ROAM: a Rich Object Appearance Model with Application to Rotoscoping

Ondrej Miksik, Juan-Manuel Pérez-Rúa, Philip H S Torr, Patrick Pérez

ROAM: a Rich Object Appearance Model with Application to Rotoscoping

Ondrej Miksik[1*] Juan-Manuel Pérez-Rúa[2*] Philip H. S. Torr[1] Patrick Pérez[2]
[1]University of Oxford
[2]Technicolor Research & Innovation

Figure 1: ROAM **for video object segmentation**. Designed to help *rotoscoping*, the proposed object appearance model allows the automatic delineation of a complex object in a shot, starting from an initial outline provided by the user.

Abstract

Rotoscoping, the detailed delineation of scene elements through a video shot, is a painstaking task of tremendous importance in professional post-production pipelines. While pixel-wise segmentation techniques can help for this task, professional rotoscoping tools rely on parametric curves that offer the artists a much better interactive control on the definition, editing and manipulation of the segments of interest. Sticking to this prevalent rotoscoping paradigm, we propose a novel framework to capture and track the visual aspect of an arbitrary object in a scene, given a first closed outline of this object. This model combines a collection of local foreground/background appearance models spread along the outline, a global appearance model of the enclosed object and a set of distinctive foreground landmarks. The structure of this rich appearance model allows simple initialization, efficient iterative optimization with exact minimization at each step, and on-line adaptation in videos. We demonstrate qualitatively and quantitatively the merit of this framework through comparisons with tools based on either dynamic segmentation with a closed curve or pixel-wise binary labelling.

1. Introduction

Modern high-end visual effects (vfx) and post-production rely on complex workflows whereby each shot undergoes a succession of artistic operations. Among those, rotoscoping is probably the most ubiquitous and demanding one [7, 18]. Rotoscoping amounts to outlining accurately one or several scene elements in each frame of a shot. This is a key operation for compositing [30] (insertion of a different background, whether natural or synthetic), where it serves as an input to subsequent operations such as matting and motion blur removal.[1] Rotoscoping is also a pre-requisite for other important operations, such as object colour grading, rig removal and new view synthesis, with large amounts of elements to be handled in the latter case.

Creating such binary masks is a painstaking task accomplished by trained artists. It can take up to several days of work for a complex shot of only a few seconds, using dedicated tools within video editing softwares like Silhouettefx, Adobe After Effect, Autodesk Flame or The Foundry Nuke. As discussed in [18], professional roto artists use mostly tools based on *roto-curves*, *i.e.*, parametric closed curves that can be easily defined, moved and edited throughout shots. By contrast, these artists hardly use brush-based tools, even if empowered by local appearance modelling, graph-based regularization and optic flow-based tracking as After Effect's ROTOBRUSH.

*Assert joint first authorship.
J-M is also with Inria (Centre Rennes - Bretagne Atlantique, France).
Code available at http://www.miksik.co.uk
This work was supported by Technicolor, EPSRC, ERC grant ERC-2012-AdG 321162-HELIOS, EPSRC grant Seebibyte EP/M013774/1, EPSRC/-MURI grant EP/N019474/1, ANRT and Cifre convention No. 2014/1041.

[1]The use of blue or green screens on set can ease compositing but remains a contrived set-up. Even if accessible, such screens lead to chroma-keying and de-spilling operations that are not trivial and are not suited to all foreground elements, thus rotoscoping remains crucial.

Due to its massive prevalence in professional workflows, we address here rotoscoping in its closed contour form, which we aim to facilitate. Roto-curves being interactively placed in selected keyframes, automation can be sought either at the key-frame level (reducing the number of user's inputs) or at the tracking level (reducing the number of required key-frames). In their recent work, Li *et al*. [18] offers with ROTO++, a tool that helps on both fronts, thanks to an elegant shape modelling.

In the present work, we explore a complementary route that focuses on the automatic tracking from a given keyframe. In essence, we propose to equip the roto-curve with a rich, adaptive modelling of the appearance of the enclosed object. This model, coined ROAM for Rich Online Appearance Model, combines in a flexible way various appearance modelling ingredients: (i) Local foreground/background colour modelling, in the spirit of VIDEO SNAPCUT [3] but attached here to the roto-curve; (ii) Fragment-based modelling to handle large displacements and deformations and (iii) Global appearance modelling, which has proved very powerful in binary segmentation with graph cuts, *e.g*. in [6].

We would like to emphasize that our model is the first that combines local appearance models along the closed contour with global appearance model of the enclosed object using discrete Green theorem, and pictorial structure to capture locally rigid deformations, in a principled structured prediction framework. As demonstrated on recent benchmarks, ROAM outperforms state-of-art approaches when a single initial roto-curve is provided. It is in particular less prone to spurious changes of topology that lead to eventual losses than After Effect ROTOBRUSH, and more robust than ROTO++ [18] in the absence of additional user inputs. This robustness makes it appealing to facilitate rotoscoping, either as a standalone tool, or combined with existing curve-based tools such as ROTO++.

2. Related work and motivation

Rotoscoping is a form of interactive "video object"[2] segmentation. As such, the relevant literature is vast. For sake of brevity, we focus mostly our discussion on works that explicitly target rotoscoping or very similar scenarios.

