

# Coherent Parametric Contours for Interactive Video Object Segmentation

Yao Lu<sup>1</sup>, Xue Bai<sup>2</sup>, Linda Shapiro<sup>1</sup>, and Jue Wang<sup>2</sup>

<sup>1</sup>University of Washington , {luyao, shapiro}@cs.washington.edu

<sup>2</sup>Adobe , {xubai, juewang}@adobe.com

## Abstract

*Interactive video segmentation systems aim at producing sub-pixel-level object boundaries for visual effect applications. Recent approaches mainly focus on using sparse user input (i.e. scribbles) for efficient segmentation; however, the quality of the final object boundaries is not satisfactory for the following reasons: (1) the boundary on each frame is often not accurate; (2) boundaries across adjacent frames wiggle around inconsistently, causing temporal flickering; and (3) there is a lack of direct user control for fine tuning.*

*We propose Coherent Parametric Contours, a novel video segmentation propagation framework that addresses all the above issues. Our approach directly models the object boundary using a set of parametric curves, providing direct user controls for manual adjustment. A spatio-temporal optimization algorithm is employed to produce object boundaries that are spatially accurate and temporally stable. We show that existing evaluation datasets are limited and demonstrate a new set to cover the common cases in professional rotoscoping. A new metric for evaluating temporal consistency is proposed. Results show that our approach generates higher quality, more coherent segmentation results than previous methods.*

## 1. Introduction

Interactive, or supervised video object segmentation [17, 26, 4, 19] is an essential step in professional video production, enabling numerous post-processing possibilities such as background replacement. The standard industrial approach for this task is rotoscoping, where boundaries of the foreground objects are first annotated manually at sparse keyframes, using parametric and controllable shapes such as Bézier curves. These curves are then smoothly interpolated for the in-between frames. Given the high precision and controllability of parametric curves, rotoscoping can achieve highly accurate and temporally stable results by

artists; however it is an extremely labor-intensive process and requires professional expertise.

Recently, interactive video object segmentation based on sparse user input (i.e. foreground and background scribbles) has gained considerable attention given its ability to quickly generate reasonable segmentation results with a small amount of user input [22, 4, 7, 28]. While these scribble-based methods greatly improve the segmentation efficiency, they often suffer from inaccurate and/or inconsistent segmentation boundaries that prevent them from real production. Most of these approaches generate the segmentation results from pixels in single frames through a global optimization method such as graph cuts, which is easily affected by background clutter, image noise and edge pixelation. Furthermore, the factors affecting the global optimization often change across frames; thus even for a rigid object, there is no guarantee of the temporal shape consistency. Resulting boundaries often wiggle around across frames, causing *temporal boundary jitter* as shown in [Figure 1](#). Manipulating the pixel-wise boundaries frame-by-frame in such non-parametric systems is practically not possible.

In this project, we propose a new method called *Coherent Parametric Contours* (CPC), which explicitly models the object boundary as a set of evolving Bézier curves for interactive video object segmentation. These curves are initialized by the user on the first frame, and automatically propagated to the following frames through a spatio-temporal optimization algorithm that seeks both spatial accuracy and temporal shape consistency. Since object boundaries are represented as parametric curves, users have the full access to local boundary shapes; manipulating the curves is therefore straightforward.

Previously, the evaluation datasets proposed in [4, 19] do not provide ground-truth labeling with professional-level accuracy; they are not suitable for evaluating parametric algorithms as well, due to the ambiguity in parameterizing shapes with complex topology. Besides, there is also a lack of evaluation metric that focuses on temporal boundary consistency. We analyze the requirements for professional

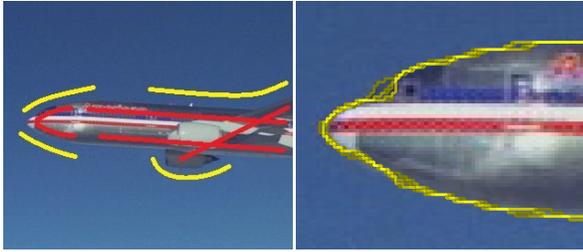


Figure 1. Temporal shape inconsistency is a common problem in scribble-based video object segmentation systems. **Left:** for a simple example with a rigid object, the user marks several scribbles on the first frame to indicate the foreground (red) and background (yellow). **Right:** overlaying boundaries (in yellow) of multiple frames reveals the temporal boundary inconsistency. The contours wiggle around despite the rigidity of the object. The video result for this example is shown at <http://yao.lu/CPC.html>

video object segmentation and construct a dataset containing various types of videos that commonly occur in real production. We provide ground-truth parametric boundaries carefully labeled by professionals using an industrial software package. A new metric is proposed to measure the temporal shape consistency for video object segmentation. Experimental results show that our approach outperforms state-of-the-art scribble-based segmentation methods, as well as rigid shape tracking algorithms.

## 2. Related Work

Multiple approaches exist for interactive video object segmentation. We discuss different types of systems below categorized by the input and the boundary propagation algorithm. (1) *Rotoscoping* tools such as Mocha [2] leverage simple shape interpolation methods to propagate parametric curves from keyframes to in-between frames, without looking at the underlying image content. Rotoscoping meets the quality requirement for production; consistent results are obtained for unlimited types of video objects (rigid, non-rigid, occluded, etc.). However rotoscoping is labor-intensive and requires much professional expertise for the users. To improve the efficiency, Agarwala et al. [3] encode image features through a non-linear optimization framework to interpolate for intermediate object shapes between user-annotated keyframes. (2) *Scribble-based* image and video segmentation systems take sparse user input and efficiently generate non-parametric segmentation results. Grabcut [20] segments the foreground object within input bounding boxes. SnapCut [4] improves foreground models using local classifiers. Most scribble-based frameworks utilize graph cuts to determine the boundary, hence potential solutions to enhance the fine boundary include using a prior to guide the graph cuts [25], and adjusting the affinity between image regions [13]. However, due to the nature of non-parametric curves, it is still tedious to perform pixel- or subpixel-wise annotation for fine bound-

ary manipulation. Besides, such manipulation is needed for every single frame, since results in these systems usually contain temporal jitter. (3) *Keypoint-based* contour tracking approaches extract local image descriptors and compute the homography between neighboring frames; the object boundary is then propagated to later frames using the homography. These systems provide efficient propagation of parametric contours. However they are usually limited to one planar surface; artifacts exist for objects with multiple parts or non-rigid motions. The Rigid Mask Tracker in Adobe After Effects [1] is a state-of-the-art implementation of this approach.

