Globally Optimal Toon Tracking

Haichao Zhu[1,2] Xueting Liu[1,2] Tien-Tsin Wong[1,2] Pheng-Ann Heng[1]
[1]The Chinese University of Hong Kong*
[2]Shenzhen Research Institute, The Chinese University of Hong Kong

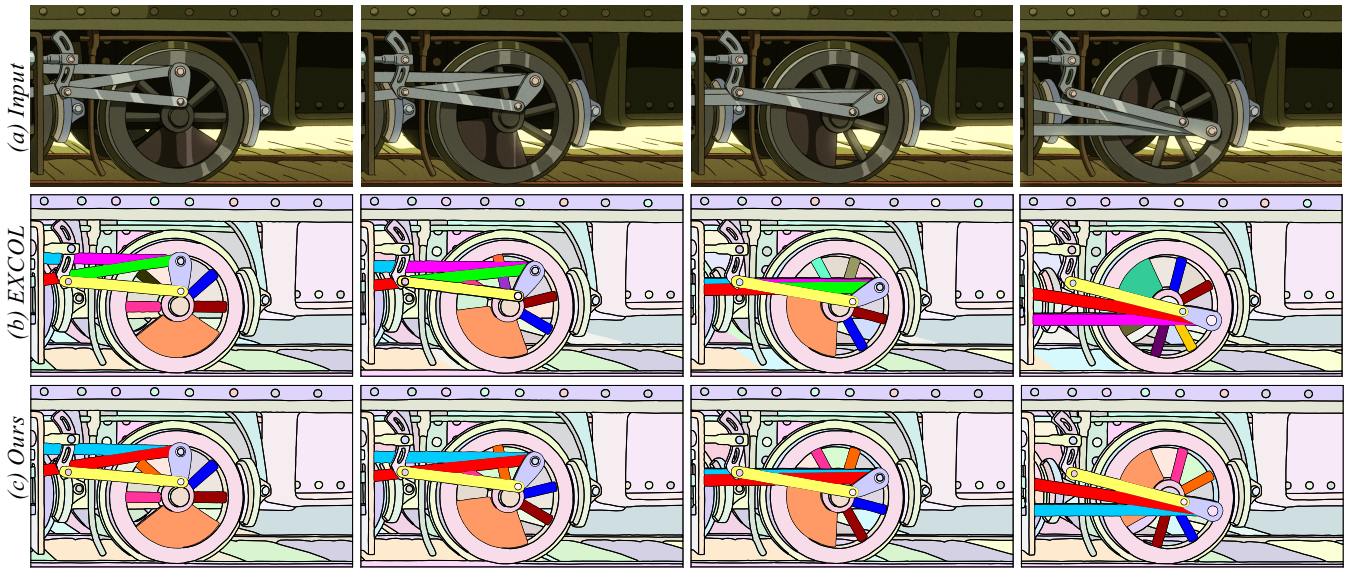

Figure 1: *"The train wheel."(a) Snapshots of an input sequence (10 frames, 1221×668) exhibiting severe partial occlusions (coupling rods) and similar objects (wheel spokes). (b) Region correspondences estimated by the state-of-the-art EXCOL. Note how coupling rods and wheel spokes are wrongly corresponded throughout the sequence. (c) Our result is highly accurate and consistent. For better comparison and visualization, we dress regions in (b) and (c) with the same colors as much as possible, and dim less-interesting regions.*

Abstract

The ability to identify objects or region correspondences between consecutive frames of a given hand-drawn animation sequence is an indispensable tool for automating animation modification tasks such as sequence-wide recoloring or shape-editing of a specific animated character. Existing correspondence identification methods heavily rely on appearance features, but these features alone are insufficient to reliably identify region correspondences when there exist occlusions or when two or more objects share similar appearances. To resolve the above problems, manual assistance is often required. In this paper, we propose a new correspondence identification method which considers both appearance features and motions of regions in a global manner. We formulate correspondence likelihoods between temporal region pairs as a network flow graph problem which can be solved by a well-established optimization algorithm. We have evaluated our method with various animation sequences and results show that our method consistently outperforms the state-of-the-art methods without any user guidance.

Keywords: toon tracking, region correspondence

*e-mail:{hczhu,xtliu,ttwong,pheng}@cse.cuhk.edu.hk

SIGGRAPH '16 Technical Paper, July 24-28, 2016, Anaheim, CA
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925872

Concepts: •**Applied computing** → **Arts and humanities;** *Fine arts;*

1 Introduction

Although 2D hand-drawn cel animations can now be produced cost-effectively on computer platforms, animators still have to manually draw each single frame via digital tablets (raster-based or vector-based) or physical papers followed by scanning (raster-based only). Once a hand-drawn animation sequence is completed, further modifications such as recoloring or shape-editing become tedious and troublesome. Animators need to manually modify the animation sequence in a frame by frame manner because correspondence information is often unavailable in a completed animation sequence. If we can automatically acquire spatiotemporal correspondence information in an animation sequence, we can further prorogate the above modifications automatically between frames. Besides, in the production of new animations, by knowing correspondence information, new inbetweening frames can be created by morphing. This can produce high-quality smooth-motion animations without increasing labor cost.

Note that it is impractical to request animators to provide correspondence information during initial drawing since animators are accustomed to drawing sketches freely without specifying correspondences. Identifying correspondences between objects in live-action videos is a well-studied area in computer vision, but state-of-the-art computer vision techniques such as optical flow estimation [Horn and Schunck 1981; Lucas and Kanade 1981] and region tracking [Shitrit et al. 2014; Park et al. 2015] still cannot be directly applied to track regions in cel animations

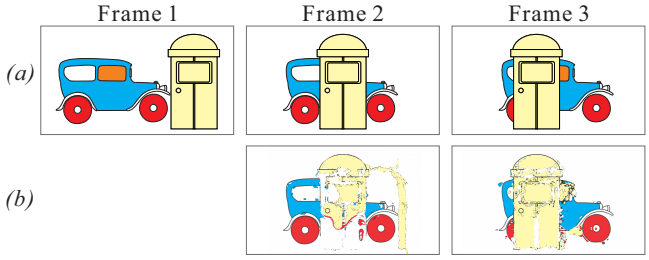

Figure 2: *(a) Input sequence. (b) Warped images from previous frames via optical flow estimation.*

because of three fundamental differences between cel animations and live-action videos. Firstly, there is no guarantee that animation contents are physically correct. Secondly, object motions in cel animations are more choppy and vigorous than those in live-action videos [Bregler et al. 2002]. Thirdly, animation objects are usually texture-less. All these violate the basic assumptions that modern optical flow estimation and region tracking algorithms rely on. Figure 2(b) visualizes a poor correspondence result generated by a state-of-the-art optical flow estimation method [Xu et al. 2012].

Research attempts have also been made in estimating region correspondences between two consecutive frames in animations or cartoons based on appearance features, such as color, shape, and distance [Sỳkora et al. 2005; Zhang et al. 2009; Zhang et al. 2012; Liu et al. 2013]. However, appearance features alone are insufficient to reliably identify region correspondences, when there exist complete or partial occlusions (e.g. the orange car window and the blue car body in Figure 2), or when two or more objects share same or similar appearances (e.g. the two red wheels in Figure 2). Correspondence estimation is further complicated when regions are split or merged during time (e.g. the blue car body in frame 1 is split into two parts in frame 2 in Figure 2).

