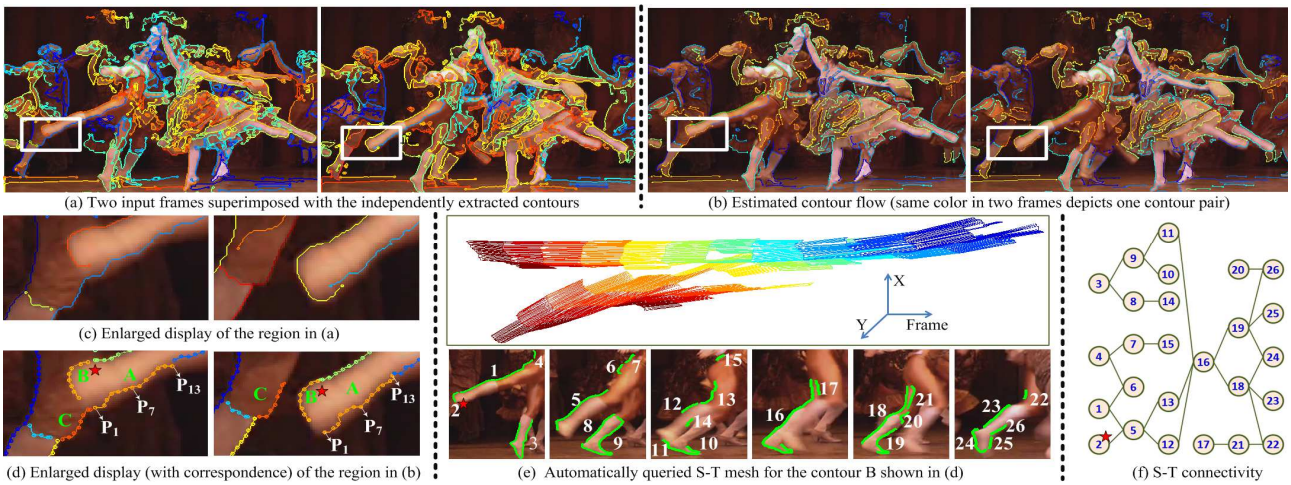


Figure 1: The inputs (a, c), outputs (b, d), and capabilities (e, f) of contour flow, see text for details.

Abstract

Our goal is to estimate contour flow (the contour pairs with consistent point correspondence) from inconsistent contours extracted independently in two video frames. We formulate the contour flow estimation locally as a motion segmentation problem where motion patterns grouped from optical flow field are exploited for local correspondence measurement. To solve local ambiguities, contour flow estimation is further formulated globally as a contour alignment problem. We propose a novel two-staged strategy to obtain global consistent point correspondence under various contour transitions such as splitting, merging and branching. The goal of the first stage is to obtain possible accurate contour-to-contour alignments, and the second stage aims to make a consistent fusion of many partial alignments. Such a strategy can properly balance the accuracy and the consistency, which enables a middle-level motion representation to be constructed by just concatenating frame-by-frame contour flow estimation. Experiments prove the effectiveness of our method.

This work was supported in part by National Natural Science Foundation of China (No. 61003098, 61273273, and 61271374), the Specialized Fund for Joint Building Program of Beijing Municipal Education Commission, and research Fund for Doctoral Program of Higher Education of China (No. 20121101110043).

1. Introduction

General-purpose motion estimation is a long-standing research topic in computer vision, due to its potential difficulties and broad applications. Optical flow estimation (e.g. [1]-[3], to name a few) and point tracking [4]-[7] are two representative methods, which have found their wide usages in motion/video segmentation [8][9], object tracking [10], human pose estimation [11]-[13] and action recognition [7], *etc*. Despite the widespread utility, these methods provide only low level motion information, making further motion analysis or recognition from their results challenging. The related results lack compact yet descriptive information about the shape and the motion of objects in the scene, where optical flow is too dense to explicitly reflect structures, while point trajectories are too sparse or unorganized to represent sophisticated structures.

Aiming at complementing these methods, this paper addresses a problem of contour flow estimation. As shown in Fig. 1a and 1c, the edge pixels in each video frame are linked independently into several contours. And the goal of this paper is to find consistent correspondence of the points on the contours between two consecutive frames (see Fig. 1d). According to motion consistency the original contours from two frames are dismembered and regrouped into coincident ones (see Fig. 1b and 1d). Such coincident contour pairs with point correspondence are called contour

flow in this paper.

Contour flow offers a valuable middle-level motion representation which helps to bridge the gap between high level concepts and low level motion cues. By linking adjacent edge pixels into ordered point sets, image contours possess two important characteristics: ***orderly connectivity*** and ***good boundary localization***. Contour flow further extends these two characteristics into space-time domain:

1) The contour pairs with point correspondence will form meshes, which possess spatiotemporal (S-T) connectivity (see Fig. 1e and 1f). By concatenating the contour flow frame by frame, a compact mesh-based middle-level motion representation can be obtained. There are two distinct advantages of such motion information. One is that the S-T connectivity offers a certain motion grouping. For instance, Fig. 1e shows a queried connected component from the whole S-T connectivity network. *It is worthy of noting that such motion grouping results are just by-products of a motion estimation method which does not pay special attention to motion grouping.* The second advantage is that more sophisticated motion descriptors for recognition can be considered based on the geometric properties of the meshes.

2) As object boundaries are often aligned with image edges, the image contours possess the capability of good boundary localization. With the extension by the contour flow, the achieved mesh-based motion representation can provide spatiotemporal boundary constraints, which could help to improve the accuracy of motion/video segmentation and human pose estimation, or the robustness of object tracking.