To benefit from existing approaches as well as to avoid their limitations, the following designs are made in our proposed system. (1) We mimic the rotoscoping artists to produce high quality boundaries for unlimited types of objects. However unlike rotoscoping, the users are only required to annotate the first frame. Our system automatically propagates the shapes to subsequent frames, which greatly reduces the workload. (2) We leverage parametric curves as input to our system, since non-parametric object boundaries produced by scribble-based methods are difficult for fine adjustment. (3) Locally instead of globally rigid motion is assumed to overcome the difficulty for handling non-rigid objects in keypoint-based systems.

Our proposed parametric boundary propagation framework is built upon active contours, a classic model for non-parametric image segmentation. Several variations exist such as snakes [11], intelligent scissors [16], and level sets [6]. Active contours has also been applied in several video object segmentation and tracking frameworks [10, 18, 21]. Unlike these approaches, our method models the input in a Bézier curve representation, and we perform spatio-temporal optimization upon these parametric boundaries to obtain accurate and smooth results.

Several datasets [4, 19, 28] exist to evaluate the effectiveness of interactive video object segmentation. Zhong et al. [28] provide a video set with binary masks for the foreground objects. However the groundtruth boundaries are not manually annotated and contain noticeable visual artifacts. These non-parametric datasets are not suitable for evaluating parametric algorithms due to the ambiguity in representing complex topology. Further, no cross-frame boundary correspondence is provided; evaluating the temporal consistency is indirect in these datasets.

In this paper, we demonstrate a parametric video dataset annotated by professional rotoscoping artists, with the goal of evaluating both spatial accuracy and temporal consistency for interactive video object segmentation. Automatic key frame selection methods are proposed in [27, 24]; however our annotation is on equally sampled frames so as to balance the difficulty between videos. For evaluation metrics, [12] to our best knowledge is the only work to eval-

uate the temporal consistency for soft object masks produced by video matting algorithms. Their metric is feature-dependent; label difference versus feature difference is calculated as the consistency scores. Since our dataset provides groundtruth annotated by professional artists, there is no need to further involve image features in the evaluation. Hence, we propose a novel temporal consistency metric for parametric curves in this paper. Our metric is defined purely on boundary shapes, allowing efficient and precise measurement without looking into image regions that sometimes become unreliable.

### 3. Coherent Parametric Contours (CPC)

#### 3.1. CPC in Single Images

Let  $I : [0, a] \times [0, b] \rightarrow \mathbf{R}^+$  be an intensity image, and let  $C(q) : [0, 1] \rightarrow \mathbf{R}^2$  be a parametric curve. Recall that the energy for the active contour [11, 6] is defined as:

$$E(C) = \alpha \int_0^1 |C'(q)|^2 dq + \beta \int_0^1 |C''(q)|^2 dq - \lambda \int_0^1 |\nabla I(C(q))| dq. \quad (1)$$

The first two terms penalize the first and second order discontinuities to ensure a smooth and continuous output, and the last term fits the curve to image gradient  $\nabla I$ . CPC on single images has a similar energy formation. In particular, the curve  $C(q)$  is represented using a set of connected Bézier curves:

$$C(q) = \{B(p^i)\}, i = 1, \dots, m, \quad (2)$$

where  $B(p^i)$  depicts the  $i$ -th cubic Bézier curve with parameter  $p^i = \{p_0^i, p_1^i, p_2^i, p_3^i\}$ , and  $m$  is the number of segments. For simplicity we denote  $B(p^i) = b_i(s) : s \in [0, 1] \rightarrow [p_0^i, p_3^i]$ . Here  $b_i(0) = p_0^i$  and  $b_i(1) = p_3^i$  are the two *terminal control points* of Bézier curve  $B(p^i)$ , and  $p_1^i, p_2^i$  are the two *intermediate control points*. Note that the Bézier curves are connected, hence  $b_i(1) = b_{i+1}(0)$ . We rewrite the fitness term in Equation 1 as:

$$\int_0^1 |\nabla I(C(q))| dq = \sum_{i=1}^m \int_0^1 |\nabla I(b_i(s))| ds. \quad (3)$$

Bézier curves are always continuous and smooth inside. To produce a reasonable CPC, the only requirement is the smoothness near the joints where two adjacent Bézier curves meet. Then the energy of CPC on a single frame can be written as:

$$E(B) = \sum_{i=1}^m |b_i(0)''|^2 - \lambda \sum_{i=1}^m \int_0^1 |\nabla I(b_i(s))| ds. \quad (4)$$

#### 3.2. CPC in Video

Given a video sequence  $V = (I_1, \dots, I_n)$ , to achieve spatio-temporal accuracy as well as consistency, we optimize the total energy of CPCs on the video sequence:

$$\min_{B_t \in \mathcal{B}_t, \forall t} \sum_{t=1}^n E(B_t) + \gamma \sum_{t=1}^{n-1} E(B_t, B_{t+1}), \quad (5)$$

where the first term ensures the quality of CPC on each individual frame.  $\mathcal{B}_t$  is the candidate boundary set for frame  $t$ , from which we select an optimal boundary to construct the global solution. We discuss constructing the candidate set in Section §3.2.1.