According to the psychological studies [Adelson and Bergen 1985; Werner 1965], humans tend to group objects together based on not only their appearances but also their motions. Even if objects are texture-less, occluded and visually similar, humans can still correctly identify corresponded regions over a whole sequence based on their motions. By coupling motions and appearance features together, we are able to develop an improved correspondence identification model. In this paper, we propose a method to estimate spatiotemporal region correspondences over a cel animation sequence based on both the appearances and the motions of regions. It overcomes cases of complete or partial occlusions and is able to distinguish objects with similar appearances but different semantics. Our method can survive even in cases of region splitting or merging. The key is to *simultaneously* analyze correspondences of *all involved regions* and their *complete trajectories* over the whole animation sequence instead of tracking each individual region between two local consecutive frames. We formulate this spatiotemporal region correspondence estimation problem as a network flow problem and solve for the global optimum using the k-shortest path algorithm [Yen 1971]. Each flow in the network corresponds to a potential correspondence trajectory of a region. Since the optimal flows can branch and merge, we can natively support the complex region splitting and merging scenarios. Through the global optimization and a motion model, global region correspondences can be inferred even when there exist complete/partial occlusions and similar/identical objects. Figure 1(c) shows how our method effectively handles occlusions and distinguishes similar regions, respectively.

To validate the effectiveness of our method, we apply our method to various real-world challenging animation sequences with complete/partial occlusions, multiple identical/similar objects, or region splitting and merging. Convincing results are obtained in all tested cases. Our main contributions are summarized as follows:

- We propose to simultaneously estimate correspondences of all regions over the whole sequence based on both their appearances and motions.
- We formulate the problem as a network flow problem and solve for the global optimum using the k-shortest path algorithm.

2 Related Work

Existing works can be roughly classified into two categories: techniques tailored for live-action videos, and techniques tailored for cartoons and animations.

Techniques for Live-Action Videos For live-action videos, in order to estimate pixel correspondences across frames, state-of-the-art methods are commonly based on optical flow estimation [Horn and Schunck 1981; Lucas and Kanade 1981] which models motion of a point as spatial displacement. A comprehensive survey can be found in [Baker et al. 2011]. However, optical flow techniques cannot be directly applied to cel animations. The vigorous object motions in animations may violate the motion smoothness assumption in optical flow techniques. The texture-less nature of cartoons may also make the texture-based feature matching unreliable.

Multi-object tracking techniques tailored for live-action videos are also related to our application. These methods first detect moving objects in each frame and then link them into trajectories. To find optimal trajectories, a constrained network flow problem is usually formulated and solved via various approaches, such as bipartite graph algorithm [Pellegrini et al. 2010], k-shortest path optimization [Berclaz et al. 2011; Shitrit et al. 2014], and minimum-cost network flow [Butt and Collins 2013; Park et al. 2015]. An indepth survey can be found in [Smeulders et al. 2014]. These existing multi-object tracking methods also fail to be applied to cel animations because of the vigorous motions and texture-less regions in cartoons. Note that our network flow model is different from these existing methods by containing an explicit region motion model. This explicit motion model allows us to reliably depict choppy region motions between frames, even with occlusions.

Techniques for Cartoons The existing correspondence analysis methods tailored for cartoons and animations can be roughly classified into three categories with respect to the differences in primitives: point-based, stroke-based, and region-based.

Point-based correspondence estimation methods estimate point correspondences between consecutive frames based on local image features of points. In particular, Sỳkora et al. [2009; 2011] and Noris et al. [2011] proposed to model the correspondence between two cartoon frames as a as-rigid-as-possible deformation. Color information is required in order to measure local similarity between two image points. More attempts have been made in registering black-and-white line drawings without relying on color information [Song et al. 2013; Xing et al. 2015]. However, they are only applicable to small object deformations, and fail when the shapes of objects change vigorously (large motion).

Stroke-based correspondence estimation methods take vector-based animation frames as input, and find correspondences of strokes between every two consecutive frames. By assuming the depth ordering of objects unchanged throughout the sequence, Kort [2002] proposed a rule-based method to infer stroke

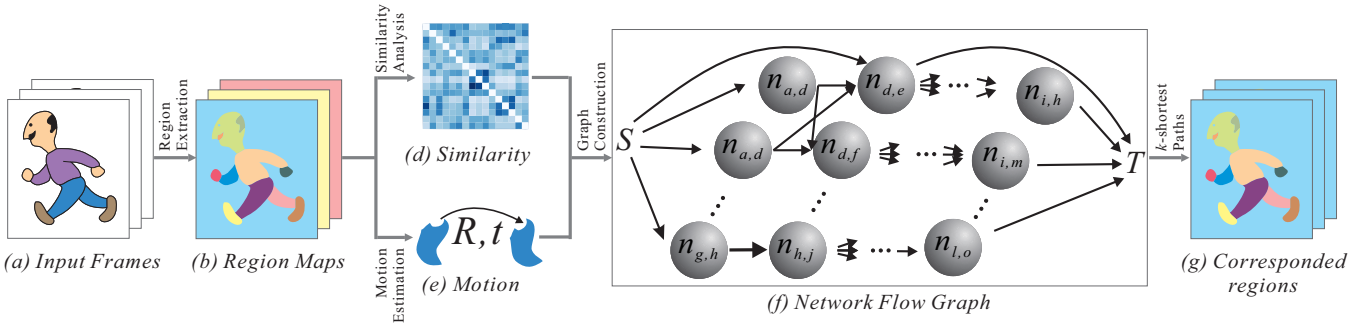

Figure 3: *System Overview. (b) Pixels color-coded with the same color represent a region. (d) Inter-frame appearance dissimilarity matrix. (e) Inter-region motion term. R is the rotation matrix. t is the translation vector. (f) A non-terminal node $n_{a,b}$ represents the correspondence relationship between two regions a and b. (g) Pixels color-coded with the same color in all frames represent a corresponded region.*

correspondences. Whited et al. [2010] proposed to find stroke correspondences by a set of user-guided semi-automatic techniques. Although these methods may handle occlusions in some specific scenarios, they generally fail to solve occlusions in real-world animations, which are more complicated and violate their assumptions.

Region-based correspondence estimation methods take regions as primitives, and find region correspondences between two consecutive frames based on appearance features, such as color, shape and distance [Zhang et al. 2009; Zhang et al. 2012; Liu et al. 2013]. They are quite vulnerable to partial occlusions where the shape of an occluded region may change abruptly. To tolerate slight appearance changes, topological relationships among regions within each frame are also considered [Madeira et al. 1996; Qiu et al. 2005; Sỳkora et al. 2005]. However, these methods still suffer from their local nature where region correspondences are only analyzed between two frames. Therefore, they fail to detect global region correspondences when regions are similar/identical, partial/complete occluded, or split/merged over time. In this paper, our method also takes regions as primitives, but we explicitly model region motions and globally optimize correspondences of all involved regions simultaneously over the whole sequence. This allows us to infer region correspondences in various complex scenarios.

3 Overview

Given an animation sequence (Figure 3(a)), we first extract and label regions for each input frame using a cartoon-tailored region extraction method [Liu et al. 2013] (Figure 3(b)). Our goal is to identify corresponded regions over the whole sequence in a semantically sensible fashion. For example, in the last row of Figure 1, corresponded regions are color-coded with the same color throughout the sequence.

With appearance and motion terms (Figure 3(d)&(e)) defined in the following section, we form a network flow graph (Figure 3(f)). There are a source and a sink in this graph. Each flow corresponds to a trajectory of a region, and must start from the source and end at the sink. Unlike traditional network flow formulations where each node represents a region, we propose to formulate the correspondence relationship between every pair of regions (from any two frames) as a node. The costs associated with nodes and edges encode appearance similarities and motion smoothness, respectively. By allowing region correspondences between any two frames (consecutive or non-consecutive), we are able to handle complete occlusions. By globally optimizing motion trajectories, we are able to distinguish regions with similar appearances but different semantics. Furthermore, the formulated network flow problem can be solved via the k-shortest path algorithm [Yen 1971] which already supports optimal flows to split and merge. Therefore, we can naturally handle region splitting and merging. The result of this network flow optimization is a set of optimal flows, corresponding to the optimal motion trajectories of all regions throughout the whole sequence. Our global optimization of region correspondences is detailed in Section 5.