However, contour flow estimation is not an easy task. Previous work [14]-[23] had dealt with related problems (see Section 2), but no available practically useful methods can be applied in our problem domain. The key of contour flow estimation is to establish the consistent point correspondence among the inconsistent contours from two video frames under practical conditions such as cluttered environment. A successful estimation of contour flow requires simultaneously tackling two coupled difficulties. One is that the clutter of environment could give rise to highly competitive candidates for point correspondence. It is challenging to design a measurement to locally pick correct correspondence. The other is that the inconsistent linkage of edges in different frames will lead to various contour transitions (such as splitting, merging and branching, see Fig. 1c), which bring troubles in carrying out proper global reasoning to solve the local ambiguities.

By tackling these difficulties, this paper makes two distinct contributions. First, in contrast to the previous work which measures the correspondence based on static image information, we propose a parallel idea for correspondence measurement that uses motion information from optical flow. Our motivation draws from the observation that the motion of a contour locating on a physical surface is usually consistent with the motion of surface's interiors. In this regard, we view contour flow estimation locally as a motion segmentation problem, i.e. picking the right motion label for each contour point from its surrounding motion patterns which are extracted from optical flow field. The success of this idea is on the basis of impressive progress of optical flow. On the contrary, purely relying on static image information makes previous work isolated and brittle under practical conditions.

Our second contribution is that we propose a novel two-staged strategy for global reasoning which can handle the above mentioned difficulties caused by various contour transitions. The goal of the first stage is to obtain possible accurate contour-to-contour alignments, and the second stage aims to make a consistent fusion of many partial alignments from the first stage.

The strength of the proposed strategy is that it can properly balance the accuracy and the consistency. At the first stage, we consider the tentative alignment of only one pair of contours at a time. This helps to make the situation clear, and the accuracy of alignment can be ensured by enforcing tight shape constraints (e.g. to penalize disordering, bending, stretching or shrinking). At the second stage, the contour flow is finally obtained by selecting consistent portions of the alignments from the first stage. The selection is achieved by solving a labeling problem as a whole under loose constraints of piecewise motion continuity.

There are two advantages of such a balance between the accuracy and the consistency. The first one is that sufficient accuracy allows meaningful long-term motion information to be generated by just concatenating frame-by-frame contour flow. The second one is that sufficient consistency ensures stable contour trajectories to be obtained; otherwise only the short ones may be generated. Owing to these advantages, a meaningful and stable mesh-based middle-level motion representation is able to be constructed.

The paper is organized as following. Related work is reviewed in Section 2. The formal problem formulation and contour flow algorithm are presented in Section 3. We dedicate Section 4 to discuss how to concatenate contour flow into mesh-based motion representation and long-term trajectories. Experiments and applications are reported in Section 5, and Section 6 gives the conclusion.

2. Related Work and Discussions

We discuss the related work on motion/correspondence estimation of contours or edges [14]-[23] from two aspects: correspondence measurement to locally score the candidates of point correspondence, and proper global reasoning to solve the local ambiguities.

Previous work had investigated various descriptors for correspondence measurement, such as color gradients at edge pixels [16], image statistics on the two sides of edges

[17][20], curvature [19][21][22] and shape context [18][22][23]. The primary difference between our method and all these work is that we view the correspondence problem locally as a motion segmentation problem, but they take it as a feature matching problem. Experiments show the significant improvements by the correspondence measurement based on motion segmentation, comparing with the ones purely based on static image information.

The work in [3] also views motion estimation as a motion segmentation problem. The difference between their idea and ours is that they combine motion segmentation with optical flow formulation, whereas we combine motion segmentation with contour alignment. In their work, motion segmentation provides solution space for optical flow formulation. While in this paper, motion segmentation provides correspondence measurement for contour alignment, and our data term can be purely based on the motion cues from pre-estimated optical flow field. However our results are not the reproductions of the motion information from the optical flow field. Experiments provide the evidence that contour alignment could help to resist certain noises in the flow field.

In terms of how global reasoning is carried out, related methods can be roughly classified into three categories:

1) Motion field estimation along contours, where smooth motion field along contours is inferred from locally determined partial motion information such as perpendicular components [14] or motion vectors with 1D uncertainty [15]. These methods are dedicated to tackle the long-standing aperture problem, with solid theoretical analysis [14] or impressive results [15]. However, the locally determined motion component could be error prone under cluttered environment, which would in turn affect the inferred complete motion field.

2) Motion estimation of contours/edges based on global transformations (such as affine or similarity) [16]-[18], where consistent correspondence of several contours/edges are reflected in the estimated motion transformation shared by them. Although the global transformations can simplify the process of motion estimation, it could not handle local non-rigid deformations, and could cause error accumulation for long term non-rigid motion estimation.

3) Contour alignment [19]-[22], where the smoothness constraint of contour's deformation can help select optimal correspondence along contours from locally scored matching candidates. The methods in this category are most related to our work. However our distinct characteristic is that we can obtain global consistent correspondence among a set of contours under various transitions such as splitting, merging and branching. The work in [19]-[22] consider only the problem of independent motion/correspondence estimation for each contour, where the obtained correspondence among the contours may be inconsistent or even conflicted with each other. Although contour transitions are explicitly handled as a post-process in [19], they consider only joining the contour fragments, but do not carry out a global optimization of the correspondence.

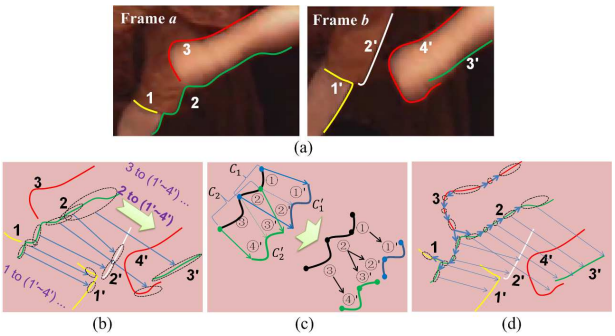


Figure 2: An overview of our contour flow algorithm. **(a) Input contours**: the input contours in two frames may be inconsistent. To handle various contour transitions, our algorithm consists of three steps which are illustrated in (b)~(d). **(b) Contour to contour alignment**: for each contour in frame *a*, to find its all possible partial alignments in frame *b*. **(c) Confliction analysis**: for each contour in frame *a*, by analyzing how its partial alignments are overlapped, break it into several fragments the alignments of which do not conflict with each other. For instance, for the black contour shown in (c), its two partial alignments ($C_1 \sim C_1'$ and $C_2 \sim C_2'$) are overlapped. After confliction analysis, the black contour is broken down into three non-overlapped fragments which are ② $= C_1 \cap C_2$, ① $= C_1$-②, and ③ $= C_2$-②. **(d) Global optimization**: to obtain globally consistent contour flow, the fragments from the previous step will serve as the target nodes of global optimization, and the goal is to solve a labeling problem under a tree neighborhood structure of the nodes.