$E(B_t, B_{t+1})$  is the *temporal consistency cost* that measures the pairwise consistency between CPCs in neighboring frames. It is define as:

$$E(B_t, B_{t+1}) = \text{dist}(B_t \oplus \vec{f}, B_{t+1}), \quad (6)$$

where  $\vec{f}$  is the locally rigid motion vector, and  $\oplus$  is a boundary warping operation. We compare the warped boundary  $B_t \oplus \vec{f}$  in frame  $t$  with the candidate boundary  $B_{t+1}$  in frame  $t+1$ ; their pixel-wise distance is calculated and minimized, so that the resulting CPCs are consistent and deform progressively across frames according to the locally rigid motion. Note that  $\vec{f}$  is only locally rigid; it can be non-rigid globally. Details about estimating the locally rigid motion and warping the contours are provided in Section §3.2.2.

A well established criteria [14, 5] is applied to measure the distance between the boundary  $m$  and  $n$  in two frames. It can be formulated as the percentage of pixels on  $m$  that have correspondences to  $n$ :

$$\text{dist}(m, n) = \frac{1}{|m|} \sum 1_{p \in m, \exists q \in n, s.t. \|p-q\| \leq th} \quad (7)$$

where  $th$  is a tolerance threshold indicating the distance between corresponding boundary pixels.

To optimize Equation 5, we apply dynamic programming; the problem can then be solved within  $\mathcal{O}(mk^2)$  time, where  $k$  is the number of candidate boundaries for each frame. To speed up the computation, distances between candidate boundaries are calculated in parallel. Meanwhile, a standard multi-scale approach is applied to narrow the search space for  $k$ ; we generate CPCs on a rough scale and then refine them on a fine scale.

##### 3.2.1 Generating Boundary Hypotheses

Under the Bézier curve representation, generating a boundary hypothesis is equivalent to locating a set of control points. A two-stage approach is utilized to propose the candidate boundary set  $\mathcal{B}_t$  for frame  $t$ . We first generate the terminal points; then the intermediate control points can be determined by solving a least-squares fitting problem.

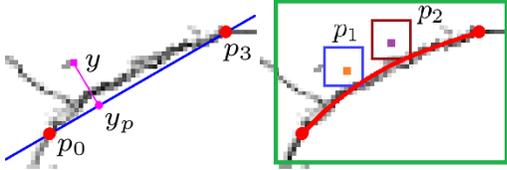


Figure 2. Illustration of our Bézier fitting scheme. **Left:** given  $p_0$  and  $p_3$  as the two terminal control points, project each nearby pixel (in the green box) to  $\overline{p_0p_3}$  for parameterization. **Right:** we use the image gradient values as weights and perform least-squares Bézier fitting to estimate optimal intermediate control points  $p_1$  and  $p_2$ . Two box constraints are used to stabilize the fitting.

**Proposing Candidate Terminal Points.** The criteria to generate candidate terminal points are two-fold. First, their displacement across frames should be consistent with the estimated local object motion. Second, they should snap to strong image edges. Hence, the candidates  $\{b_i^t(0)\}_{i=1}^m$  on frame  $t$  should minimize the following energy:

$$\min_{\{b_i^t(0)\}} \sum_{i=1}^m |b_i^t(0) - b_i^{t-1}(0) - \vec{f}_i^{t-1}| - \alpha |\nabla I_t(b_i^t(0))|, \quad (8)$$

where  $\vec{f}_i^{t-1}$  is the locally rigid motion of point  $i$ 's neighborhood within the object from frame  $t-1$  to  $t$ , which will be described in Section §3.2.2.

To solve Equation 8, computing and sorting the energies on a permutation set is a potential solution. However, moving candidate terminal points one or two pixels around yields a large permutation set with low variety; the computational complexity for Equation 5 is therefore high. We thus leverage a random sampling approach to obtain a small candidate set with large variety, similar in spirit to other sampling approaches applied in different applications such as matting [9, 23].

The possibility of a pixel  $x$  being a terminal point in frame  $t$  can be formulated as:

$$P_i^t(x) \propto \nabla I_t(x) \cdot N(b_i^{t-1}(0) + \vec{f}_i^{t-1}, \sigma) \quad (9)$$

where  $N(\cdot)$  is a 2D normal distribution centered at the terminal point projected using the locally rigid motion. Therefore, terminal points for different segments of the Bézier curves are generated independently and randomly according to  $P_i^t(x)$ , so that  $P(B^t) = \prod_{i=1}^m P_i^t(x)$ . We obtain  $k$  sets of terminal points for each frame. In our experiments  $k$  is typically chosen to be 300, and  $\sigma$  is set to be 5 pixels.

**Least-Squares Bézier Fitting.** Given the two terminal points  $p_0$  and  $p_3$  belonging to Bézier curve  $b$ , we perform constrained weighted least-squares fitting to infer the opti-



Figure 3. Estimating locally rigid motion for control points. **Left:** frame 1 with parametric annotation  $B$  in pink. Terminal points are shown in rectangles. **Right:** frame 2 with warped contour  $B \oplus \vec{f}$ . Small circles represent the keypoints, and short lines indicate their displacement from frame 1. For estimating the motion of the terminal point pointed to by the yellow arrow, we look into a local region within a radius  $r$ . Keypoints nearby are used to calculate the homography. Terminal points pointed to by red arrows do not have sufficient nearby keypoints, thus their motions are propagated from neighboring terminal points to keep shape rigidity.

mal locations for  $p_1$  and  $p_2$ :

$$\min_{p_1, p_2} \sum_{y \in \mathcal{C}} \nabla I(y) \cdot [y - b(l(y))]^2, \quad (10)$$

$$s.t. \quad 0 \leq l(y) \leq 1,$$

$$r_{lu}^1 \leq p_1 \leq r_{rb}^1,$$

$$r_{lu}^2 \leq p_2 \leq r_{rb}^2.$$

where  $b(s)$  is the cubic Bézier curve:  $b(s) = (1-s)^3 \cdot p_0 + 3(1-s)^2 s \cdot p_1 + 3(1-s)s^2 \cdot p_2 + s^3 \cdot p_3$ ,  $s \in [0, 1]$ . Figure 2 demonstrates an example of the Bézier fitting.