To validate the effectiveness of our method in estimating region correspondences, we present several challenging real-world animation sequences and our results in Section 6.

4 Appearance and Motion Terms

Before describing our graph formulation, we first introduce the appearance features and the motion model utilized. Our correspondence estimation relies on both appearances and motions.

Color and Shape Similarities To measure the appearance similarity of two regions, we utilize both color and shape features. The color dissimilarity of two regions, a and b, is defined as the difference between their color histograms:

$$\mathcal{C}(a,b) = \|o(a) - o(b)\|_2 \tag{1}$$

Here, $o(a)$ and $o(b)$ are the color histograms (24 bins in our case) of regions a and b respectively. $\|\cdot\|_2$ is the L_2 norm operator.

To measure the shape similarity between two regions, we adopt the inner-distance shape context (IDSC) descriptor [Ling and Jacobs 2007] because it is more effective in capturing partial structures (higher tolerance to partial occlusions) than the classical shape context descriptor. For completeness, we briefly present the IDSC descriptor here. Given two regions a and b, the contour points of the regions are first extracted and denoted as $\boldsymbol{p}_1^a, \boldsymbol{p}_2^a, \cdots$ and $\boldsymbol{p}_1^b, \boldsymbol{p}_2^b, \cdots$, respectively. Here, $\boldsymbol{p}_i^a$ is the i-th contour point of region a. Then we compute the shape context histogram (60 bins in our case) for each contour point for both regions. The local shape dissimilarity between two contour points $\boldsymbol{p}_i^a$ and $\boldsymbol{p}_j^b$ on regions a and b respectively is measured as

$$s(\boldsymbol{p}_i^a, \boldsymbol{p}_j^b) = \frac{1}{2}\sum_k \frac{\left(h_i^a(k) - h_j^b(k)\right)^2}{h_i^a(k) + h_j^b(k)} \tag{2}$$

where h_i^a and h_j^b are the shape context histograms of $\boldsymbol{p}_i^a$ and $\boldsymbol{p}_j^b$, respectively. $h_i^a(k)$ is the value of the k-th bin in h_i^a. If the local shapes of $\boldsymbol{p}_i^a$ and $\boldsymbol{p}_j^b$ are similar, their shape context histograms h_i^a and h_j^b should also be similar, and their local shape dissimilarity $s(\boldsymbol{p}_i^a, \boldsymbol{p}_j^b)$ should be small, and vice versa.

The global shape dissimilarity between two regions is computed by first finding an optimal alignment between the contour points

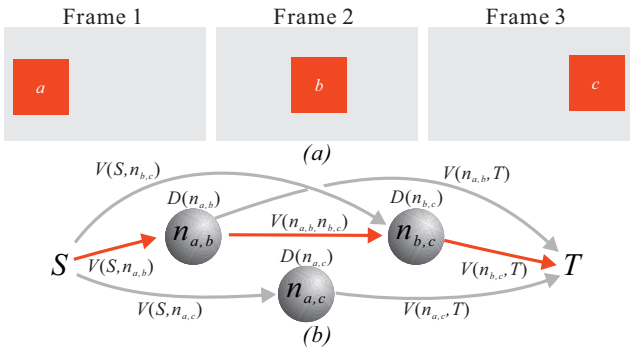

Figure 4: *Network flow formulation. (a) Input sequence. (b) Constructed complete network flow graph of (a). The background region is neglected. The red edges indicate the optimal flow.*

using dynamic programming, and then adding up all local shape dissimilarities between aligned contour points as

$$\mathcal{S}(a,b) = \sum_{\{\boldsymbol{p}_i^a, \boldsymbol{p}_j^b\} \in \Pi_{a,b}} s(\boldsymbol{p}_i^a, \boldsymbol{p}_j^b) \tag{3}$$

Here, $\Pi_{a,b}$ is the set of pairs of all aligned contour points.

Motion Term With IDSC, we then define the motion term between any two potentially corresponded regions, a and b, from any two frames. Note that the frames that contain a and b may be either consecutive or non-consecutive. Analyzing potential motions between regions from non-consecutive frames allows us to overcome complete occlusions (even occluded for over more than one frame). This motion term is defined as the optimal rigid transformation from a to b such that the transformed a is optimally matched with b, in terms of IDSC. A rigid transformation is represented as a rotation $\mathbf{R} \in \mathbb{R}^{2\times 2}$ and a translation $\mathbf{t} \in \mathbb{R}^2$. The optimal transformation, $\{\mathbf{R}_{a,b}, \mathbf{t}_{a,b}\}$, from a to b is then computed as

$$\{\mathbf{R}_{a,b}, \mathbf{t}_{a,b}\} = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{\{\boldsymbol{p}_i^a, \boldsymbol{p}_j^b\} \in \Pi_{a,b}} \left\| \mathbf{R} \cdot \boldsymbol{p}_i^a + \mathbf{t} - \boldsymbol{p}_j^b \right\|_2 \tag{4}$$

where $\Pi_{a,b}$ is the same as in Eq. 3. Note that the estimated motion may be imprecise if the contour points are poorly aligned. To make motion estimation more stable, we adopt the random sample consensus (RANSAC) method to remove unreliably aligned contour point pairs, and only leave reliable pairs in $\Pi_{a,b}$.

5 Spatiotemporal Region Correspondence

5.1 Network Flow Formulation

With the defined appearance and motion terms, we can now formulate this spatiotemporal region correspondence problem as a network flow optimization, and solve for the global optimal correspondences of all regions simultaneously. A network flow optimization is to find an optimal set of flows from a directed acyclic graph (DAG) with respect to some required constraints. Firstly, we define two terminal nodes, source and sink (denoted as S and T), respectively. Each flow must start from the source S and end at the sink T. Then, for every pair of regions a and b from any two different frames u and v where $u < v$, we construct a non-terminal node $n_{a,b}$ in our graph to represent the *correspondence relationship* between a and b. If there exists an optimal flow passing through node $n_{a,b}$, it means that a and b are corresponded regions in the optimal solution. This is very different from traditional network flow formulations which formulate each region as a node. Note also that we construct nodes for all possible pairwise combinations of regions from any two consecutive or non-consecutive frames. This allows us to resolve complete occlusions. A region in frame u can be directly corresponded to a region in frame $u+2$ or even further apart in time.

Figure 4(a) shows an example of a three-frame animation, in which a square moves from left to right. The corresponding network flow graph is shown in Figure 4(b). There exist three regions altogether in the three frames, and three non-terminal nodes are formed. Note that, we neglect the background region for illustrative simplicity in Figures 4-7. But in practice, the background region(s) is(are) also accounted during actual computations. Next, for every pair of non-terminal nodes $n_{a,b}$ and $n_{b,c}$ where the second region of the first node and the first region of the second node are identical, we connect them with a directed edge pointing from $n_{a,b}$ to $n_{b,c}$. Furthermore, for each non-terminal node $n_{a,b}$, we create a directed edge from S to $n_{a,b}$ and another edge from $n_{a,b}$ to T.

A flow is a sequence of linked nodes starting from the source node S, passing through the non-terminal nodes, and ending at the sink node T, e.g. $S \rightarrow n_{a,b} \rightarrow n_{b,c} \rightarrow T$. Each flow in the final solution corresponds to a region trajectory over the whole sequence. If this flow goes from $n_{a,b}$ to $n_{b,c}$, it means that regions a, b and c are temporally corresponded. To determine the optimal flows, we then associate a cost to each node and each edge.