2.1. Rotoscoping and curve-based approaches

Li *et al*. [18] recently released a very detailed study of professional rotoscoping workflow. They first establish that trained artists mostly use parametric curves such as Bezier splines to delineate objects of interest in key-frames, "track" them from one frame to the next, edit them at any stage of the pipeline and, last but not least, pass them in a compact and manipulable format to the next stage of the vfx pipeline, *e.g*., to the compo-artists. Professional rotoscoping tools such as Silhouettefx, Blender, Nuke or Flame are thus based on parametric curves, which can be either interpolated between key-frames or tracked with a homographic "planar tracker" when suitable. Sticking to this ubiquitous workflow, the authors propose ROTO++ to speed it up. Bezier roto-curves defined by the artist in the selected key-frames allow the real-time learning of a non-linear low-dimensional shape space based on a Gaussian process latent variable model. Shape tracking between key-frames, as well as subsequent edits, are then constrained within this smooth manifold (up to planar transforms), with substantial gains in work time. Our work is fully complementary to ROTO++: while ROAM does not use a strong shape prior in its current form, it allows to capture the dynamic appearance of the video object, something that ROTO++ does not address.

In their seminal rotoscoping work, Agarwala *et al*. [1] proposed a complete interactive system to track and edit Bezier roto-curves. It relies on the popular active contour framework [5, 14]: a curve, parametrized by control points, finely discretized and equipped with a second-order smoothness prior is encouraged to evolve smoothly and snap to strong image edges. Their energy-based approach also uses local optical flow along each side of the shape's border. In contrast to this work, our approach offers a richer local appearance modelling along the roto-shape as well as additional intra-object appearance modelling.

Similarly to [1], Lu *et al*. [19] recently introduced an interactive object segmentation system called "coherence parametric contours" (CPC), which combines planar tracking with active contours. Our system includes similar ingredients, with the difference that the planar tracker is subsumed by a fragment-based tracker and that the appearance of the object and of its close surrounding is also captured and modeled. We demonstrate the benefits of these additional features on the evaluation dataset introduced by Lu *et al*. [19].

2.2. Masks and region-based approaches

Other notable approaches to interactive video segmentation address directly the problem of extracting binary masks, *i.e*. labelling pixels of non-keyframes as foreground or background. As discussed in [18, 19], a region-based approach is less compatible with professional rotoscoping, yet provides powerful tools. Bai *et al*. [3] introduced VIDEO SNAPCUT, which lies at the heart of After Effect's ROTOBRUSH. Interaction in VIDEO SNAPCUT is based on foreground/background brushes, following the popular scribble paradigm of Boykov and Jolly [6]. The mask available in a given frame is tracked to the next frame through the propagation of local windows that straddle its border. Each window is equipped with a local foreground/background colour model and a local shape template, both updated through time. After propagation along an object-centric optical-flow, these

[2]Throughout, "video object", or simply "object", is a generic term to designate a scene element of interest and the associated image region in the video.

windows provide suitable pixel-wise unaries that are fed to a classic graph-cut. This approach provides a powerful way to capture on-the-fly local colour models and combine them adaptively with some shape persistence. However, being based on graph-cut (pixel-wise labelling), ROTOBRUSH can be penalized by its extreme topology flexibility: as will be showed in the experiments, rapid movements of the object, for instance, can cause large spurious deformations of the mask that can eventually lead to complete losses in the absence of user intervention. In ROAM, we take inspiration from the local colour modelling at the object's border and revisit it in a curve-based segmentation framework that allows tighter shape control and easier subsequent interaction.

More recently, Fan *et al*. introduced JUMPCUT [10], another mask-based approach where frame-to-frame propagation is replaced by mask transfer from the key-frame(s) to distant frames. This long-range transfer leverages dense patch correspondences computed over the inside and outside of the known mask, respectively. The transfered mask is subsequently refined using a standard level set segmentation (region encoded via a spatial map). A salient edge classifier is trained online to locate likely fragments of object's new silhouette and drive the level set accordingly. They reported impressive results with complex deformable objects going through rapid changes in scene foreground. However, similarly to ROTOBRUSH, this agility might also become a drawback in real rotoscoping scenarios, as is the lack of shape parametrization. Also, the underlying figure/ground assumption (the object is moving distinctly in front of a background) is not met in many cases, *e.g*. rotoscoping of a static scene element or of an object in a dynamic surrounding.

3. Introducing ROAM

Our model consists of a graphical model with the following components: (i) a closed curve that defines an object and a collection of local foreground/background[3] appearance models along it; (ii) a global appearance model of the enclosed object; and (iii) a set of distinctive object's landmarks. While the global appearance model captures image statistics as in graph-cut approaches [6, 25], it is the set of local fg/bg appearance models placed along the boundary that enables accurate object delineation. The distinctive object's landmarks organized in a star-shaped model (Fig. 3, left) help to prevent the contour from sliding along itself and to control the level of non-rigid deformations. The landmarks are also used to robustly estimate a rigid transformation between the frames to "pre-warp" the contour, which significantly speeds-up the inference. In addition, the control points of the roto-curve, as well as the local fg/bg models and the landmarks are maintained through time, which provides us with different types of temporal correspondences.

[3] "Foreground/background" terminology, "fg/bg" in short, merely refers here to inside and outside of the roto-curve; it does not imply that the object stands at the forefront of the 3D scene with a background behind it.