Let  $\mathcal{C}$  be the set of possible pixel locations for Bézier curve  $b$ . In practise, so as to reduce the search space,  $\mathcal{C}$  is set to be the pixels within the bounding box to contain  $p_0$  and  $p_3$  plus a constant margin (the green box in Figure 2).  $l(y) = \frac{\|y_p - p_0\|}{\|p_3 - p_0\|}$  parameterizes an arbitrary point  $y$ , and  $y_p$  is its projection on  $\overline{p_0p_3}$ . The image gradient  $\nabla I$  is used to weight the Bézier curves, so that they fit to strong edges. We apply additional constraints on the intermediate control points. Two box constraints  $r^1 = [r_{lu}^1, r_{rb}^1]$  and  $r^2 = [r_{lu}^2, r_{rb}^2]$  are used; their positions are bilinearly interpolated and projected from the previous frame using the locally rigid motion vector. We force the intermediate control points to locate within the box constraints so as to stabilize the fitting result. Equation 10 is solved using the Levenberg-Marquardt method [15] with RANSAC [8]; a subset of pixels in  $\mathcal{C}$  is selected in a sample-and-test manner to best fit the Bézier curve.

### 3.2.2 Locally Rigid Motion Estimation

Previous keypoint tracking methods treat the object as a single plane and calculate a global homography between frames. This is sub-optimal for segmenting non-rigid, or rigid but non-planar video objects. In this work, we instead emphasize the local rigidity of objects and leverage local affinities to estimate the motion for the terminal points. Figure 3 is an example of the estimation process.

The local homography  $H_x$  for point  $x$  on the object boundary is calculated using the keypoints within a radius  $r$  centered at  $x$ ; keypoints outside of the object boundary are not considered. RANSAC [8] is applied in calculating the homography to eliminate outliers. Hence the locally rigid motion vector is denoted as  $\vec{f} = H \cdot x - x$ .

For terminal points without enough neighboring keypoints, their motion vectors are propagated from nearby terminal points to keep shape rigidity, so that  $\vec{f} = (d^+ \cdot \vec{f}^+ + d^- \cdot \vec{f}^-)/(d^+ + d^-)$ , in which  $\vec{f}^+$  ( $\vec{f}^-$ ) is the motion of the next (previous) terminal point on the boundary (assuming the terminal points are annotated clockwise), and  $d^+$  ( $d^-$ ) is the distance on the object boundary to that point.

**Constructing warped contours.** Given the parametric Bézier curve  $B$  and the locally rigid motion vector  $\vec{f}$ , computing the warped contour  $B \oplus \vec{f}$  is therefore straightforward. For terminal control points,  $\vec{f}$  is applied directly; for intermediate control points, we apply motion vectors that are bilinear interpolated from the two neighboring terminal control points. Discrete pixels on the boundary are generated using these Bézier parameters.

## 4. Experiments

### 4.1. Dataset

**Motivation.** Currently, datasets used to evaluate scribble-based frameworks [28, 4, 19] emphasize the overall correctness of video segmentation, while the sub-pixel-level boundary quality is weighted less. Furthermore, the existing datasets are not designed for evaluating parametric methods due to the ambiguity in the parametric contour representation. Temporal consistency cannot be measured either, since no cross-frame correspondence is provided. In this paper, we propose a video set for evaluating parametric video object segmentation algorithms with emphasis on both spatial and temporal boundary qualities. We consider the following issues in the dataset construction.

1. *Complex topology.* Modeling objects with complex and changing topology using a single parametric curve is an ill-defined problem, not feasible even for professional rotoscoping artists. The standard solution in video production is to divide complex objects into overlapping parts with simple shapes, while each part can still be deformable, and rough boundaries are put between parts. Figure 4a shows an example of this process. To mostly follow the production practise, videos with partial boundary (PB) and occlusion (OC) should be included in our dataset (Figure 4b).

2. *Furry boundaries.* For objects with furry boundaries, rotoscoping artists will first generate consistent outlines for the whole object, and then apply soft matting locally to the furry part. For the binary segmentation step, we also categorize this case as partial boundary (PB).

3. *Occlusion.* Occluded objects pose extra challenges for

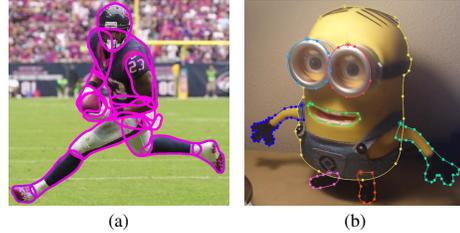


Figure 4. (a) Production practise for high quality interactive video object segmentation. Complex topology is decomposed into parts for segmentation. (b) A dataset is proposed to cover several common cases in production. In this case non-rigid motion (NR), partial boundary (PB) and motion blur (MB) exist.

Sequence	Init size	Len	Anno	Motion	MB	OC	PB
Boy	190x300	60	30	NR			✓
Drop	80x100	84	28	NR			
Minion	420x570	102	34	NR	✓		✓
Car	80x60	111	37	R	✓		
ToyMonkey	210x190	120	40	R		✓	
ToyHorse	320x290	120	40	R			✓
Plane	800x810	93	31	R			
Sunset	310x90	128	32	R		✓	
Tower	220x600	60	30	R			

Table 1. Overview of our proposed dataset. Len: original length of frames. Anno: number of annotated keyframes. Init size: rough object size (in pixel) on the first frame. R: rigid motion. NR: non-rigid motion. MB: motion blur. OC: occlusion. PB: partially annotated boundaries.

videos containing dynamic scenes. The segment boundaries should be consistent with the occlusion boundaries (OC).

4. *Motion blur* (MB) is a common artifact for videos containing intensive motions. It causes blurry object boundaries that are difficult for segmentation. Further, when there is severe motion blur, even humans cannot see the boundaries clearly. To handle such cases in production, a standard practice is to estimate temporally smooth boundaries and sacrifice spatial accuracy.