3. Contour Flow Algorithm

An overview of our algorithm is shown in Fig. 2. Given an input video sequence, the detected edges in each frame are linked locally into several image contours (see Fig. 2a). We do not assume any sophisticated method for edge linkage. A simple edge linking procedure is applied. Adjacent edge points that roughly lie on a straight line are linked firstly into line segments. Then these line segments together with remaining edge points are further linked into the contours for our inputs. Each contour is represented as an ordered set of edge points.

As shown in Fig. 2b, given the contours in two video frames (say frame *a* and *b*), we first consider a sub-problem of contour-to-contour alignment (from *a* to *b*) under tight shape constraints such as the penalties of disordering, bending, stretching or shrinking. By performing such a tentative alignment for all possible contour pairs inside a certain search range, we can obtain a pool of hypotheses for contour flow, which contains all possible pieces of correspondence under various contour transitions.

However, some flow hypotheses may conflict with each other, e.g. their overlapped portion at frame *a* may have different correspondence at frame *b* (see Fig. 2c). Based on conflicting information among the flow hypotheses, we

break the contours in frame a into the fragments the alignments of which do not conflict with each other (see Fig. 2c). These fragments will be the targets to solve a labeling problem as a whole (see Fig. 2d), where global consistent flow hypotheses are selected from the pool under loose constraints of piecewise motion continuity. The contour flow is finally generated by relinking the fragments that are adjacent in both frames.

In the following, we will first discuss in Section 3.1 the correspondence measurement which is used by contour-to-contour alignment in Section 3.2. The solution to flow optimization is covered in Section 3.3.

3.1. Correspondence Measurement

To apply motion segmentation for correspondence measurement, we first extract a set of local motion patterns which are about the motions of small portions on physical surfaces. Then the correspondence of each contour point can be measured by the extent of the consistency between the motion reflected by the correspondence and the local motion patterns near that point.

A small overlapped sliding window (e.g. with size of 20 pixels and 10 pixels overlapping) is scanned over the optical flow field, where the local motion patterns are identified independently in each window. We use similarity motion transformation to extract the motion patterns. RANSAC algorithm is applied to estimate the parameters of the similarity transformations. When one motion pattern is found, we remove the inlier points which are sufficiently close to the current motion pattern. Then, the remaining points are used to find other motion patterns. This procedure is repeated until no point left.

Given one contour point (say p) in frame a, its surrounding motion patterns within a circular region of radius R_{mo} are used to measure the quality of its correspondence candidate (say q) in frame b. The measurement is based on local motion segmentation:

$$D_{Mo}(p,q) = \min_k \|T_k(p) - q\|, \quad (1)$$

where T_k is the similarity motion transformation of a motion pattern surrounding p. The min operation in the equation means that how the correspondence can be best explained by the local motion patterns. In other words, it looks for one portion of the physical surfaces whose motion can best support the correspondence.

3.2. Tentative Contour-to-Contour Alignment

Given the contours in frame a and b, we carry out a tentative contour-to-contour alignment for all possible contour pairs inside a certain search range R_{srh} (more specifically, the contour pairs with at least two points p and q where the distance $D_{Mo}(p, q) \leqslant R_{srh}$). By such alignments, we can obtain a pool of hypotheses of contour flow which contains all possible pieces of correspondence under various contour transitions.

Suppose we are aligning a contour C_a at frame a to a contour C_b at frame b, where the two contours are defined as ordered point sets $C_a=\{p_i; i=1, 2, \ldots, N_a\}$ and $C_b=\{q_j; j=1, 2, \ldots, N_b\}$ respectively. The primary goal of the alignment is to find an optimal correspondence assignment m_i for each point p_i in C_a w.r.t. the points in C_b. When $m_i=j$, it means that p_i is aligned to q_j. As two contours might not be aligned completely, partial alignment must be considered. Therefore, a binary variable v_i is introduced, which indicates the visibility of p_i in frame b. When $v_i=0/1$, it indicates that the correspondence of p_i cannot/can be found. As we are aligning two ordered point sets, the ordering can be useful constraint to enable consistent correspondence. To handle possible descend or ascend alignments of two contours, another binary variable o_i (takes 1 or -1) is defined for each point p_i in C_a. When o_i = -1, it implies that the point pair $\{p_{i-1}, p_i\}$ is aligned to C_b in reversed order, i.e. $m_i<m_{i-1}$; otherwise the vice. The ultimate goal of the alignment is to obtain an optimal state $U_i \equiv \{m_i, v_i, o_i\}$ for each point p_i, such that C_a can be best aligned to C_b under tight shape constraints.