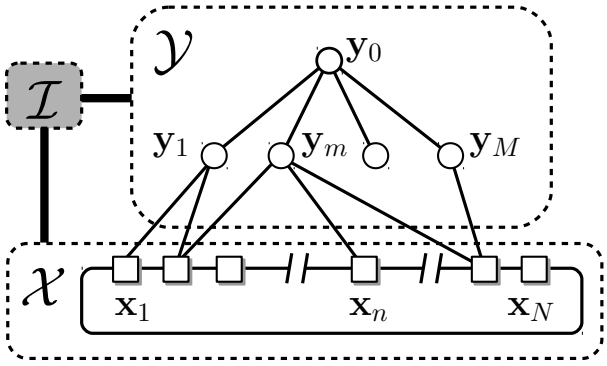

Figure 2: **Graphical model of** ROAM. In joint model defined by energy $E(\mathcal{X}, \mathcal{Y}; \mathcal{I})$ in (1), contour node variables (white squares) form a closed 1-st order chain conditioned on image data (grey box) and landmark variables (white circles), the latter variables forming a shallow tree conditioned on all others.

Given a colour image $\mathcal{I} = \{\mathbf{I_p}\}_{\mathbf{p}\in\mathcal{P}}$, a conditional graphical model (Fig. 2) is defined through the energy function

$$E(\mathcal{X}, \mathcal{Y}; \mathcal{I}) := E^C(\mathcal{X}; \mathcal{I}) + E^L(\mathcal{Y}; \mathcal{I}) + E^J(\mathcal{X}, \mathcal{Y}), \quad (1)$$

where E^C and E^L depend only on the roto-curve configuration $\mathcal{X}$ and the landmarks configuration $\mathcal{Y}$ respectively, and E^J links the two together (independently of the image). In the following, we describe these three energy terms in detail.

3.1. Curve-based modelling: E^C

While Bezier splines are a popular representation for rotoscoping [1, 18], we simply consider polygonal shapes here: roto-curve $\mathcal{X}$ is a polyline with N vertices $\mathbf{x}_1 \dots \mathbf{x}_N \in \mathbb{Z}^2$ and N non-intersecting edges $\mathbf{e}_n = (\mathbf{x}_n, \mathbf{x}_{n+1})$, where $\mathbf{x}_{N+1}$ stands for $\mathbf{x}_1$, *i.e*. the curve is closed. Given an orientation convention (*e.g*. clockwise), the interior of this curve defines a connected subset $R(\mathcal{X}) \subset \mathcal{P}$ of the image pixel grid (Fig. 3, left), which will be denoted R in short when allowed by the context.

Energy E^C is composed of two types of edge potentials ψ_n^{loc} and ψ_n^{glob} that relate to local and global appearance respectively:

$$E^C(\mathcal{X}; \mathcal{I}) := \sum_{n=1}^{N} [\psi_n^{\text{loc}}(\mathbf{e}_n) + \psi_n^{\text{glob}}(\mathbf{e}_n)]. \quad (2)$$

As with classic active contours [14], the first type of potential will encapsulate both a simple ℓ_2-regularizer that penalizes stretching and acts as a curve prior (we are not using second-order smoothing in the current model), and a data term that encourages the shape to snap to strong edges. It will in addition capture colour likelihood of pixels on each side of each edge via local appearance models. The second set of potentials results from the transformation of object-wise

colour statistics (discrete surface integral) into edge-based costs (discrete line integrals).

Note that, since we do not impose any constraint on the various potentials, the one specified below could be replaced by more sophisticated ones, *e.g.* using semantic edges [8] instead of intensity gradients, or using statistics of convolutional features [12] rather than colour for local and global appearance modelling.

Local appearance model. Each edge $\mathbf{e}_n$ is equipped with a local appearance model $p_n = (p_n^{\mathrm{f}}, p_n^{\mathrm{b}})$ composed of a fg/bg colour distribution and of a rectangular support R_n, with the edge as medial axis and a fixed width in the perpendicular direction (Fig. 3, right). Denoting R_n^{in} and R_n^{out} the two equal-sized parts of R_n that are respectively inside and outside R, we construct a simple edge-based energy term (the smaller, the better) that rewards edge-configurations such that colours in R_n^{in} (resp. R_n^{out}) are well explained by model p_n^{f} (resp. p_n^{g}) and edge $\mathbf{e}_n$ is short and goes through high intensity gradients:

$$\psi_n^{\mathrm{loc}}(\mathbf{e}_n) := -\sum_{\mathbf{p}\in R_n^{\mathrm{in}}} \ln p_n^{\mathrm{f}}(\mathbf{I_p}) - \sum_{\mathbf{p}\in R_n^{\mathrm{out}}} \ln p_n^{\mathrm{b}}(\mathbf{I_p}) + \mu\|\mathbf{x}_{n+1}-\mathbf{x}_n\|^2 - \sum_{\mathbf{p}\in\mathbf{e}_n} \lambda\|\nabla\mathcal{I}(\mathbf{p}))\|^2, \quad (3)$$

with μ and λ two positive parameters.

Global appearance model. A global appearance model captures image statistics over the object's interior. As such, it also helps pushing the roto-curve closer to the object's boundary, especially when local boundary terms are not able to explain foreground and background reliably. Defining $p_0 = (p_0^{\mathrm{f}}, p_0^{\mathrm{b}})$ the global fg/bg colour distribution, the bag-of-pixel assumption allows us to define region energy term

$$\sum_{\mathbf{p}\in R} \ln \frac{p_0^{\mathrm{b}}(\mathbf{I_p})}{p_0^{\mathrm{f}}(\mathbf{I_p})}. \quad (4)$$