5. *Rigidity.* Rigid (R) and non-rigid (NR) objects are equally important for video segmentation, while previous systems overlook the boundary stability of rigid objects.

**Dataset construction.** We construct a dataset with 9 video sequences ranging from 60 to 128 frames in length (15-100fps), where each video contains a single shot for the rotoscoping process. We ask a professional rotoscoping artist to carefully label the boundary of a basic unit for each video, using the Bézier Pen tool in Mocha [2]. The annotation is later verified and refined by another rotoscoping artist. To reduce the cost for annotation and to balance the difficulties of different videos, we sample 28 to 40 frames from each video sequence with fixed intervals. Finally precise groundtruth in parametric curves is obtained. We demonstrate the dataset in Figure 4b and Figure 5, and Table 1 summarizes the videos properties.

### 4.2. Metrics

**Evaluating the spatial accuracy.** We follow the edge comparison strategy [14, 5] mentioned in Section §3.2 to

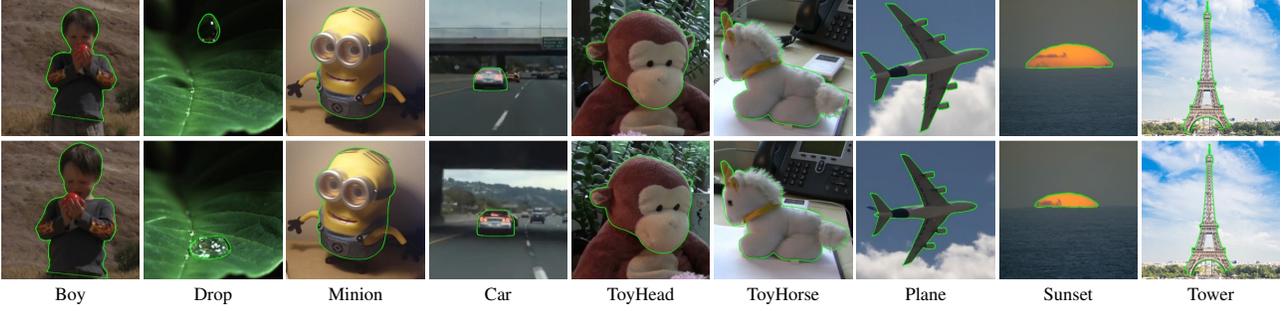


Figure 5. Different types of videos in our dataset. First row: frame 1. Second row: frame 25. The images are cropped for better visualization. The first three video sequences contain objects with non-rigid motion, and the rest contain objects with rigid motion. Groundtruth boundaries are shown in green.

evaluate the spatial accuracy for video object segmentation. Given the segmentation boundary  $sb$  and groundtruth boundary  $gt$  in discrete pixel format, the segmentation accuracy for a single frame is  $dist(gt, sb)$ .

**Evaluating the temporal consistency.** We propose a novel metric to measure the temporal consistency for parametric video object segmentation. Given the segmentation boundaries  $sb_1$  and  $sb_2$ , and groundtruth boundaries  $gt_1$  and  $gt_2$  for a pair of consecutive frames, let  $p$  ( $q$ ) be the closest pixels on  $sb_1$  ( $sb_2$ ) to the groundtruth  $gt_1$  ( $gt_2$ ), the consistency is therefore defined as:

$$consist(sb_1, sb_2) = \frac{1}{|gt|} \cdot \sum_i 1_{\| (gt_1^i - p^i) - (gt_2^i - q^i) \| \leq th} \quad (11)$$

The basic idea is to calculate percentage of pixels on  $gt_1$  and  $gt_2$  that have coherent correspondences to  $sb_1$  and  $sb_2$ . At this point, pixels on  $gt_1$  and  $gt_2$  should be registered so as to perform the per-pixel matching. Since our groundtruth boundaries are generated from Bézier parameters, we know the exact correspondence between  $gt_1$  and  $gt_2$ . Hence calculating Equation 11 is straightforward; we demonstrate the process in Figure 6.

### 4.3. Evaluations

We compare our proposed approach with several existing techniques as well as one variant of our framework. (1) Global plane tracking (GP) is a standard technique for video object segmentation. We compare with the Rigid Mask Tracker within Adobe After Effects [1], which is a state-of-the-art implementation of boundary tracking based on keypoints. We choose a perspective transform for the tracker. (2) We generate warped contours without the spatio-temporal optimization to see the effectiveness of the locally rigid motion (LR). (3) Finally, we compare with the state-of-the-art scribble-based (SB) approach SnapCut [4].

Since the videos are different in length, we divide each of them into multiple overlapping clips with a 5-frame offset between the clips. For each video clip, we evaluate the performances for  $len = 2, 6, 11$  and  $th = 1, 2, 4$  upon the

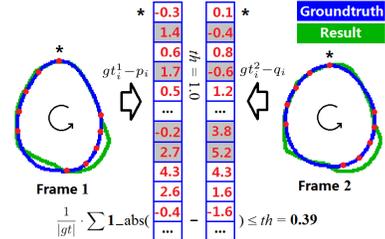


Figure 6. Demonstration of our temporal consistency metric. We calculate signed distances for each pair of corresponding pixels on the groundtruth boundaries (in blue) to the result boundaries (in green). The percentage of pixels in consensus between the two distances (in white cells) under a tolerance threshold  $th$  is calculated as the consistency score. Note that in this example we only demonstrate sparse correspondences.

annotated frames. Note that we find the results of previous tools often deteriorate quickly (e.g. across less than 10 frames); if the result is already not acceptable after propagating 11 frames, propagating more is not a meaningful comparison. The same first-frame annotations are fed to different methods as initialization; we then calculate accuracy per frame and consistency per consecutive two frames. Finally the average performances are reported on all the clips in each video.