Node Cost We assign each non-terminal node $n_{a,b}$ with a node cost $\mathcal{D}(n_{a,b})$ which suggests how likely regions a and b from frames u and v respectively are corresponded. It accounts for both appearances and motions, and is defined as a function of color dissimilarity, shape dissimilarity, inter-frame motion, and frame distance as

$$\mathcal{D}(n_{a,b}) = (\mathcal{C}(a,b) + \lambda_1 \mathcal{S}(a,b) + \lambda_2 \mathcal{M}(a,b)) \cdot G(v-u) \tag{5}$$

where $\mathcal{C}(a,b)$ and $\mathcal{S}(a,b)$ are the color and shape dissimilarities defined in Eq. 1 and Eq. 3 respectively. If two regions are quite similar in terms of color and shape, the node cost should be small. λ_1 and λ_2 are weighting factors and empirically set to $0.01 \sim 0.1$ and $0.7 \sim 1.2$ respectively. Inter-frame motion

$$\mathcal{M}(a,b) = \|\mathbf{\Omega}_{a,b}\|_1 + \|\mathbf{t}_{a,b}\|_1 \tag{6}$$

measures the local transformation between a and b where $\mathbf{\Omega}_{a,b}$ is the rotation angle of the estimated rotation matrix $\mathbf{R}_{a,b}$ defined in Eq. 4, $\mathbf{t}_{a,b}$ is the estimated translation vector defined in Eq. 4, and $\|\cdot\|_1$ denotes the L_1 norm operator. Intuitively, if two regions remain unchanged temporally, it is very likely that they are corresponded. $G(x)$ is a frame distance function defined as

$$G(x) = \alpha^x \tag{7}$$

which penalizes the correspondence choice if a and b are coming from two further apart frames. With this formulation, we favor correspondences of regions from temporally closer frames and encourage the optimal trajectories to pass through as many frames as possible, while we are still able to handle possible complete occlusions. In fact, $G(x)$ can be any monotonic increasing function. We empirically select the exponential form and set α to 100 for all our experiments.

Edge Cost We associate each edge that links $n_{a,b}$ to $n_{b,c}$ with an edge cost $\mathcal{V}(n_{a,b}, n_{b,c})$ to ensure smoothness of object motions. Since objects in animations are very likely to move with constant velocities, we model the motion smoothness among a sequence of regions as the change in velocity (in some sense, acceleration):

$$\mathcal{V}(n_{a,b}, n_{b,c}) = \|\nabla \mathbf{\Omega}_{a,b,c}\|_1 + \|\nabla \mathbf{t}_{a,b,c}\|_1 \tag{8}$$

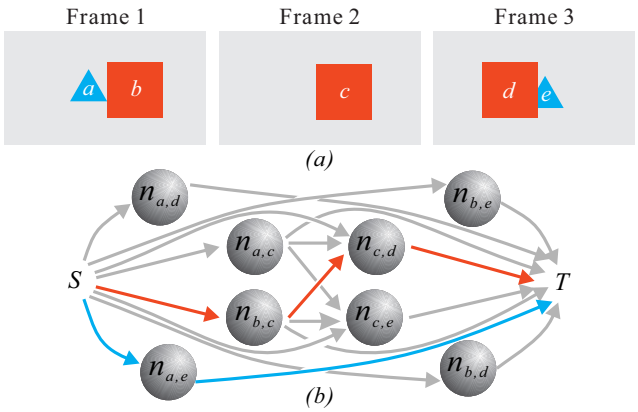

Figure 5: *Occlusion. (a) Input sequence. (b) Network flow graph of (a). The optimal flows are color-coded in the same colors with corresponded regions.*

Here,

$$\nabla\mathbf{\Omega}_{a,b,c} = \left\| \frac{\mathbf{\Omega}_{b,c}}{w-v} - \frac{\mathbf{\Omega}_{a,b}}{v-u} \right\|_1 \quad (9)$$

$$\nabla\mathbf{t}_{a,b,c} = \left\| \frac{\mathbf{t}_{b,c}}{w-v} - \frac{\mathbf{t}_{a,b}}{v-u} \right\|_1 \quad (10)$$

can be regarded as angular acceleration and acceleration normalized by frame distance (time), respectively. u, v, and w are the indices of frames containing regions a, b, and c, respectively. In other words, if the transformations (motions) remain unchanged (or the acceleration is zero) among a sequence of regions, there is a higher chance that these regions are corresponded. For each directed edge that links S and $n_{a,b}$ or $n_{a,b}$ and T, we associate it with an edge cost as:

$$\mathcal{V}(S, n_{a,b}) = G(u-1) \quad (11)$$

$$\mathcal{V}(n_{a,b}, T) = G(N-v) \quad (12)$$

Here, u, v are the indices of frames containing a and b respectively. N is the number of frames of the input sequence.

Optimization Based on the formed network flow graph, the overall energy function is defined as

$$\sum_J \left(\sum_{n_{a,b} \in J} \mathcal{D}(n_{a,b}) + \sum_{(n_{a,b}, n_{b,c}) \in J} \lambda_3 \mathcal{V}(n_{a,b}, n_{b,c}) \right) \quad (13)$$

where J indicates an optimal flow, and λ_3 is a weighting factor and set to $1.0 \sim 2.0$ empirically. This network flow problem can be solved via the k-shortest path algorithm [Yen 1971] by minimizing the overall energy. This energy minimization is solved via an iterative approach. One shortest path is found during each iteration. The optimization terminates until enough amount of shortest paths have been found so that each region appears in at least one path. The result of this global optimization is a set of optimal flows (paths) where each flow corresponds to a sequence of corresponded regions. Every region, that visually appears in the animation, should have been accounted for its correspondences.

5.2 Case Studies

We now explain how the network flow optimization acts when tackling three challenging scenarios, including occlusions, identical regions, and region splitting. In each case, we show a meaningful toy example and the computed optimal flow(s), to illustrate how our model tackles these challenges.

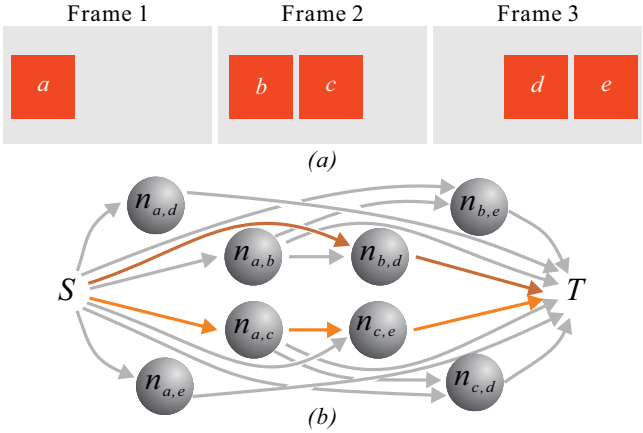

Figure 6: *Multiple identical regions. (a) Input sequence. (b) Network flow graph of (a). The orange and brown edges visualize two optimal flows.*

Occlusion Figure 5(a) shows an example where the moving blue triangle is completely occluded by the orange square and disappears in frame 2. Even if the triangle reappears in frame 3, it is still partially occluded. Since we construct nodes for all possible correspondences over the whole sequence, there exists the node $n_{a,e}$ in our graph that represents the potential correspondence of regions a and e. Even though the triangle is no longer in triangular shape in frame 3, the appearance dissimilarity between a and e (node cost of $n_{a,e}$) is still low since the IDSC descriptor can tolerate a certain level of partial occlusions. Figure 5(b) plots the corresponding graph with the optimal flows (region correspondences) color-coded in the same colors with the corresponding regions. While our model penalizes long durations of complete occlusions with the penalty function G, we can still effectively find the correspondence between a and e.