This discrete region integral can be turned into a discrete contour integral using one form of discrete Green theorem [27]. Using horizontal line integrals for instance, we get

$$\sum_{\mathbf{p}\in R} \ln \frac{p_0^{\mathrm{b}}(\mathbf{I_p})}{p_0^{\mathrm{f}}(\mathbf{I_p})} = \sum_{n=1}^{N} \underbrace{\sum_{\mathbf{p}\in\mathbf{e}_n} \alpha_n(\mathbf{p}) Q(\mathbf{p})}_{:=\psi_n^{\mathrm{glob}}(\mathbf{e}_n)}, \quad (5)$$

where $Q(\mathbf{p}) = \sum_{\text{“}\mathbf{q}\leqslant\mathbf{p}\text{”}} \ln(p_0^{\mathrm{b}}(\mathbf{I_p})/p_0^{\mathrm{f}}(\mathbf{I_p}))$ is the discrete line integral over pixels to the left of $\mathbf{p}$ on the same row, and $\alpha_n(\mathbf{p}) \in \{-1, +1\}$ depends on the direction and orientation, relative to curve's interior, of the oriented edge $\mathbf{e}_n$. In (5), the second sum in r.h.s. is taken over the pixel chain resulting from the discretization of the line segment $[\mathbf{x}_n, \mathbf{x}_{n+1}]$ with the final vertex excluded to avoid double-counting.

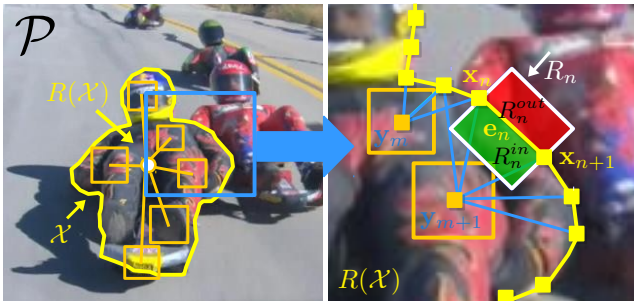

Figure 3: **Structure and notations of proposed model**. (Left) A simple closed curve $\mathcal{X}$ outlines the object region $R(\mathcal{X})$ in the image plane $\mathcal{P}$. Several landmarks, forming a star-shaped graphical model, are defined in this region. (Right) Each edge $\mathbf{e}_n$ of the closed polyline defines a region R_n that staddles $R(\mathcal{X})$; each node $\mathbf{x}_n$ of the polyline is possibly connected to one or several landmarks.

3.2. Landmark-based modelling: E^L

Our model also makes use of a set $\mathcal{Y}$ of M distinctive landmarks $\mathbf{y}_1 \dots \mathbf{y}_M \in R(\mathcal{X})$ detected inside the object of interest. Similarly to pictorial structures [11], these landmarks form the leaves of a star-shaped graphical model[4] with a virtual root-node $\mathbf{y}_0$. This part of the model is defined by leaf potentials $\phi_m(\mathbf{y}_m)$ and leaf to root potentials $\varphi_m(\mathbf{y}_0, \mathbf{y}_m)$:

$$E^L(\mathcal{Y};\mathcal{I}) := \sum_{m=1}^{M} \phi_m(\mathbf{y}_m) + \sum_{m=1}^{M} \varphi_m(\mathbf{y}_0, \mathbf{y}_m). \quad (6)$$

Each landmark is associated with a model, *e.g.* a template or a filter, that allows the computation of a matching cost at any location in the image. The leave potential $\phi_m(\mathbf{y}_m)$ corresponds to the negative matching cost for m-th landmark. The pairwise potentials φ_m penalize the difference in ℓ_2-norm between the current configuration and the one, $\hat{\mathcal{Y}}$, estimated in previous frame:

$$\varphi_m(\mathbf{y}_0, \mathbf{y}_m) = \frac{1}{2}\|\mathbf{y}_m - \mathbf{y}_0 - \hat{\mathbf{y}}_m + \hat{\mathbf{y}}_0\|^2. \quad (7)$$

3.3. Curve-landmarks interaction: E^J

The joint energy $E^J(\mathcal{X}, \mathcal{Y})$ captures correlation between object's outline and object's landmarks. Based on proximity, shape vertices and landmarks can be associated. Let denote $n \sim m$ the pairing of vertex $\mathbf{x}_n$ with landmark $\mathbf{y}_m$. Energy term E^J decomposes over all such pairs as:

$$E^J(\mathcal{X}, \mathcal{Y}) = \sum_{n\sim m} \xi_{nm}(\mathbf{x}_n, \mathbf{y}_m). \quad (8)$$

For each pair $n \sim m$, the interaction potential is defined as:

$$\xi_{mn}(\mathbf{x}_n, \mathbf{y}_m) = \frac{1}{2}\|\mathbf{x}_n - \mathbf{y}_m - \boldsymbol{\mu}_{mn}\|^2, \quad (9)$$

where $\boldsymbol{\mu}_{mn}$ is the landmark-to-vertex shift vector in the first image.

[4] The star shape is used for its simplicity but could be replaced by another tree-shaped structure.

4. Using ROAM

Sequential alternating inference. Using ROAM to outline the object of interest in a new image amounts to solving the discrete optimization problem:

$$\min_{\mathcal{X},\mathcal{Y}} E(\mathcal{X},\mathcal{Y};\mathcal{I}), \tag{10}$$

where E is defined by (1) and depends on previous curve/landmarks configuration $(\hat{\mathcal{X}},\hat{\mathcal{Y}})$ through several of its components. Despite this problem could be formulated as an integer linear program, we opt for simpler alternating optimization with exact minimization at each step which converges within a few iterations.