We report the quantitative evaluations in Table 2 for the accuracy and consistency with  $th = 1$ . Under this tight setting the resulting boundaries should be within one pixel from the groundtruth. We have several observations from this comparison. (1) Global plane tracking (GP) works well on objects with rigid motion (R) in terms of both accuracy and consistency. It handles partially annotated boundaries (PB) but cannot cope with occlusion (OC). In contrast our approach achieves comparable performances on rigid tracking and finds the correct occlusion boundary in a better way. (2) As expected, global plane tracking does not work on objects with non-rigid (NR) motion. In contrast our framework handles non-rigid motion well, since we assume locally instead of globally rigid motion. (3) The scribble-based method (SB) is not comparable to our approach; the output boundaries are rough and temporally inconsistent. SB is good at detecting occlusion boundaries but does not

Setting		Boy	Drop	Minion	Car	Plane	Sunset	Tower	Monkey	Horse	Avg	R	NR	MB	OC	PB
SA & TC <i>len</i> = 2	SB	0.643	0.395	0.756	0.538	0.816	0.736	0.621	0.653	0.531	0.632	0.649	0.598	0.647	0.695	0.644
	GP	0.857	<b>0.514</b>	0.819	<b>0.940</b>	<b>0.984</b>	0.891	<b>1.000</b>	<b>0.997</b>	<b>0.999</b>	<b>0.889</b>	<b>0.969</b>	0.730	0.880	0.944	0.892
	LR	0.758	0.354	0.732	0.800	0.780	0.771	0.929	0.893	0.917	0.771	0.848	0.614	0.766	0.832	0.803
	Ours	<b>0.906</b>	0.460	<b>0.893</b>	0.933	0.910	<b>0.923</b>	0.990	0.990	0.996	<b>0.889</b>	0.957	<b>0.753</b>	<b>0.913</b>	<b>0.956</b>	<b>0.932</b>
SA <i>len</i> = 6	SB	0.614	0.274	0.719	0.444	0.784	0.721	0.535	0.640	0.519	0.583	0.607	0.536	0.582	0.680	0.618
	GP	0.602	<b>0.325</b>	0.613	<b>0.843</b>	<b>0.869</b>	0.548	<b>1.000</b>	<b>0.949</b>	0.965	0.746	0.862	0.513	0.880	0.748	0.727
	LR	0.587	0.233	0.645	0.666	0.628	0.653	0.925	0.793	0.831	0.662	0.749	0.488	0.766	0.723	0.688
	Ours	<b>0.741</b>	0.323	<b>0.791</b>	0.841	0.766	<b>0.855</b>	0.990	0.940	<b>0.970</b>	<b>0.802</b>	<b>0.894</b>	<b>0.618</b>	<b>0.913</b>	<b>0.897</b>	<b>0.834</b>
SA <i>len</i> = 11	SB	0.569	0.204	0.707	0.393	<b>0.733</b>	0.734	0.518	0.625	0.500	0.554	0.584	0.493	0.550	0.679	0.592
	GP	0.433	0.250	0.511	<b>0.766</b>	0.722	<b>0.999</b>	<b>0.864</b>	0.880	0.880	0.644	0.767	0.398	0.638	0.618	0.608
	LR	0.487	0.204	0.588	0.568	0.482	0.628	0.913	0.685	0.747	0.589	0.671	0.426	0.578	0.723	0.607
	Ours	<b>0.631</b>	<b>0.281</b>	<b>0.736</b>	0.721	0.607	<b>0.817</b>	0.990	0.853	<b>0.905</b>	<b>0.727</b>	<b>0.816</b>	<b>0.549</b>	<b>0.729</b>	<b>0.835</b>	<b>0.757</b>
TC <i>len</i> = 6	SB	0.655	0.273	0.718	0.484	0.768	0.727	0.618	0.724	0.616	0.620	0.656	0.549	0.601	0.725	0.663
	GP	0.663	<b>0.345</b>	0.665	0.845	<b>0.904</b>	0.675	<b>1.000</b>	0.958	0.975	0.781	0.893	0.558	0.755	0.817	0.768
	LR	0.673	0.250	0.683	0.739	0.697	0.710	0.982	0.887	0.918	0.726	0.822	0.535	0.766	0.798	0.758
	Ours	<b>0.813</b>	0.341	<b>0.809</b>	<b>0.850</b>	0.819	<b>0.863</b>	0.997	<b>0.962</b>	<b>0.977</b>	<b>0.826</b>	<b>0.911</b>	<b>0.654</b>	<b>0.830</b>	<b>0.913</b>	<b>0.866</b>
TC <i>len</i> = 11	SB	0.609	0.194	0.708	0.425	0.726	0.741	0.616	0.723	0.611	0.595	0.640	0.504	0.566	0.732	0.643
	GP	0.531	0.287	0.597	<b>0.789</b>	<b>0.833</b>	0.566	<b>1.000</b>	0.911	0.930	0.716	0.838	0.472	0.693	0.738	0.686
	LR	0.596	0.216	0.641	0.657	0.601	0.706	0.988	0.835	0.872	0.679	0.777	0.484	0.649	0.770	0.703
	Ours	<b>0.742</b>	<b>0.311</b>	<b>0.769</b>	0.782	0.734	<b>0.843</b>	0.998	<b>0.922</b>	<b>0.942</b>	<b>0.783</b>	<b>0.870</b>	<b>0.607</b>	<b>0.775</b>	<b>0.883</b>	<b>0.818</b>

Table 2. Quantitative evaluation and comparison with a tight tolerance threshold  $th = 1$  and different video length  $len = 2, 6, 11$ . SA: spatial accuracy. TC: temporal consistency. We compare our method with a scribble-based method (SB) [4], global plane tracking (GP) [1], and locally rigid motions (LR). Note that for  $len = 2$  the accuracy and consistency are the same. Our approach outperforms state-of-the-art methods in both accuracy and consistency.