The energy function of the alignment can be generally written as

$$E(U) = \sum_{i=1}^{N_a} D(U_i) + \sum_{i=2}^{N_a} S(U_{i-1}, U_i), \quad (2)$$

Where $D(U_i)$ is a data term and $S(U_{i-1}, U_i)$ is a piecewise smoothness term. The optimal alignment can be obtained by minimizing $E(U)$ w.r.t. U. $D(U_i)$ measures the quality of the correspondence defined by U_i, where the motion based measurement $D_{Mo}(p, q)$ can be used. More specifically,

$$D(U_i) = \begin{cases} \lambda \frac{D_{Mo}(p_i, q_{m_i})}{\sigma_{Mo}} + (1-\lambda) \frac{D_{Im}(p_i, q_{m_i})}{\sigma_{Im}}, & if\ v_i = 1, \\ \xi, & otherwise \end{cases} \quad (3)$$

where we also consider the use of image-based measurement $D_{Im}(p_i, q_{m_i})$ for the comparisons (see experiment for more details), and λ ($0 \leq \lambda \leq 1$) controls the contribution of the two types of measurements. σ_{Mo} and σ_{Im} are used for metric scaling. The constant ξ means that the measurement from data is ignored when p_i is invisible in frame b (i.e. $v_i=0$). ξ can be viewed as a local threshold to determine v_i from data term. When the data term for all possible values of m_i is greater than ξ, locally minimizing $D(U_i)$ will give the decision of $v_i=0$. In the sense of giving the same decision of v_i, increasing the value of σ_{Mo} and σ_{Im} is equivalent to the increasing of the value of ξ. Therefore, among these parameters, there's one free parameter that can be fixed (e.g., ξ can be treated as a constant). One can tune/learn the other parameters for proper thresholding.

The smoothness term $S(U_{i-1}, U_i)$ can be designed as

$$\begin{aligned} S(U_{i-1}, U_i) = {} & S_{co}(m_{i-1}, m_i, o_i, v_{i-1}, v_i) + \\ & S_o(o_{i-1}, o_i) + S_{cs}(m_{i-1}, m_i, v_{i-1}, v_i) + \\ & S_{cb}(m_{i-1}, m_i, v_{i-1}, v_i) + S_v(v_{i-1}, v_i), \end{aligned} \quad (4)$$

where S_{co} is an order constraint term, S_{cs} is a scale constraint term, and S_{cb} is a bending constraint term. These three terms ensure the smoothness of the correspondence of the point pair p_{i-1} and p_i, by penalizing disordering, stretching or

shrinking, and bending, respectively.

The order constraint term S_{co} is defined as

$$S_{co}(m_{i-1}, m_i, o_i, v_{i-1}, v_i) = \begin{cases} \infty & , if\ v_{i-1} \cdot v_i \cdot o_i \cdot (m_i - m_{i-1}) < 0 \\ 0 & , otherwise \end{cases}. \quad (5)$$

When p_{i-1} or p_i is invisible, the order constraint will not be considered (i.e. S_{co} takes zero when v_{i-1}=0 or v_i=0). And disordering will be forbidden when the correspondence exists, that is S_{co} will take ∞ if the correspondence is not in the expected order (o_i = 1 but m_i<m_{i-1}, or o_i = -1 but m_i>m_{i-1}).

Although S_{co} describes the order constraint on m_{i-1} and m_i, it totally relies on the value of o_i. In other words, S_{co} alone cannot provide the expected order constraint. Given any values of m_{i-1} and m_i, one can freely find a certain value of o_i that makes S_{co} zero. Therefore additional constraint of o_i is required and S_o in (4) plays such a role. S_o is a continuity constraint term of the order, and defined as

$$S_o(o_{i-1}, o_i) = \begin{cases} 0 & , if\ o_{i-1} = o_i \\ \alpha & , otherwise \end{cases}. \quad (6)$$

Together with (5), by minimizing the energy in (2), one can obtain proper order information o_i as well as enforce order constraint on m_{i-1} and m_i.

To penalize large stretching or shrinking, the scale constraint term S_{cs} in (4) is defined as

$$S_{cs}(m_{i-1}, m_i, v_{i-1}, v_i) = \begin{cases} f\left(\frac{\left|q_{m_i} - q_{m_{i-1}}\right| - |p_i - p_{i-1}|}{|p_i - p_{i-1}|}\right) & , if\ v_{i-1} = v_i = 1 \\ \xi & , otherwise \end{cases}. \quad (7)$$

The extent of stretching or shrinking is measured by the ratio of change of the distance between the (i-1)th and the i^{th} points under the correspondence m_{i-1} and m_i. The function f imposes the related penalty, given the signed ratio of change (positive value for stretching and negative value for shrinking) as argument. One can choose suitable f to enforce desired penalty. A simple f can be

$$f(x) = \rho \cdot |x|, \quad (8)$$

which is a linear penalty w.r.t. the ratio of change. The constant ξ (i.e., the ξ in this paper are all the same) is again used in (7) as a local threshold to partially determine v_{i-1} and v_i from the scale term. That is, when all possible correspondence cause large stretching or shrinking (the values of f under possible combinations of m_{i-1} and m_i are all greater than ξ), it will be very likely that p_{i-1} or p_i is invisible (v_{i-1}=0 or v_i=0).

The bending constraint term S_{cb} in (4) penalizes large non-rigid deformation. Although at least three points are required to describe locally the deformation of a contour, a two-point translation-based bending constraint is used here for the sake of simplicity. The S_{cb} is defined as

$$S_{cb}(m_{i-1}, m_i, v_{i-1}, v_i) = \begin{cases} \left|\left(q_{m_i} - q_{m_{i-1}}\right) - (p_i - p_{i-1})\right| / \sigma_T & , if\ v_{i-1} = v_i = 1 \\ \xi & , otherwise \end{cases}. \quad (9)$$

This term will penalize any kind of motion of a contour except for translation. It might be an over restriction, but a larger sigma σ_T can reduce this side effect. The constant ξ in (9) plays a similar role in handling the visibility. The difference between (7) and (9) is that the change of the length $|p_i - p_{i-1}|$ and $\left|q_{m_i} - q_{m_{i-1}}\right|$ is penalized in (7), while the change of the offset $(p_i - p_{i-1})$ and $\left(q_{m_i} - q_{m_{i-1}}\right)$ is penalized in (9).