In the first step, we fix the roto-curve $\mathcal{X}$ and find the best configuration of landmarks $\mathcal{Y}$ using dynamic programming. Exact solution for such a problem can be obtained in two passes, solving exactly

$$\min_{\mathbf{y}_0} \min_{\mathbf{y}_{1:M}} \sum_{m=1}^{M} \big(\phi_m(\mathbf{y}_m) + \varphi_m(\mathbf{y}_0,\mathbf{y}_m) + \sum_{n\sim m} \xi_{mn}(\mathbf{x}_n,\mathbf{y}_m)\big). \tag{11}$$

Default implementation leads to complexity $\mathcal{O}(MS^2)$, with S the size of individual landmark state-spaces, *i.e.* the number of possible pixel positions allowed for each. However, the quadratic form of the pairwise terms allows making it linear in the number of pixels, *i.e.* $\mathcal{O}(MS)$, by resorting to generalized distance transform [11].

In the second step, we fix the landmarks $\mathcal{Y}$ and find the best configuration of contour $\mathcal{X}$. This is a classic first-order active contour problem. Allowing continuous values for nodes coordinates, a gradient descent can be conducted with all nodes being moved simultaneously at each iteration. We prefer the discrete approach, whereby only integral positions are allowed and dynamic programming can be used [2]. In that formulation, exact global inference is theoretically possible, but with a prohibitive complexity of $\mathcal{O}(NP^3)$, where $P=\mathrm{card}(\mathcal{P})$ is the number of pixels in images. We follow the classic iterative approach that considers only D possible moves $\Delta\mathbf{x}$ for each node around its current position. For each of the D positions of first node $\mathbf{x}_1$, Viterbi algorithm provides the best moves of all others in two passes and with complexity $\mathcal{O}(ND^2)$. Final complexity is thus $\mathcal{O}(ND^3)$ for each iteration of optimal update of previous contour, solving:

$$\min_{\Delta\mathbf{x}_1} \min_{\Delta\mathbf{x}_{2:N}} \sum_{n=1}^{N} \big(\psi_n^{\mathrm{loc}}(\mathbf{e}_n+\Delta\mathbf{e}_n) + \psi_n^{\mathrm{glob}}(\mathbf{e}_n+\Delta\mathbf{e}_n) + \sum_{m\sim n} \xi_{mn}(\mathbf{x}_n+\Delta\mathbf{e}_n,\mathbf{y}_m)\big). \tag{12}$$

Note that sacrifying optimality of each update, the complexity could even been reduced as much as $\mathcal{O}(ND)$ [29].

Given some initialization for $(\mathcal{X},\mathcal{Y})$, we thus alternate between two *exact* block-wise inference procedures. This guaranties convergence toward a local minima of joint energy $E(\mathcal{X},\mathcal{Y};\mathcal{I})$. Also, the complexity of each iteration is linear in the number of vertices and landmarks, linear in the number of pixels, and cubic in the small number of allowed moves for a curve's vertex.

Online learning of appearance models. Local fg/bg colour models p_ns and global colour model p_0 are GMMs. Given the roto-curve in the initial frame, these GMMs are first learned over region pairs $(R_n^{\mathrm{in}}, R_n^{\mathrm{out}})$s and $(R, \mathcal{P}\setminus R)$ respectively and subsequently adapted through time using Stauffer and Grimson's classic technique [26].

Selection and adaption of landmarks. A pool of distinctive landmarks is maintained at each instant. They can be any type of classic interest points. In order to handle texture-less objects, we use maximally stable extremal regions (MSERs) [21]. Each landmark is associated with a correlation filter whose response over a given image area can be computed very efficiently [13]. At any time, landmarks whose filter response is too ambiguous are deemed insufficiently discriminative and removed from the current pool in the same way tracker loss is monitored in [13]. The collection is re-populated through new detections. Note that correlation filters can be computed over arbitrary features and kernelized [13]; for simplicity, we use just grayscale features without kernel function.

Allowing topology changes. Using a closed curve is crucial to comply with rotoscoping workflows and allows the definition of a rich appearance model. Also, it prevents abrupt changes of topology. While this behavior is overall beneficial (See §5), segmenting a complete articulated 3D object as in Fig. 1 might turn difficult. Roto-artists naturally handle this by using multiple independent roto-curves, one per meaningful part of the object. As an alternative for less professional, more automatic scenarios, we propose to make ROAM benefit from the best of both worlds: standard graph-cut based segmentation [6], with its superior agility, is used to *propose* drastic changes to current curve, if relevant. Space-dependent unaries are derived in ad-hoc way from both global and local colour models and combined with classic contrast-dependent spatial regularization.[5] The exact minimizer of this instrumental cost function is obtained through graph-cut (or its dynamic variant for efficiency [16]) and compared to the binary segmentation associated to the current shape $\mathcal{X}$. At places where the two differ significantly, a modification of current configuration (displacement, removal or addition of vertices) is proposed and accepted if it reduces the energy $E(\mathcal{X},\mathcal{Y};\mathcal{I})$.

[5] Note that this instrumental energy is too poor to compete on its own with proposed model, but is a good enough proxy for the purpose of proposing possibly interesting new shapes at certain instants. It is also very different from the one in the final graph-cut of VIDEO SNAPCUT where unaries are based on the already computed soft segmentation to turn it into a hard segmentation. Also, graph-cut segmentation is the final output in VIDEO SNAPCUT, unless further interaction is used, while we only use it to explore alternative topologies under the control of our joint energy model.

5. Results

We report experimental comparisons that focus on the minimum input scenario: an initial object selection (curve or mask, depending on the tool) is provided to the system and automatic object segmentation is produced in the rest of the shot.[6] We do not consider additional user interactions.