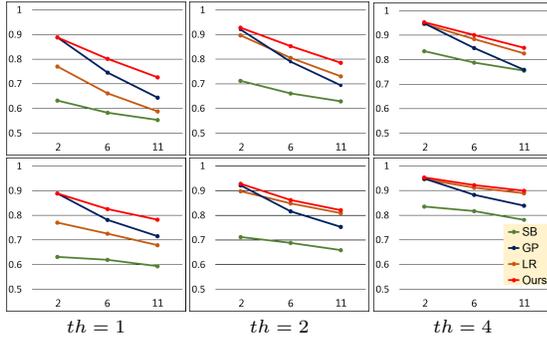


Figure 7. Quantitative evaluation for different video length (x-axis) and tolerance threshold ( $th = 1, 2, 4$ ). First row: accuracy. Second row: consistency. Our method performs the best.

handle partial boundaries (PB). (4) Tracking with locally rigid motion (LR) works reasonably well, but its performance is significantly worse than the full system, indicating the importance of the spatio-temporal optimization in our system. (5) Looking at the results for individual video sequences, we notice that (i) GP requires adequate texture for tracking; it fails in textureless regions (ToyHorse); (ii) SB works poorly on objects with long skinny structures (Tower); (iii) motion blur (MB) is a common obstacle to video object segmentation; our spatio-temporal smoothing scheme still generates consistent object boundaries.

We illustrate the overall performance with different thresholds  $th$  in Figure 7. Average performances over all video sequences and clips are reported. Our proposed framework behaves similarly well and is the best in both accuracy and consistency under different tolerance settings.

**Discussion on  $\nabla I$ .** In the case of strong shadow or motion blur, the object boundary could become vague, or even disappear completely, as shown in Figure 8. We have tried

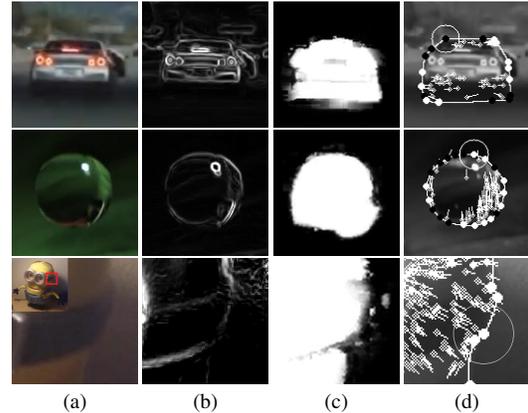


Figure 8. Demonstration of three hard cases with vague or missing boundaries. (a,b) Object boundaries can be easily affected by shadow, motion blur or background clutter. (c) Probabilistic foreground mask  $G$  obtained using SnapCut [4]. In these hard cases,  $G$  and  $\nabla G$  provide no meaningful improvement. (d) Our proposed framework keeps the local shape rigidity to mimic rotoscoping artists. Please refer to Figure 3 for the explanation.

using  $\nabla G$ , the gradient of the probabilistic foreground mask produced by SnapCut [4], to replace the image gradient  $\nabla I$ , but there is no meaningful improvement as the foreground mask itself is often erroneous in such cases (Figure 8c). In our system, weak edges and control points are regularized by their local as well as neighboring affinities. Once a control point or a Bézier curve is incorrectly snapped to a strong background edge, the resulting boundary shape will be constrained by Equation 6.

**Qualitative Evaluation.** Figure 9 shows visual comparisons of segmented boundaries in four video sequences. For GP, we can easily notice the boundary errors when there is occlusion (Sunset), or the object is deforming (Minion). SB produces zigzag and temporal inconsistency boundaries (ToyHorse and Car). Our method finds false object bound-

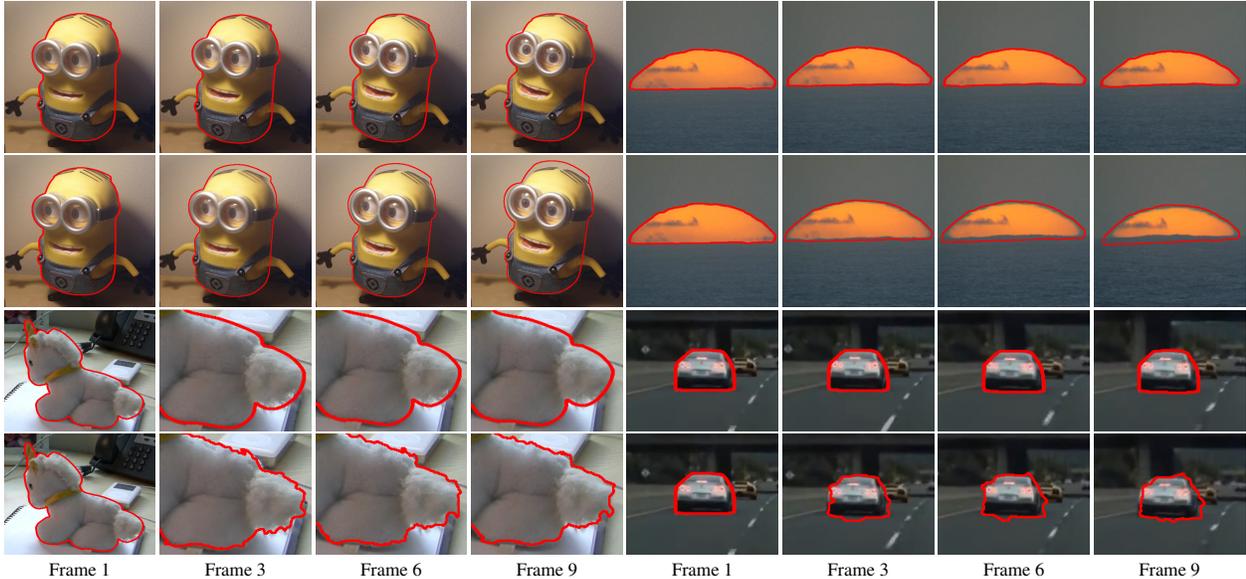


Figure 9. Qualitative evaluation and comparison. **First and third row**: our results. **Second row**: Rigid Mask Tracker [1]. **Last row**: SnapCut [4]. All methods start from the same annotation on the first frame. Images are cropped to suitable regions for visualization. In comparison, global shape tracking cannot handle non-rigid deformation (Minion) and occlusion (Sunset). Video SnapCut has lower quality for boundary smoothness and temporal consistency. Our method produces more accurate and consistent results.