Identical Regions When an animation contains regions of identical appearances (same color and shape), appearance features are no longer effective in tracking regions. In this case, the motion term in our model plays a more important role in finding correct region correspondences. Consider the example shown in Figure 6(a) where two identical orange squares move together from left to right with the same speed. If only the appearance terms are considered, it is very likely that regions a and b are corresponded while regions c and d are corresponded, because of their spatiotemporal proximity. However, with the motion smoothness requirement defined in our model, the above solution will be rejected because of the sudden appearance of c in frame 2 and e in frame 3, as well as the sudden disappearance of b in frame 3. Figure 6(b) plots the complete network flow graph and the two optimal flows computed. We intentionally color-code the two optimal flows in different colors for illustrative purpose. The orange flow corresponds to $a \rightarrow c \rightarrow e$ and the brown flow corresponds to $b \rightarrow d$.

Region Splitting Because of occlusions, it is possible that a region is split into multiple sub-regions (e.g. Figure 7(a)). While a is split into sub-regions d and e, the IDSC descriptor still returns low node costs for both $n_{a,d}$ and $n_{a,e}$ since both a, d and a, e are partially matched. At the same time, b and c are identical in appearance. All these factors lead to the optimal flows presented in Figure 7(b).

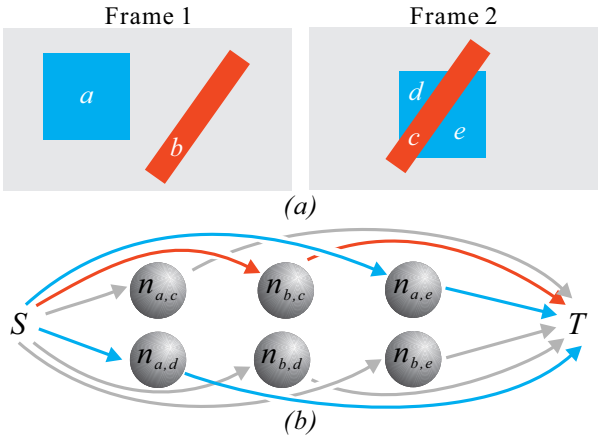

Figure 7: *Region splitting. (a) Input sequence. (b) Network flow graph of (a). The optimal flows are color-coded in the same colors with corresponded regions.*

6 Results and Discussions

To validate the effectiveness of our method, we apply it to various styles of challenging animation sequences. While the animations in Figures 1, 9, 11 and 12 are in oriental style, the animations in Figures 10, 13, and 14 are in western style. Besides, we also demonstrate the applicability of our method on gray-scale animations (Figure 10) and animated line drawings (Figure 14). We intentionally choose challenging shots which contain partial/complete occlusions, multiple similar objects, region splitting and merging, or change of region ordering.

Comparisons with Existing Methods In our experiments, we choose the state-of-the-art EXCOL [Zhang et al. 2012] for comparison. Most other existing methods perform local correspondence estimation similarly to that in EXCOL. While our method is fully automatic, EXCOL requires manual input when correspondences of some regions cannot be determined. To be fair, we minimize the user input to EXCOL while allowing it to process all frames. We choose the best results we can obtain from EXCOL to compare with ours. For better visualization, we also try to use the same colors for regions in EXCOL and our results whenever possible.

Figure 9(a) shows snapshots from an animation sequence where an old lady is walking behind a column. In this sequence, both partial and complete occlusions occur. Without the global analysis of region correspondences, EXCOL fails to find the corresponded regions before and after the occlusion (Figure 9(b)). In contrast, our global optimization method can correctly identify region correspondences even after the occlusion (Figure 9(c)). The gray-scale cartoon "Popeye" in Figure 10(a) shows a typical walking sequence. Identifying its region correspondences is well-known to be challenging because of the continuous partial or complete occlusions, as well as similar appearances of left and right legs. The left leg is even split into two regions in the last snapshot. Even though the input sequence lacks chromaticity and the movement of the character is vigorous, our method can still accurately estimate the region correspondences (Figure 10(b)).

Figure 11(a) shows an animation of two moving dishes holding multiple similar objects (beans and hams). EXCOL fails to correctly distinguish these similar-appearance regions (Figure 11(b)), while our method correctly estimates the correspondences (Figure 11(c)). A more complicated example is presented in Figure 1. Here, the coupling rods of the train wheels are occluded and split into sub-regions. The wheel spokes which are almost identical further pose difficulty in correspondence determination. Again, our competitor fails to recognize that the split sub-regions are actually the same coupling rods semantically, and gets confused by the similar spokes. In sharp contrast, our method can successfully identify the region correspondences regardless of rotation, similar-appearance regions, and complex occlusions. Two more complicated sequences are shown in Figure 12 and Figure 13 respectively and one colorless sequence is shown in Figure 14. Our method obtains satisfactory results in all tested cases, with a small amount of errors. In general, the failure of our competitor is due to their local correspondence determination. Our method, on the other hand, can resolve the above challenges via a global optimization and an explicit motion model.

In Figure 8(a), the ordering of objects changes over time. Note that how our competitor is confused by the two legs and two shoes in Figure 8(b). Our method is robust to such ordering changes, even though depth ordering is not modeled in our formulation (Figure 8(c)).

To compare quantitatively, we measure the correctness of correspondence for both EXCOL and our method by comparing the estimated region correspondences with groundtruth. The groundtruth of region correspondences is obtained by manual labeling. Please refer to the supplementary materials for groundtruth comparisons. To measure the correctness of correspondences, we first measure the pairwise correctness for every two frames u and v (not necessarily consecutive) by counting the number of pixels (i.e. regions weighted by their areas) in u that are correctly corresponded in v. This count is normalized, by the total number of pixels, to the range of [0,100%]. The overall correctness is simply the average of pairwise correctness values of all possible frame pairs. Intuitively, an overall correctness of 100% means all correspondences are correctly identified. On average, our method obtains an overall correctness of 95.82%, while EXCOL obtains 82.48%. However, because images are very likely to be dominated by background regions, a clearer picture of correctness can be obtained by ignoring background pixels during the measurement. In this case, we achieve 92.26% and EXCOL achieves 53.50%. Again, our method outperforms EXCOL in terms of correctness. We tabulate the correctness (with and without background pixels accounted) of each tested case in Table 1.

Implementation Details When the animation sequence is very long, the number of nodes, the number of edges and the processing time may grow significantly as we account for all possible pairs of regions in our network flow graph. For practical implementation, we propose two strategies to boost the efficiency of our system. The first strategy is "pruning," in which we remove nodes and edges from the constructed graph before optimization if the associated node costs or edge costs are extremely large. Formally speaking, for each node $n_{a,b}$, if its node cost $\mathcal{D}(n_{a,b})$ is larger than $G(v-u)\Theta_{\mathcal{D}}$ where $\Theta_{\mathcal{D}}$ is a user-defined threshold, it will be removed from the graph together with its connected edges. Besides, for each edge that links $n_{a,b}$ to $n_{b,c}$, if its edge cost $\mathcal{V}(n_{a,b}, n_{b,c})$ is larger than a user-defined threshold $\Theta_{\mathcal{V}}$, this edge will be removed from the graph. Empirically, we set $\Theta_{\mathcal{D}} \in [1.0, 2.0]$ and $\Theta_{\mathcal{V}} \in [0.5, 1.0]$. This can remove a large set of nodes and edges which are definitely not inside the optimal paths.