For the visibility, $D(U_i)$ provides unitary data term for each v_i. S_{co}, S_{cs}, and S_{cb} provide pairwise data term for one pair of v_{i-1} and v_i. The smoothness term of visibility is captured by S_v in (4), which is defined as

$$S_v(v_{i-1}, v_i) = \begin{cases} 0 & , if\ v_{i-1} = v_i \\ \beta & , otherwise \end{cases}. \quad (10)$$

Based on all above definitions, one can minimize the energy in (2) to determine the optimal alignment from contour C_a to contour C_b. And dynamic programing (DP) can be applied to achieve a global optimization efficiently.

3.3. Flow Optimization from Many Alignments

By the contour-to-contour alignment discussed in previous subsection, we can obtain a pool of hypotheses which contains all possible pieces of correspondence under various contour transitions. To solve the confliction among the correspondence in the pool, a final contour flow optimization is considered here.

As shown in Fig. 2c, by analyzing the confliction in the hypotheses pool, one can split the original input contours into several non-overlapping fragments which will serve as the target nodes for contour flow optimization. The goal of contour flow optimization is to solve a labeling problem, by globally selecting consistent flow hypotheses from the pool under loose constraint of piecewise motion continuity.

For the target nodes, suppose we have a set of labels $L \equiv \{L_k,\ k = 1,2,\dots,K\}$, where K is the total number of the nodes and L_k is an integer variable whose domain are the indexes of the correspondence hypotheses of the k^{th} node. The flow optimization is achieved by minimizing following energy

$$E(L) = \sum_{k=1}^{K} D(L_k) + \sum_{(k,k')\in Nbr} S(L_k, L_{k'}). \quad (11)$$

The data term $D(L_k)$ is measured based on the alignment cost E in (2) under related correspondence hypotheses.

The smoothness term $S(L_k, L_{k'})$ is defined on a tree neighborhood structure of the nodes (see Fig. 2d), for the ease of optimization (DP can be applied again). The tree can be obtained by generating a minimum spanning tree from the graph defined on the target nodes with Euclidean distance between nodes as edge weight. The $S(L_k, L_{k'})$ consists of two sub-terms

$$S(L_k, L_{k'}) = S_{Mo}(L_k, L_{k'}) + S_L(L_k, L_{k'}). \quad (12)$$

The term S_{Mo} is about piecewise motion smoothness, which is defined in the similar way with S_{cb} in (9). For two neighboring nodes k and k', the term S_L is designed to select the labels from the same original piece of alignment (the one before split, see Fig. 2c top), as possible as one could.

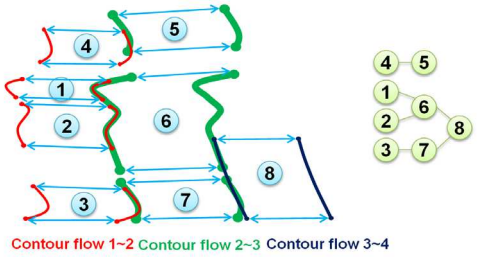


Figure 3: Illustration of contour flow concatenation.

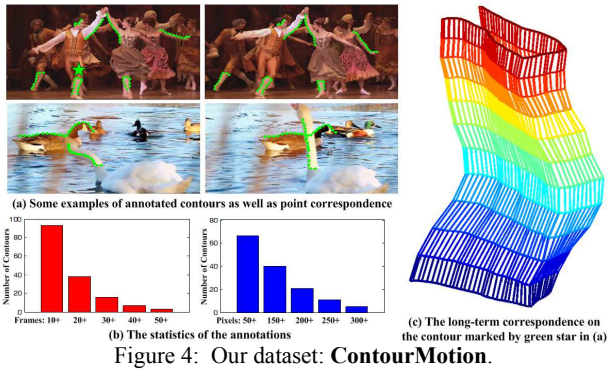


Figure 4: Our dataset: **ContourMotion**.

S_L is defined as

$$S_L(L_k, L_{k'}) = \begin{cases} 0 & , if\ from\ the\ same\ alignment \\ \gamma & , otherwise \end{cases}. \quad (13)$$

By minimizing the energy in (11), the final estimation of contour flow can be obtained, where according to motion consistency the original contours from two frames are dismembered and regrouped into coincident ones.

4. Contour Flow Concatenation

There are two goals of contour flow concatenation: 1) to obtain the S-T connectivity of the contour pairs, 2) to obtain the long-term motion information. The concatenation can be achieved in a very simple way: 1) The S-T connectivity can be obtained by just reading the connectivity information from the frame-by-frame contour flow and connecting the contour pairs (see Fig. 3). 2) For the points on the contours, the long-term motion can be obtained by directly concatenating the frame-by-frame motion information from contour flow. This is similar to the way of generating point trajectories from optical flow [6].

5. Experiments

5.1. Datasets and Parameters Setting

As there is no related public dataset to evaluate the performance of contour flow estimation, we introduce a dataset with the annotations of contours and point correspondence (see Fig. 4). This is also an additional contribution of this paper. Our dataset can be used for other purposes as well, such as the evaluation of motion estimation at object boundaries, or the evaluation of shape registration under practical conditions. We are continuously extending this dataset, and intend to release it to promote related researches.

Currently, there are total 12 image sequences with 100+ contour annotated in the dataset. The image sequences are selected from video segmentation dataset BVSD [24], human pose estimation dataset VideoPose2.0 [11], and action recognition dataset UT-Interaction [25]. Our experiments are carried out on this dataset with fixed parameters for all the image sequences. The parameters setting are: the radius R_{mo} is set as 20 pixels, and the search range R_{srh} is chosen as 5 pixels. The threshold ξ is fixed as 0.6. The sigma σ_{Mo}, σ_{Im} and σ_T are set as 3, 0.5, and 2 respectively. The penalty α, β and γ are selected as 1, 0.4, and 0.3 respectively. The parameter ρ is set as 0.2. In subsection 5.2, λ for calculating $D(U)$ is set as 1, which means that correspondence measurement $D(U)$ is purely based on the motion information from optical flow (generated via the method in [1]). In subsection 5.3, λ is adjusted for quantitative analyses. How to choose above parameters is also discussed in subsection 5.3.