Datasets. We evaluate our approach on the recent CPC rotoscoping dataset [19]. It contains 9 videos consisting of 60 to 128 frames which represent typical length of shots for rotoscoping. These sequences were annotated by professional artists using standard post-production tools. We also provide qualitative results on shots from the ROTO++ [18] dataset for which the authors have not released the ground-truth yet, as well as from the VIDEO SNAPCUT dataset [3].

In addition to that, we use the DAVIS video segmentation dataset [22] which comprises 50 challenging sequences with a wide range of difficulties: large occlusions, long-range displacements, non-rigid deformations, camouflaging effects and complex multi-part objects. Let us note that this dataset is intended to benchmark pixel-based video segmentation methods, not rotoscoping tools based on roto-curves.

Evaluation Metrics. We use standard video segmentation evaluation metrics and report the average *accuracy*, *i.e.*, the proportion of ground-truth pixels that are correctly identified, and the more demanding average *intersection-over-union* (IoU), *i.e.*, the area of the intersection of ground-truth and extracted objects over the area of their union. We also report runtimes and evolution of IoU as sequences proceed.

Baselines. We compare with several state-of-the-art methods. Our main comparison is with recent approaches that rely on a closed-curve, *i.e.*, CPC [19] and ROTO++ [18]. We initialize all methods with the same object and measure their performance over the rest of each sequence. Since ROTO++ requires at least two key-frames to benefit from its online shape model, we report scores with letting the method access the ground-truth of the last frame as well. We also run it with the initial keyframe only, a configuration in which ROTO++ boils down to the Blender planar tracker.

In addition to that, we also compare with two approaches based on pixel-wise labelling: JUMPCUT [10] and VIDEO SNAPCUT [3] as implemented in After Effect ROTOBRUSH and three recent video-segmentation approaches [9, 20, 28]. As a naive baseline, we use a combination of a bounding-box tracker [13] and GRABCUT [25].

Ablation study. To evaluate the importance of each part of our model, we consider 4 different configurations:

- Baseline: negative gradient with ℓ_2-regularizer;
- Lean: baseline + local appearance model;
- Medium: lean + landmarks;
- Full: medium + automatic re-parametrization and global appearance model;

[6] Video results are available at https://youtu.be/UvO7IacS9pQ

Table 1: Quantitative evaluation on CPC dataset (*: partial evaluation only, see text)

Method	Avg. Accuracy	Avg. IoU	Time (s) / frame min	max	avg
GCUT [25] + KCF [13]	.891	.572	0.394	0.875	0.455
AE ROTOBRUSH [3]	.992	.895	—	—	—
ROTO++(1 keyframe) [18]	.969	.642	—	—	0.108
ROTO++(2 keyframes) [18]	.974	.691	—	—	0.156
CPC [19]	.998*	.975*	—	—	—
NLCV [9]	.896*	.194*	—	—	—
BSVS [20]	.991	.872	—	—	—
OBJECTFLOW [28]	.968	.502	—	—	—
ROAM: Baseline Conf.	.993	.932	0.011	0.155	0.040
ROAM: Lean Conf.	**.995**	.938	0.092	0.377	0.102
ROAM: Medium Conf.	**.995**	.939	0.279	0.875	0.652
ROAM: Full Conf.	**.995**	**.951**	0.874	8.78	3.052

For all configurations, we used cross-validation (maximizing the mean IoU) on the training fold of the DAVIS dataset to set the parameters and kept them fixed for all experiments.

Quantitative results. The quantitative results for the CPC dataset are summarized in Tab. 1. While average accuracy is quite similar and saturated for all methods, all configurations of ROAM outperform all baselines. In terms of IoU, all versions of ROAM outperform significantly all others with the full configuration being the best. The reason why landmarks ("medium conf.") do not add much to ROAM is that the CPC dataset does not exhibit many large displacements. The CPC method [19] was evaluated only on the first ten frames of each sequence since their authors have released results only on these frames and not yet their code. Hence, the scores reported in Tab. 1 for CPC are based on partial videos, as opposed to the scores for all the other methods (including ours). When similarly restricted to the first 10 frames, ROAM performs on par with CPC for all the sequences except "drop" sequence. This sequence shows a water drop falling down – a transparent object, making color models (both local and global) useless if not harmful, and exhibits a very smooth round shape. For this sequence, the CPC method [19] performs better since it uses Bézier curves and relies solely on the strength of the image gradients.

Results for the DAVIS dataset are reported in Tab. 2. While our method is on par with JUMPCUT (pixel-wise labelling), we again significantly outperform ROTO++ by almost 25 IoU points (note that using ROTO++ with only two keyframes is not a typical scenario, however, this shows how complementary our approaches are). Despite [20] is better by 100 and [28] by 17 IoU points on DAVIS, our model outperforms [20] by 80 and [28] by 450 points on the CPC. In other words, our approach should in the worst case be considered on par. However, we would like to stress that [9, 20, 28] are not our (main) competitors since all are based on pixel-wise labelling and as such **cannot** provide the same flexibility for rotoscoping as the closed contour counter-