The second strategy is a hierarchical optimization which gradually solves a whole sequence from local to global. To do so, we first solve Eq. 13 for every three frames and remove nodes and edges that are not in the optimal paths. Intuitively, region trajectories that are locally not optimal cannot be globally optimal either, so they can be removed in advance. Then we continue to enlarge the local sequence from three frames to five frames and perform similarly.

	With background	Fig.1	Fig.8	Fig.9	Fig.10	Fig.11	Fig.12	Fig.13	Fig.14	Average
EXCOL	Yes	55.78%	97.77%	94.38%	85.22%	95.82%	88.07%	88.38%	54.38%	82.48%
	No	55.78%	63.81%	70.66%	45.22%	62.82%	39.49%	35.82%	54.38%	53.50%
Ours	Yes	93.62%	100.0%	98.67%	93.64%	99.13%	94.14%	90.14%	97.21%	95.82%
	No	93.62%	100.0%	98.13%	86.21%	98.59%	81.07%	83.26%	97.21%	92.26%

Table 1: *Correctness of correspondence.*

We repeat such process until the local sequence covers the whole animation sequence. We tabulate the number of nodes and edges at three stages (originally constructed graph, after naive pruning, and after hierarchical optimization) in Table 2.

Running Time All our experiments are conducted on a PC with Intel Core i7-4710 2.5GHz, 16GB RAM. The computational time for all examples in the paper are summarized in Table 2. The precomputation time includes the computation of appearance and motion terms between every two regions in two different frames (without acceleration). The optimization is accelerated by the proposed two strategies (pruning and hierarchical optimization). According to our experiments, pruning can reduce about half of the optimization time. A detailed comparison can be found in Table 2 (last two rows). We also found that the hierarchical optimization is extremely important in making the optimization tractable. Without the hierarchical solution, even the short sequence in Fig. 8 takes more than 4 hours to optimize (comparing to 31 seconds after the proposed acceleration). This is due to the large scale of the network flow graph where the number of nodes is exponentially related to the number of regions and the number of frames. Our current system is implemented using MATLAB without any GPU acceleration. We believe that the running time, especially the precomputation time, can be significantly reduced with the parallelization of GPU.

Limitations Since our method is a region-based correspondence estimation method, it fails if region segmentation is poor. For example, the contour lines of regions may be blurry because of some special effects (e.g. glare, fog, etc) or compression artifacts (e.g. JPEG blocking artifacts), so two semantically independent regions may be erroneously merged into one region during the segmentation phase. Currently, we can only ask users to fix the segmentation via an interactive tool. Besides, our current motion term is a rigid transformation. It may not work well when the deformation or the movement of regions is too vigorous so that the motion term cannot be well represented by a rotation and a translation. Our method may still be confused if the scene contains a vast number of moving, similar-appearance objects which are very close to each other. This confusion is caused by vigorous motions of cel animations as well as the imprecise motion model.

Moreover, while our system can handle partial or even complete occlusions, our system may still fail when the range of the occlusion is very long. In order to handle long-range occlusions, we may modify the frame distance function (Eq. 7) by setting α to a smaller value than that we used in experiments. But consequently, if α becomes small, the penalty of region correspondence from two further apart frames becomes small. This may lead to incorrect correspondence path where one or two frames are skipped. For example, we have three corresponded regions a, b, and c in three consecutive frames. With a small α, it may become that the path $S \rightarrow n_{a,c} \rightarrow T$ is more preferred than the optimal path $S \rightarrow n_{a,b} \rightarrow n_{b,c} \rightarrow T$. Further study is needed in designing a more sophisticated frame distance function to tolerate long-range occlusions.

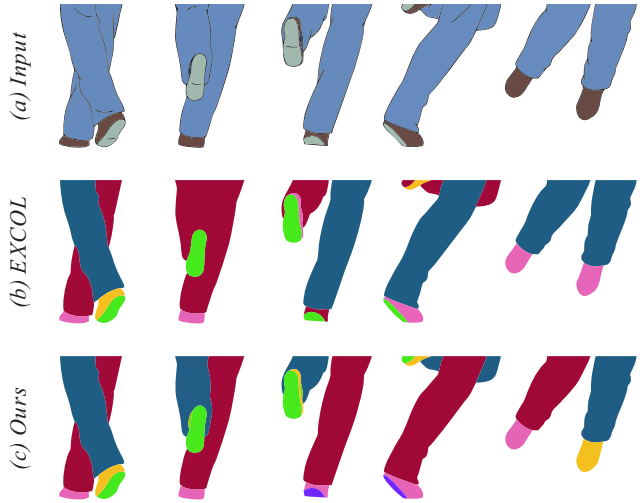

Figure 8: *"The legs." The input sequence contains 11 frames* (1280×720).

7 Conclusion

In this paper, we propose an optimization-based method to estimate spatiotemporal region correspondences over an animation sequence based on both region appearances and motions. We formulate the problem as a network flow problem and solve for the global optimum using the k-shortest path algorithm. The existing methods are easily confused by some challenging scenarios, such as partial/complete occlusions, multiple similar/identical objects, and region splitting/merging. In contrast, our method remains robust to all these tough scenarios and obtains convincing and consistent results, thanks to the explicit motion model and the global optimization to simultaneously infer all optimal region correspondences.

Our current formulation of motion term is rather crude (rigid transformation), and a more sophisticated motion model may further improve the correspondence estimation when the motion depicted in the animation is more complex. Currently, we do not know the hierarchical relationships among regions. We believe it is possible to utilize motion trajectories to deduce the hierarchy of regions. For instance, consider an example of a walking person, the trajectories of arm regions should be close to that of body regions, but superimposed with higher-frequency oscillations. With the obtained region correspondences, it should also be interesting to explore various applications.

Acknowledgments

This project is supported by NSFC (Project No. 61272293), and Research Grants Council of the Hong Kong Special Administrative Region, under RGC General Research Fund (Project No. CUHK 14200915).

References

ADELSON, E. H., AND BERGEN, J. R. 1985. Spatiotemporal energy models for the perception of motion. *Journal of the*