5.2. The Capabilities of Contour Flow

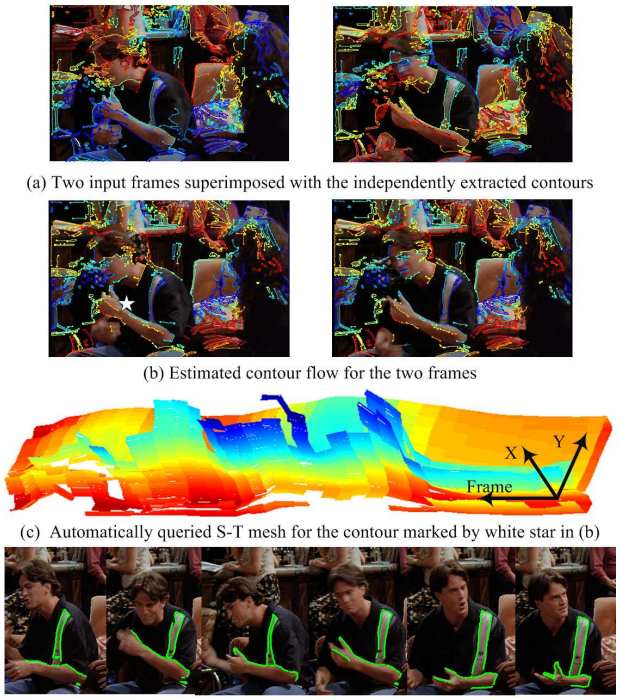
We present visual results of contour flow estimation at three cases (shown in Fig. 1, 5, and 6), to demonstrate the potential capabilities of contour flow for video segmentation, human pose estimation, and action recognition. Given two consecutive frames shown in Fig. 5a, the results of contour flow estimation are shown in Fig. 5b. By comparing contour flow with the inconsistent input contours shown in Fig. 5a, one can verify that quite consistent flows are obtained. If the sub-figures are too dazzled, see Fig. 1c and 1d for an enlarged verification on another sequence.

As shown in Fig. 3, by concatenating contour flow frame by frame, an S-T mesh network can be obtained. Given a contour in that network, we can query its connected component. Fig. 5c shows such a queried component for the star-marked hand contour in Fig. 5b. The hand motion is reflected in the S-T meshes. Fig. 5d exhibits some frame slices with more details of the contours, which can be useful proposals for pose estimation.

In a similar manner, Fig. 6 shows the results on another sequence. The queried meshes shown in Fig. 6 could provide the focus of attention and also help to design rich motion descriptors for action recognition.

5.3. Quantitative Analysis

The quantitative analysis is carried out by comparing related results with the annotated ground truth of the long-term point correspondence on the contours in the dataset (see Fig. 4c). The comparison is accomplished by calculating how much portion of the ground truth can be covered by the results under different error threshold. And the coverage is counted only under sufficient temporal



(a) Two input frames superimposed with the independently extracted contours

(b) Estimated contour flow for the two frames

(c) Automatically queried S-T mesh for the contour marked by white star in (b)

(d) Some frame slices of the S-T mesh shown in (c)

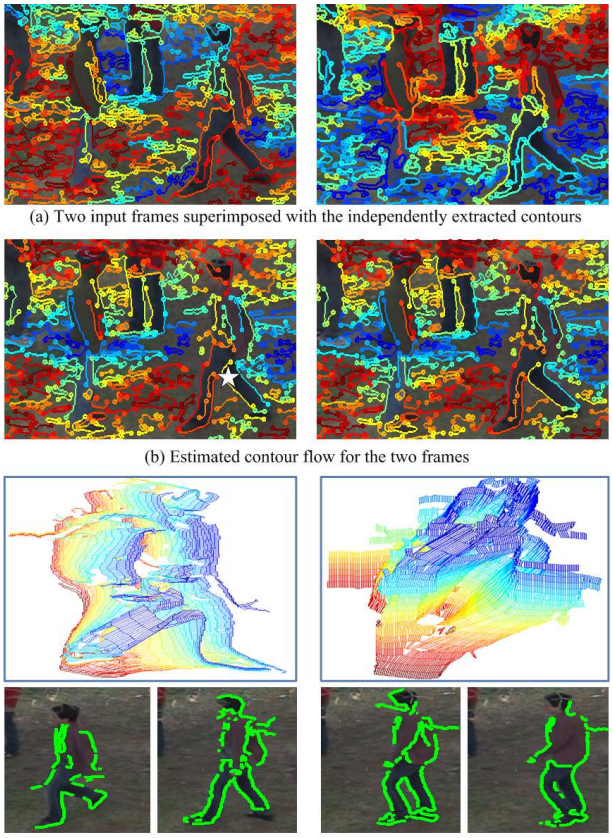
Figure 5: The results of contour flow estimation on a sequence from VideoPose2.0 [11].

overlapping (say 10 frames).

Fig. 7 shows the results of quantitative analysis. We compare the performances of correspondence measurement based on different source of information: motion segmentation only ($\lambda = 1$, CF_{Mo}), static image information only ($\lambda = 0$, gradient [20] CF_{SG} and intensity [17] CF_{SI}), and a combination of motion segmentation and static image information ($\lambda = 0.9$, CF_{Mo+SI}). The significant improvements of CF_{Mo} can be proved, if compared with the ones of CF_{SG} and CF_{SI}. And further improvements achieved by CF_{Mo+SI} can be observed.

Fig. 7 also shows the results obtained without carrying out contour alignment (Median OF, the motion of contours are driven directly by the median-filtered optical flow field), and demonstrates the necessity of contour alignment. This also provides the evidence that the output of our method is not the reproduction of the motion information from the optical flow field, and contour alignment could help to resist certain noises in the flow field.