Table 2: Quantitative comparisons on DAVIS dataset

Method	Average Accuracy			Average IoU			Time / frame (s)		
	Validation set	Training set	Full set	Validation set	Training set	Full set	min	max	avg
GRABCUT [25] + Tracker [13]	0.896	0.914	0.907	0.277	0.296	0.289	0.405	0.675	0.461
JUMPCUT [10]	0.952	0.957	0.956	0.570	0.632	0.616	—	—	—
AE ROTOBRUSH [3]	0.951	0.942	0.946	0.533	0.479	0.500	—	—	—
ROTO++ (single keyframe)[18]	0.910	0.922	0.917	0.248	0.310	0.284	—	—	0.118
ROTO++ (two keyframes) [18]	0.925	0.933	0.930	0.335	0.394	0.358	—	—	0.312
NLCV [9]	0.948	0.963	0.957	0.551	0.701	0.641	—	—	—
BSVS [20]	**0.966**	**0.974**	**0.971**	**0.683**	0.709	0.665	—	—	—
OBJECTFLOW [28]	—	—	—	0.600	**0.732**	**0.711**	—	—	—
ROAM: Baseline Conf.	0.930	0.937	0.932	0.358	0.385	0.377	0.017	0.113	0.049
ROAM: Lean Conf.	0.935	0.937	0.936	0.409	0.417	0.412	0.187	0.641	0.342
ROAM: Medium Conf.	0.942	0.952	0.948	0.532	0.591	0.564	0.302	1.785	0.746
ROAM: Full Conf.	0.952	0.956	0.953	0.583	0.624	0.615	0.546	7.952	3.058

Table 3: Different types of contour warping for handling long displacements on a subset of sequences of the DAVIS dataset.

Warping method	Average Accuracy	Average IoU
Optical flow	0.878	0.312
Node projection from landmark tracking	0.906	0.480
Robust rigid transf. from landmarks	**0.934**	**0.581**

part [18]. Note, that we could not provide more quantitative comparisons since results/implementations of other methods were not available from the authors. In particular, comparison with the CPC method [19] would be interesting since the DAVIS dataset [22] exhibits many large displacements and major topology changes.

Comparing the different configurations of ROAM – local appearance models add 3 points, landmarks 15 and global model with re-parametrization another 5 points – demonstrates the importance of all components of our framework. To examine behaviour of each method in detail, we report IoU score for each frame in Fig. 7, with in addition the effect of varying the size of the displacement space in ROAM (from windows of 3×3 to 13×13 pixels) represented with a blue shadow. It can be seen that ROAM is more robust in time, not experiencing sudden performance drops as others.

Importance of landmarks and warping. Using alternating optimization has one more benefit. We can use the predicted position of landmarks in the next frame to estimate the transformation between the two and "warp" the contour to the next frame. This allows us to reduce the number of D possible moves of nodes which i) significantly speeds-up the algorithm, ii) allows us to handle large displacement and iii) allows to better control non-rigid deformations.

We have experimented with three settings for warping of contour: using a smoothed optical flow masked as in [23], moving each node by averaging the motion of all landmarks connected to given node and robustly estimating similarity transformation with RANSAC from position of landmarks. Table 3 and Fig. 8 show the effect of using robustly estimated similarity transformation from position of landmarks.

Figure 4: **Qualitative results on the DAVIS dataset**: Comparisons on *blackswan* and *car-roundabout* sequences, between (from top to bottom for each sequence): JUMPCUT, ROTOBRUSH, ROTO++ and ROAM.

Global colour models and reparametrization. We investigated the effects of adding reparametrization and global colour models to our framework. The numeric benefits of these elements can be seen in Tab. 2 and qualitative results on the *surfer* sequence from VIDEO SNAPCUT dataset are provided in Figs. 5 and 10. Observe that the local colour models are a powerful way to capture local appearance complexities of an object through a video sequence. However, self-occlusions and articulated motion can cause these models to fail (right arm crossing the right leg of the surfer). Our contour reparametrization allows the efficient handling of this situation. Furthermore, the beneficial effect of the global colour models can be observed in Fig. 10, where the right foot of the surfer is successfully tracked along the whole

video.

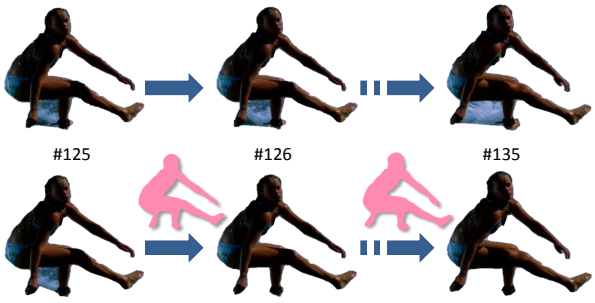

Figure 5: **Using proposals based on graph-cut**: Proposals (in pink) obtained through graph-cut minimization of an instrumental labeling energy using current colour models allows ROAM to monitor and accommodate drastic changes of object's outline (Bottom). Without this mechanism, parts of surrounding water get absorbed in surfer's region, between the leg and the moving arm (Top).

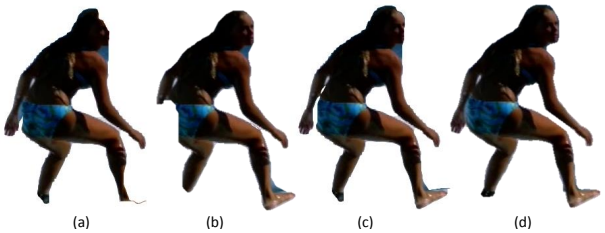

Figure 6: **Assessing first part of the model**. (a) Edge strength only; (b) Global colour model; (c) Edge strength combined with global colour model; (d) With full cost function E^C, including local colour modeling, on frame 13 from *surfer* sequence.

Qualitative results. Result samples on several sequences from DAVIS dataset in Fig. 4 demonstrate the superior robustness of ROAM compared to other approaches when rotoscoping of the first image only is provided as input (and last image as well for ROTO++). Additional results are provided in the supplementary material.