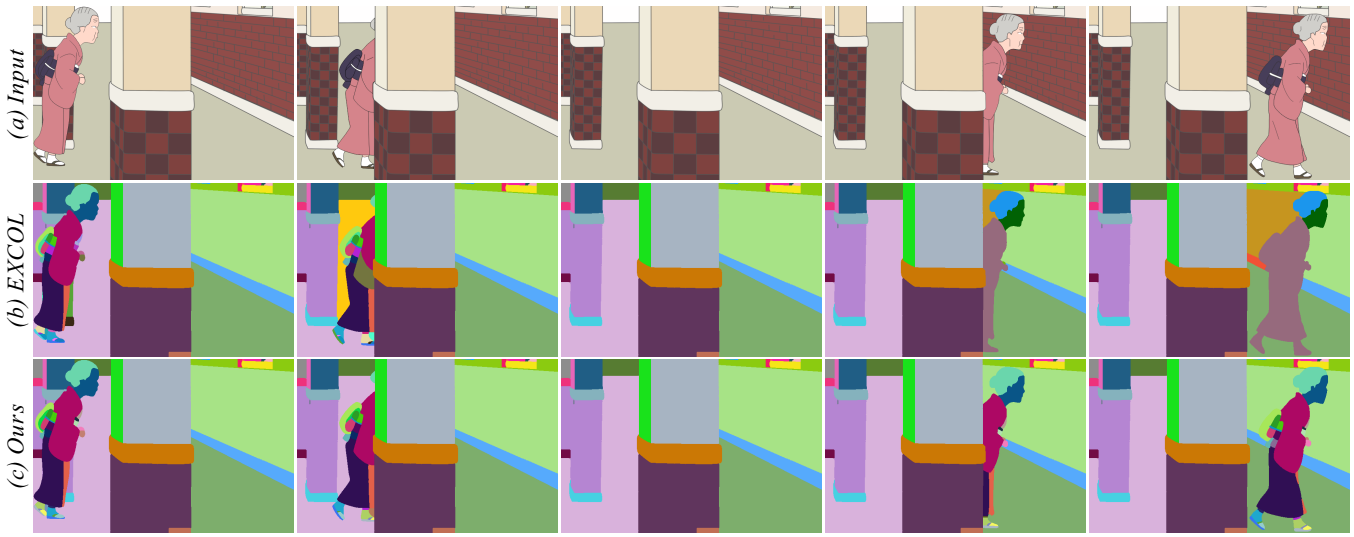

Figure 9: *"The old lady." The input sequence contains 23 frames (*760×846*).*

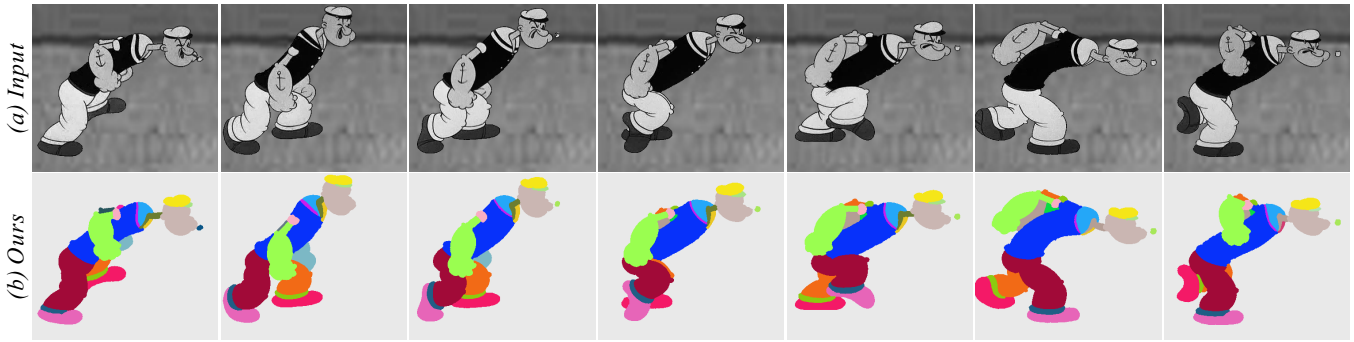

Figure 10: *"Popeye." The input sequence contains 20 frames (*720×480*).* ©*Fleischer Studios*

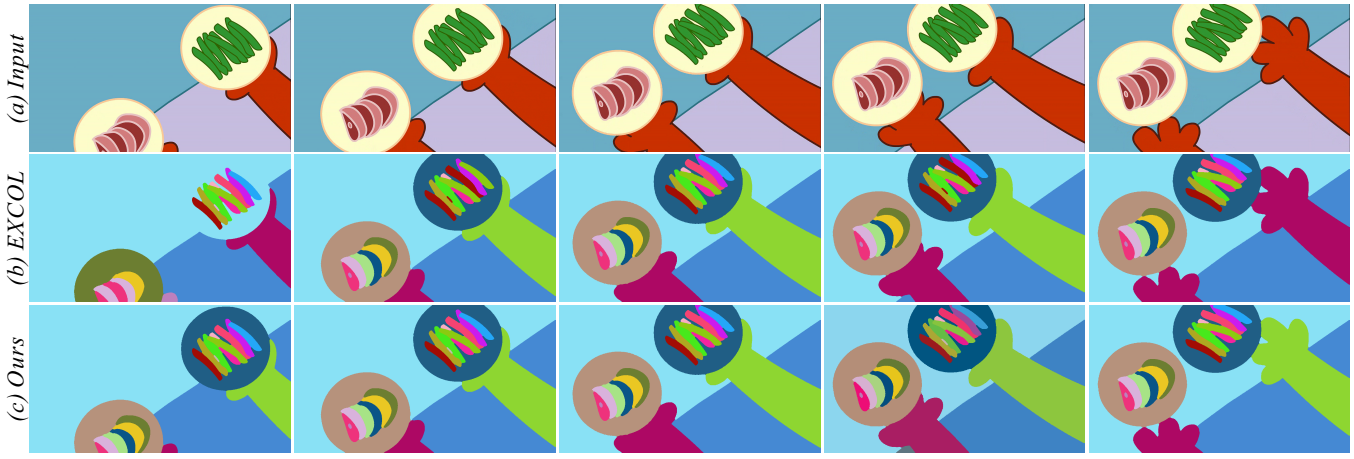

Figure 11: *"The dishes." The input sequence contains 20 frames (*1104×622*).* ©*Shanghai Quanguli E-Commerce Co., Ltd*