	Seq 1	Minion	(sec)	Seq 2	Tower	(sec)
	SnapCut	Mocha	Ours	SnapCut	Mocha	Ours
User 1	147	1357	115	201	184	75
User 2	172	1223	135	326	311	84
User 3	255	1729	144	177	197	75

Table 3. User study on the efficiency of different video segmentation systems. Our system clearly outperforms the SnapCut [4] and the Mocha [2].

aries in some cases (Minion), but overall achieves higher quality and temporally consistent results.

Further video results and comparisons are shown in the project website.<sup>1</sup>

#### 4.4. Evaluating the User Interactions

We show in Figure 10 that our approach is convenient in precise boundary manipulation. The advantages are two-fold: (1) *Spatial adjustment*. Users can directly move the control points of Bézier curves to adjust the segmentation boundary. In contrast, scribble-based approaches require several rounds of interaction. Multiple scribbles need to be added near the true object boundary for further refinement. (2) *Temporal propagation*. Since scribble-based systems do not pose strong constraints on temporal shape stability, the same refinement is needed on multiple adjacent frames. On the contrary, our proposed framework produces more reliable results; once a refinement is done for one frame, the modified object shape can stay much longer.

Table 3 demonstrates a user study showing the efficiency of different video segmentation systems. We ask three users to segment two video clips, each with 20 frames in length

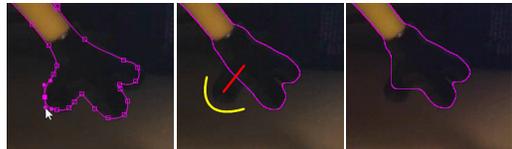


Figure 10. Our system allows direct boundary editing for segmentation error correction (left). In contrast, scribble-based systems require scribbles drawn near the object boundary (middle), however the result may still be unsatisfactory (right).

and initial contours not given. The users have the experience for more than ten hours in SnapCut and Mocha. We show them the groundtruth and ask them to achieve both accuracy and consistency. Results indicate that: (1) for SnapCut [4], although a rough foreground mask can be drawn efficiently, users spend most of the time refining the temporal boundary consistency, and (2) for segmenting the Minion clip with non-rigid motion, annotation is needed frequently in Mocha. Although our system requires more user input on the first frame, it produces better boundary curves with a greater degree of temporal stability, thus requiring less user intervention in the propagation process. Our method clearly outperforms SnapCut and Mocha in usability and efficiency.

## 5. Conclusion

We describe Coherent Parametric Contours, a boundary propagation framework for interactive video object segmentation, aiming to produce high quality object boundaries suitable for real video production. Compared with traditional scribble-based methods, it generates accurate and temporal coherent boundaries and supports direct and natural boundary editing. We also provide a new dataset and a new metric to measure the temporal boundary consistency.

<sup>1</sup><http://yao.lu/CPC.html>

## References

- [1] Adobe After Effects. <http://www.adobe.com/products/aftereffects.html>.
- [2] Mocha Software. <http://www.imagineersystems.com/>.
- [3] A. Agarwala, A. Hertzmann, D. Salesin, and S. Seitz. Keyframe-based tracking for rotoscoping and animation. *TOG*, 23(3):584–591, 2004.
- [4] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. *TOG*, 28(3):70, 2009.
- [5] K. Bowyer, C. Kranenburg, and S. Dougherty. Edge detector evaluation using empirical roc curves. In *CVPR*, 1999.
- [6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *IJCV*, 22(1):61–79, 1997.
- [7] P. Elias, A. Feinstein, and C. E. Shannon. A note on the maximum flow through a network. *T. on Information Theory*, 2(4):117–119, 1956.
- [8] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, 1981.
- [9] K. He, C. Rhemann, C. Rother, X. Tang, and J. Sun. A global sampling method for alpha matting. In *CVPR*, 2011.
- [10] S. Jehan-Besson, M. Barlaud, and G. Aubert. Video object segmentation using eulerian region-based active contours. In *ICCV*, 2001.
- [11] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988.
- [12] S.-Y. Lee, J.-C. Yoon, and I.-K. Lee. Temporally coherent video matting. *Graphical Models*, 72(3):25–33, 2010.
- [13] Y. Lu, W. Zhang, H. Lu, and X. Xue. Salient object detection using concavity context. In *ICCV*, 2011.
- [14] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *TPAMI*, 26(5):530–549, 2004.
- [15] J. J. Moré. The levenberg-marquardt algorithm: implementation and theory. *Numerical analysis*, pages 105–116, 1978.
- [16] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *Conf. on Computer graphics and interactive techniques*, 1995.
- [17] P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011.
- [18] F. Precioso and M. Barlaud. B-spline active contour with handling of topology changes for fast video segmentation. *EURASIP J. on Applied Signal Processing*, 2002(1):555–560, 2002.
- [19] B. L. Price, B. S. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *CVPR*, 2009.
- [20] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *TOG*, 23(3):309–314, 2004.
- [21] Y. Shi and W. C. Karl. Real-time tracking using level sets. In *CVPR*, 2005.
- [22] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. *TOG*, 24(3):585–594, 2005.
- [23] J. Wang and M. F. Cohen. Optimized color sampling for robust matting. In *CVPR*, 2007.
- [24] W. Wolf. Key frame selection by motion analysis. In *ICASSP*, 1996.
- [25] S. X. Yu and J. Shi. Segmentation given partial grouping constraints. *TPAMI*, 26(2):173–183, 2004.
- [26] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013.
- [27] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Summary transfer: Exemplar-based subset selection for video summarization. In *CVPR*, 2016.
- [28] F. Zhong, X. Qin, Q. Peng, and X. Meng. Discontinuity-aware video object cutout. *TOG*, 31(6):175, 2012.