To justify how the parameters are chosen, we test the impacts of the different parameter settings. The parameter testing is carried out by adjusting one parameter each time. The impacts of the parameters are shown in Fig. 8. The horizontal axis represents different testing, and five different values for each parameter are examined. The values examined are as following: σ_{Mo} and σ_T~ [0.1, 0.5, 1, 5, 7], β ~[0, 0.2, 0.6, 0.8, 1], α~[0, 0.2, 0.5, 2, 4], ρ~[0, 0.1, 0.4, 1, 2], γ~[0, 0.1, 0.6, 1, 2].



(a) Two input frames superimposed with the independently extracted contours

(b) Estimated contour flow for the two frames

(c) Queried S-T mesh and frame slices for the contour marked by white star in (b)

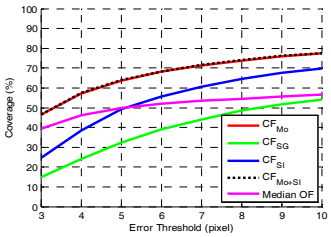
Figure 6: The results of contour flow estimation on a sequence from UT-Interaction [25].

From Fig. 8, one can observe that only two parameters σ_T and α are sensitive. A small value of σ_T could amplify the side effect of the translation motion model. But when the value of σ_T is greater than 1, it will provide positive effect. For the order penalty α, a larger value gives too much restriction. But when the value is less than 0.5, its impact is changed smoothly. Therefore, these two parameters are not necessary to be tuned precisely. And the value of other less sensitive parameters can also be chosen easily.

5.4. Application to Video Segmentation

To demonstrate the benefit from the motion representation based on contour flow, we consider the problem of refining the imperfect results of video segmentation method (e.g. [27], a method with intermediate performance). As our S-T meshes could provide spatial-temporal connectivity and boundary constraint which help to complete and shear the object, our idea is to first select those meshes belonging to the object, and then determine the object's boundary based on the selected meshes.

The mesh selection is formulated as a binary labeling problem, and the goal is to determine whether each point in



Method	Coverage (%)
CF_{Mo}	68.1
CF_{SG}	39.2
CF_{SI}	55.8
CF_{Mo+SI}	68.3
Median OF	51.9

Figure 7: Performance of different methods of motion estimation along contours. **Left**: coverage score under different error threshold. **Right**: coverage score under error threshold 6.

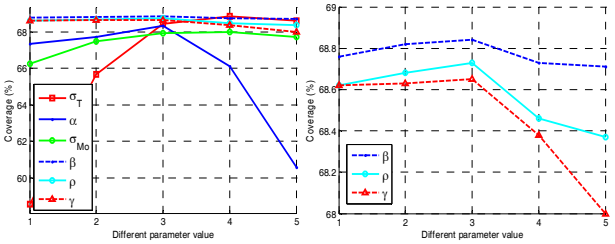


Figure 8: Impact on the performance (measured under error threshold 6) of different parameters. **Left**: all the parameters. **Right**: enlarged view for the three insensitive parameters.

the S-T meshes belongs to the object or background. In order to do this, a set of labels are defined for the mesh points: $L \equiv \{L_t^i,\ i = 1,2,\dots,N_t; t = 1,2,\dots,T\}$, where T is the number of frames in the video, and N_t is the number of the point at frame t of the mesh. The labeling problem is treated as an inference problem, and the distribution is

$$P(L|I) \propto \prod_{t=1}^{T}\prod_{i=1}^{N_t} P(I_t|L_t^i) \prod_{\{(t',i'),(t,i)\}\in Nbr_{ST}} P\left(L_{t'}^{i'}, L_t^i\right). \quad (14)$$

The Nbr_{ST} in (14) denotes the neighborhood relationship according to the spatial-temporal connectivity in the meshes. The likelihood $P(I_t|L_t^i)$ is measured based on the extent how the meshes are closed to the guessed object's regions. The guessed object's regions come from the initial results of the video segmentation method [27] and from the color segmentation based on logistic regression. The logistic regression is carried out on the features of color intensity, and with positive samples provided by the initial results of video segmentation. The continuity term $P(L_{t'}^{i'}, L_t^i)$ is simply defined as Potts model.

Loopy belief propagation is applied to solve the inference problem. Once the label is solved, the object's boundary is simply extracted by fitting a curve that encloses the mesh points of the object in each frame.

Some examples of the refinement are shown in Fig. 9, and the average number of error pixels is also compared against state-of-art methods on the SegTrack dataset [26] (see Table 1). Remarkable improvements w.r.t. [27] are achieved. If compared with the state-of-art, our simple refinement achieves the best result for one sequence, the second best on two sequences, and the second best on average. Such results prove the power of the middle-level

Method	[31]	[28]	[30]	**[27]**	[32]	[29]	**Ours**
birdfall	**186**	278	209	**155**	189	288	**177**
cheetah	**535**	824	796	**633**	806	905	**466**
girl	**761**	**1029**	**1040**	1488	1698	1785	1360
monkeydog	**358**	**192**	562	365	472	521	**294**
parchute	249	251	**207**	**220**	221	**201**	224
Avg.	**372**	**397**	427	452	542	592	**394**

Table 1: The pixel error rate of different segmentation methods.

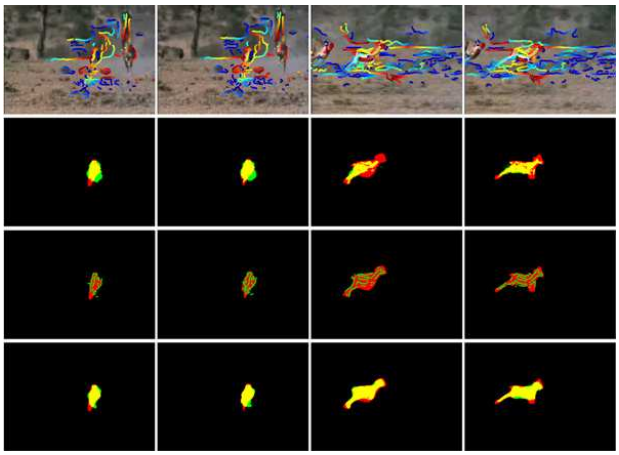


Figure 9: Visual examples of refinement. From top to bottom, in each frame: 1) Visualization of the mesh points, 2) The results of [27] (green channel) superimposed with ground truth (red channel, similar way of superposition hereinafter), 3) Labeling results of the mesh points, 4) Our refinement results.

motion representation based on contour flow.