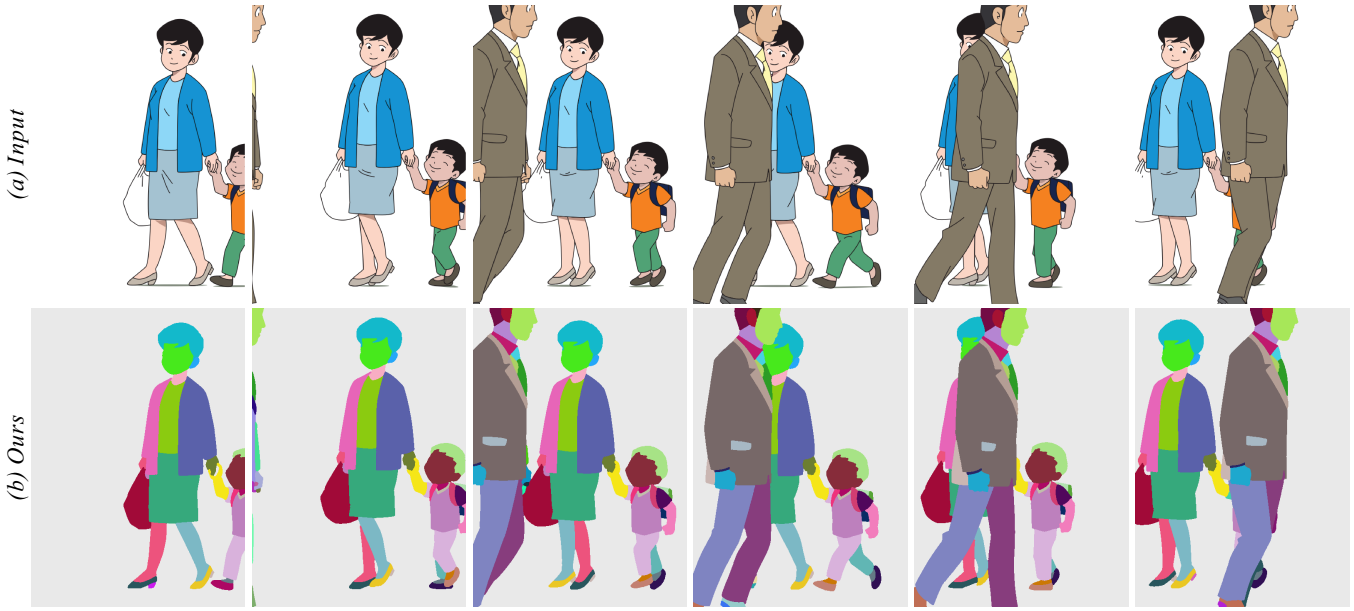

Figure 12: *"Walking." The input sequence contains 18 frames (*622×423*).*

Figure 13: *"Escaping." The input sequence contains 16 frames (*1920×1280*).*

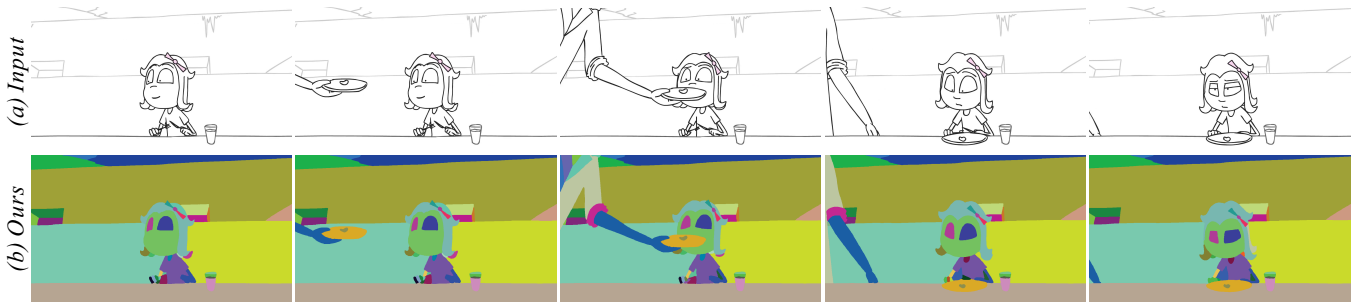

Figure 14: *"Raised by Zombies." The input sequence contains 20 frames (*1104×622*).* ©*Guy Collins*

	Stages	Fig.1	Fig.8	Fig.9	Fig.10	Fig.11	Fig.12	Fig.13	Fig.14
#nodes	original	1,306,125	2,093	637,755	359,099	179,190	174,447	246,914	891,172
	after pruning	80,370	337	47,094	18,262	32,632	8,997	3,923	14,192
	after hierach. opt.	1,453	61	1,097	542	351	525	586	1,299
#edges	original	593,052k	39k	123,662k	96,283k	46,502k	30,435k	50,723k	376,366k
	after pruning	3,113,119	1,166	1,617,535	475,610	49,920	107,938	17,739	13,430,968
	after hierach. opt.	1,243	54	1,046	520	327	461	479	1,246
time (sec)	precomputation	29,668	777	2,146	1,404	8,258	1,692	7,881	15,609
	optimization (pruning & hierach. opt.)	308	31	158	81	147	65	267	182
	optimization (hierach. opt.)	651	67	234	181	297	117	438	379

Table 2: *Data and timing statistics.*

Optical Society of America A 2, 2, 284–299.

BAKER, S., SCHARSTEIN, D., LEWIS, J. P., ROTH, S., BLACK, M. J., AND SZELISKI, R. 2011. A database and evaluation methodology for optical flow. *International Journal of Computer Vision 92*, 1, 1–31.

BERCLAZ, J., FLEURET, F., TÜRETKEN, E., AND FUA, P. 2011. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 9, 1806–1819.

BREGLER, C., LOEB, L., CHUANG, E., AND DESHPANDE, H. 2002. Turning to the masters: motion capturing cartoons. In *ACM Transactions on Graphics*, vol. 21, ACM, 399–407.

BUTT, A. A., AND COLLINS, R. T. 2013. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1846–1853.

HORN, B. K. P., AND SCHUNCK, B. G. 1981. Determining optical flow. In *Proceedings of International Society for Optics and Photonics Technical Symposium East*, 319–331.

KORT, A. 2002. Computer aided inbetweening. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*, 125–132.

LING, H., AND JACOBS, D. W. 2007. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 2, 286–299.

LIU, X., MAO, X., YANG, X., ZHANG, L., AND WONG, T.-T. 2013. Stereoscopizing cel animations. *ACM Transactions on Graphics 32*, 6, 223.

LUCAS, B. D., AND KANADE, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proceedings of International Joint Conference on Artificial Intelligence*, vol. 81, 674–679.

MADEIRA, J. S., STORK, A., AND GROSS, M. H. 1996. An approach to computer-supported cartooning. *The Visual Computer 12*, 1, 1–17.

NORIS, G., SÝKORA, D., COROS, S., WHITED, B., SIMMONS, M., HORNUNG, A., GROSS, M., AND SUMNER, R. 2011. Temporal noise control for sketchy animation. In *Proceedings of International Symposium on Non-photorealistic Animation and Rendering*, 93–98.

PARK, C., WOEHL, T. J., EVANS, J. E., AND BROWNING, N. D. 2015. Minimum cost multi-way data association for optimizing multitarget tracking of interacting objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*, 3, 611–624.

PELLEGRINI, S., ESS, A., AND GOOL, L. V. 2010. Improving data association by joint modeling of pedestrian trajectories and groupings. In *Proceedings of European Conference on Computer Vision*, Springer, 452–465.

QIU, J., SEAH, H. S., TIAN, F., WU, Z., AND CHEN, Q. 2005. Feature-and region-based auto painting for 2d animation. *The Visual Computer 21*, 11, 928–944.

SHITRIT, H. B., BERCLAZ, J., FLEURET, F., AND FUA, P. 2014. Multi-commodity network flow for tracking multiple people. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*, 8, 1614–1627.

SMEULDERS, A. W. M., CHU, D. M., CUCCHIARA, R., CALDERARA, S., DEHGHAN, A., AND SHAH, M. 2014. Visual tracking: an experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*, 7, 1442–1468.

SONG, Z., YU, J., ZHOU, C., AND WANG, M. 2013. Automatic cartoon matching in computer-assisted animation production. *Neurocomputing 120*, 397–403.

SỲKORA, D., BURIÁNEK, J., AND ŽÁRA, J. 2005. Colorization of black-and-white cartoons. *Image and Vision Computing 23*, 9, 767–782.

SỲKORA, D., DINGLIANA, J., AND COLLINS, S. 2009. As-rigid-as-possible image registration for hand-drawn cartoon animations. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*, 25–33.

SỲKORA, D., BEN-CHEN, M., ČADÍK, M., WHITED, B., AND SIMMONS, M. 2011. Textoons: practical texture mapping for hand-drawn cartoon animations. In Proceedings of International Symposium on Non-Photorealistic Animation and Rendering, 75–84.

WERNER, R. 1965. Autocorrelation: a principle for the evaluation of sensory information by the central nervous system. *Sensory Communication 303*, 318.

WHITED, B., NORIS, G., SIMMONS, M., SUMNER, R., GROSS, M., AND ROSSIGNAC, J. 2010. Betweenit: An interactive tool for tight inbetweening. *Computer Graphics Forum 29*, 2, 605–614.

XING, J., WEI, L.-Y., SHIRATORI, T., AND YATANI, K. 2015. Autocomplete hand-drawn animations. *ACM Transactions on Graphics 34*, 6, 169.

XU, L., JIA, J., AND MATSUSHITA, Y. 2012. Motion detail preserving optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 9, 1744–1757.

YEN, J. Y. 1971. Finding the k shortest loopless paths in a network. *Management Science 17*, 11, 712–716.

ZHANG, S.-H., CHEN, T., ZHANG, Y.-F., HU, S.-M., AND MARTIN, R. R. 2009. Vectorizing cartoon animations. *IEEE Transactions on Visualization and Computer Graphics 15*, 4, 618–629.

ZHANG, L., HUANG, H., AND FU, H. 2012. Excol: an extract-and-complete layering approach to cartoon animation reusing. *IEEE Transactions on Visualization and Computer Graphics 18*, 7, 1156–1169.