Timing breakdown. Table 4 provides detailed timing breakdown for our algorithm. These timings were obtained on an Intel Xeon 32@3.1GHz CPU machine with 8GB RAM and Nvidia GeForce Titan X GPU. Note that only part of the approach (evaluation of various potentials and dynamic programming) was run on the GPU. In particular, the reparametrization steps could also be easily run on the graphics card, yielding real-time performance.

Convergence. Fig. 9 demonstrates that the alternating optimization described in §4 converges quickly within a few iterations.

Table 4: Timing details for full configuration of ROAM.

Step	Min.	Max.	Avg.
DP Contour	0.018	0.113	0.084
DP Landmarks	0.003	0.072	0.052
Local models edge terms	0.342	0.671	0.581
Other terms	0.012	0.015	0.013
Reparametrization	0.032	7.403	2.226

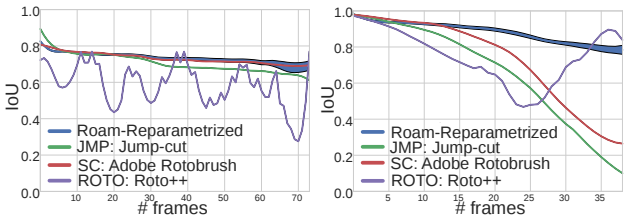

Figure 7: **Evolution of IoU for different sequences of the** DAVIS **dataset**. For our method, the blue shadow indicates influence of varying the label space size for each node (set of possible moves in dynamic programming inference).

Figure 8: **Benefit of landmarks-based modeling**. Automatically detected landmarks (orange bounding boxes) are accurately tracked on the *plane* sequence. This further improves the control of the boundary (bottom), as compared to ROAM without landmarks (top).

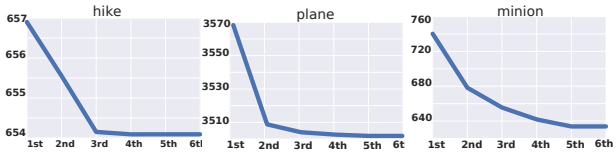

Figure 9: **Energy vs. number of iterations** on three sequences from the experimental datasets.

6. Conclusion

We have introduced ROAM, a model to capture the appearance of the object defined by a closed curve. This model is well suited to conduct rotoscoping in video shots, a difficult task of considerable importance in modern production pipelines. We have demonstrated its merit on various competitive benchmarks. Beside its use within a full rotoscoping pipeline, ROAM could also be useful for various forms of

object editing that require both accurate enough segmentation of arbitrary objects in videos and tracking through time of part correspondences, *e.g.* [15, 24]. Due to its flexibility, ROAM can be easily extended; in particular, with the recent ROTO++ and its powerful low-dimensional shape model.

References

[1] A. Agarwala, A. Hertzmann, D. H. Salesin, and S. M. Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM TOG (Proc. Siggraph)*, 2004. 2, 3

[2] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *T-PAMI*, 1990. 5

[3] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video Snapcut: Robust video object cutout using localized classifiers. *ACM TOG*, 2009. 2, 6, 7, 9

[4] C. Bibby and I. Reid. Robust real-time visual tracking using pixel-wise posteriors. In *ECCV*, 2008. 9

[5] A. Blake and M. Isard. *Active contours*. 2000. 2

[6] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in nd images. In *ICCV*, 2001. 2, 3, 5, 9

[7] B. Bratt. *Rotoscoping: Techniques and tools for the Aspiring Artist*. Taylor & Francis, 2011. 1

[8] P. Dollar and L. Zitnick. Sketch tokens: A learned mid-level representation for contour and object detection. In *CVPR*, 2013. 4

[9] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 6, 7

[10] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. JumpCut: Non-successive mask transfer and interpolation for video cutout. *ACM TOG*, 2015. 3, 6, 7

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *T-PAMI*, 2010. 4, 5

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 4

[13] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Kernelized correlation filters. *T-PAMI*, 2015. 5, 6, 7

[14] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1988. 2, 3

[15] E. A. Khan, E. Reinhard, R. W. Fleming, and H. H. Bülthoff. Image-based material editing. *ACM TOG*, 2006. 9

[16] P. Kohli and P. H. Torr. Dynamic graph cuts for efficient inference in Markov random fields. *T-PAMI*, 2007. 5

[17] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *T-PAMI*, 2008. 10

[18] W. Li, F. Viola, J. Starck, G. J. Brostow, and N. Campbell. Roto++: Accelerating professional rotoscoping using shape manifolds. *ACM TOG (Proc. Siggraph)*, 2016. 1, 2, 3, 6, 7

[19] Y. Lu, X. Bai, L. Shapiro, and J. Wang. Coherent parametric contours for interactive video object segmentation. In *CVPR*, 2016. 2, 6, 7

[20] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016. 6, 7

[21] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 2004. 5

[22] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 6, 7

[23] J.-M. Pérez-Rúa, T. Crivelli, and P. Pérez. Object-guided motion estimation. *CVIU*, 2016. 7

[24] A. Rav-Acha, P. Kohli, C. Rother, and A. Fitzgibbon. Unwrap mosaics: A new representation for video editing. *ACM TOG*, 2008. 9

[25] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM TOG (Proc. Siggraph)*, 2004. 3, 6, 7, 10

[26] C. Stauffer and E. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999. 5

[27] G. Y. Tang. A discrete version of Green's theorem. *T-PAMI*, 1982. 4

[28] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. 6, 7

[29] D. J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *CVIU*, 1992. 5

[30] S. Wright. *Digital compositing for film and video*. Taylor & Francis, 2006. 1