6. Conclusion

This paper proposes a novel contour flow algorithm which is capable of establishing global consistent point correspondence among the inconsistent input contours between two video frames. We propose to use motion segmentation for local correspondence measurement, and experiments show the significant improved performance, if comparing with the ones purely based on static image information. To solve local ambiguities, a novel two-staged strategy is introduced to perform global reasoning, which can balance properly the accuracy and the consistency. Finally, a meaningful and stable mesh-based middle-level motion representation can be constructed by just concatenating frame-by-frame contour flow.

Several topics can be considered in the future. For instance, human pose estimation based on the proposals suggested by contour flow, and action recognition based on the motion descriptors extracted from the mesh-based motion representation. In addition to the simple example of refinement, more sophisticated video segmentation can also be investigated in further, by fully exploiting the S-T connectivity and boundary constraint from contour flow. We are also extending the ContourMotion dataset, and intend to release it to promote related researches.

References

[1] T. Brox and J. Malik. Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation. TPAMI, 33(3):500-513, 2011.

[2] D. Sun, S. Roth, and M. Black. A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles behind Them. IJCV, 106(2):115-137, 2014.

[3] L. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu. Large Displacement Optical Flow from Nearest Neighbor Fields. CVPR, 2013.

[4] J. Shi and C. Tomasi. Good Features to Track. CVPR, 1994.

[5] P. Sand and S.Teller. Particle Video: Long-Range Motion Estimation Using Point Trajectories. CVPR, 2006.

[6] N. Sundaram, T.Brox, and K. Keutzer. Dense Point Trajectories by GPU-accelerated Large Displacement Optical Flow. ECCV, 2010.

[7] H. Wang, A. Kläser, C. Schmid, and C. Liu. Dense Trajectories and Motion Boundary Descriptors for Action Recognition. IJCV, 103(1):60–79, 2013.

[8] T. Brox and J. Malik. Object Segmentation by Long Term Analysis of Point Trajectories. ECCV, 2010.

[9] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the Future: Spatio-temporal Video Segmentation with Long-range Motion Cues. CVPR, 2011.

[10] K. Fragkiadaki and J. Shi. Detection Free Tracking: Exploiting Motion and Topology for Segmenting and Tracking under Entanglement. CVPR, 2011.

[11] B. Sapp, D. Weiss and B. Taskar. Parsing Human Motion with Stretchable Models. CVPR, 2011.

[12] S. Zuffi, J. Romero, C. Schmid, and M. Black. Estimating Human Pose with Flowing Puppets. ICCV, 2013.

[13] K. Fragkiadaki, H. Hu, and J. Shi. Pose from Flow and Flow from Pose. CVPR, 2013.

[14] E. Hildreth. Computations Underlying the Measurement of Visual Motion. Artificial Intelligence, 23(3):309-354, 1984.

[15] C. Liu, W. Freeman, and E. Adelson. Analysis of Contour Motions. NIPS, 2006.

[16] P. Smith, T. Drummond, and R. Cipolla. Layered Motion Segmentation and Depth Ordering by Tracking Edges. TPAMI, 26(4):479-494, 2004.

[17] Y. Tsin, Y. Genc, Y. Zhu, and V. Ramesh. Learn to Track Edges. ICCV, 2007.

[18] P. Srinivasan, L. Wang, and J. Shi. Grouping Contours via a Related Image. NIPS, 2008.

[19] V. Jain, B. Kimia, and J. Mundy. Segregation of Moving Objects Using Elastic Matching. CVIU, 108:230-242, 2007.

[20] J. Meltzer and S. Soatto. Edge Descriptors for Robust Wide-Baseline Correspondence. CVPR, 2008.

[21] H. Tagare. Shape-Based Nonrigid Correspondence with Application to Heart Motion Analysis. IEEE Trans. Medical Imaging, 18(7):570-579, 1999.

[22] S. Xiang, F. Nie, and C. Zhang. Contour Matching Based on Belief Propagation. ACCV, 2006.

[23] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. TPAMI, 2002.

[24] P. Sundberg, T. Brox, M. Maire, P. Arbeláez, and J. Malik. Occlusion Boundary Detection and Figure/Ground Assignment from Optical Flow. CVPR, 2011.

[25] M. Ryoo and J. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA), 2010.

[26] D. Tsai, M. Flagg, A. Nakazawa, and J. Rehg. Motion Coherent Tracking using Multi-label MRF Optimization. IJCV, 100(2):190-202, 2012.

[27] D. Zhang, O. Javed, and M. Shah. Video Object Segmentation Through Spatially Accurate and Temporally Dense Extraction of Primary Object Regions. CVPR, 2013.

[28] D. Giordano, F. Murabito, S. Palazzo, and C. Spampinato. Superpixel-based Video Object Segmentation using Perceptual Organization and Location Prior. CVPR, 2015.

[29] Y. Lee, J. Kim, and K. Grauman. Key-segments for Video Object Segmentation. ICCV, 2011.

[30] W. Wang, J. Shen, and F. Porikli. Saliency-Aware Geodesic Video Object Segmentation. CVPR, 2015.

[31] S. Ramakanth and R. Babu. SeamSeg: Video Object Segmentation using Patch Seams. CVPR, 2014.

[32] T. Ma and L. Latecki. Maximum Weight Cliques with Mutex Constraints for Video Object Segmentation. CVPR, 2012.