Contour tracking for rotoscoping based on trajectories of feature points

V. Garcia*, S. Boltz, É. Debreuve, and M. Barlaud

Laboratoire I3S, Université de Nice - Sophia Antipolis, CNRS
2000 route des Lucioles, 06903 Sophia Antipolis, FRANCE
{garciav,boltz,debreuve,barlaud}@i3s.unice.fr

Abstract. Rotoscoping is the generic term for methods that consist in defining the contour of a moving object in the frames of a video in order to apply a local processing. It is usually performed manually and frame by frame in the cinema industry. Semi-automatic algorithms have been proposed to reduce the load of this task. However they classically use contour-based information and consequently lack robustness in the presence of occlusions. We propose a rotoscoping method based on tracking in both the forward and backward directions. Each tracking relies on a motion estimation performed in group of pictures. Motion is estimated from temporal trajectories of region-based feature points by minimizing the entropy of the residual. The use of trajectories brings robustness to time-variant statistics due to occlusions and the use of entropy brings robustness to outliers. The proposed method seems accurate and allows to cope with large object occlusions.

* This work was partly supported by *"Le Conseil Régional Provence-Alpes-Côte d'Azur"*, France.

1 Introduction

Originally rotoscoping is a technique where animators trace live action movement, frame by frame, for use in animated cartoons. The term *rotoscoping* is now generally used for the corresponding all-digital process of tracing outlines over digital film images to produce digital contours (also called *mattes*) in order to allow special visual effects. This technique is still widely used for special cases where techniques such as bluescreen will not pull a matte accurate enough. However, the time allocated to rotoscoping increases due to special effect success and therefore semi-automatic methods become necessary.

Given the object contour represented by a 2D closed curve (*e.g.* polygon, spline) edited by the animator in the first and the last frame of a video, the semi-automatic rotoscoping problem consists in computing the object contour in the intermediate frames. The process has to be precise and robust to occlusions to provide an accurate and natural matte. The curves are defined by a set of sample points and their position on the object should be the same over time in order to allow the animator interaction in the intermediate frames.

Some tracking methods proposed in the literature [1, 2] perform object detection in the form of a bounding box. These methods are more adapted to scene analysis and understanding. Methods using global (*i.e.* region) information [3, 4] are usually based on a notion of (possibly non-trivial) homogeneity of the object (*e.g.*, intensity, motion, histogram...). If the object is complex or has a complex motion, this homogeneity description might be difficult to establish and not precise enough to guarantee an accurate tracking. The proposed approach is based on the tracking of so-called feature points of the object, which can be seen as an extension to a sequence of a registration problem between two images.

In this paper, we divide the rotoscoping problem into two tracking problems (one forward from the initial contour and one backward from the final contour) and a merging problem. The proposed approach relies on a trajectory-based contour tracking method in a group of pictures (GOP). In addition, we propose to use a robust parameter estimation based on a minimum-entropy estimation in order to increase robustness to outliers for motion estimation. The proposed method is accurate and robust to large occlusions commonly present in real sequences.

The paper is organized as follows: Section 2 focuses on the proposed contour tracking method. Section 3 describes the proposed rotoscoping method. Section 4 shows and discusses some tracking and rotoscoping examples by using a measure of quality and finally Section 5 concludes.

2 Proposed contour tracking method

In this section, we focus on the problem of contour tracking. Let v be a video of n frames $F_1, \cdots, F_n$. Let c_1 be the initial, hand-edited contour on frame F_1 segmenting an object. The contour is a set of sampling points interpolated by a curve such as a spline. The contour tracking consists in computing the contours c_i, $i \in [2, n]$, by deforming the contour c_1 over time.

2.1 Region-based tracks

The proposed tracking method is based on *tracks*. A *track* is a 1D trajectory along the time dimension of a feature point. A feature point is a point of a frame having particular characteristics (corner, junction, etc.). A track is a set of at least two feature points which all belong to different frames. It can start after the first frame, end before the last frame and its *lifetime* is independent of the other tracks.

Tracks are built as follows. First, feature points are extracted in each frame using low-level algorithms [5]. Second, a descriptor (the description of the feature point neighborhood, typically a block of luminance values centered on the point) is associated with each feature point. We make the assumption that a feature point keeps the same descriptor over time. The value of the L^2 norm between two descriptors is used to distinguish the corresponding feature points. Third, feature points are linked pairwise as follows. Let F_m be the m^{th} frame of the sequence, p_m^b be the b^{th} feature point defined in F_m and d_m^b its associated descriptor. We search for a feature point p_l^a defined in a previous frame F_l which minimizes $\| d_m^b - d_i^j \|_{L^2}$, $i < m$. Then we search for a feature point p_n^c defined in a subsequent frame F_n which minimizes $\| d_l^a - d_i^j \|_{L^2}, i > l$. If p_m^b and p_n^c are identical (*i.e.* $m = n$ and $b = c$), the link $\{p_l^a, p_m^b\}$ is created. This method is used for all the feature points in all the frames providing a set of links. Finally, feature point pairs are concatenated in order to build a set of tracks.

Then, the tracks are used to build a set of matching points as follows. Given a track $T = \{p_1, p_2, p_3, p_5, p_6\}$ defined on frames F_1, F_2, F_3, F_5 and F_6, we extract from T the set of matching points defined on consecutive frames: $S = \{\{p_1, p_2\}$, $\{p_2, p_3\}$, $\{p_5, p_6\}\}$.

2.2 Motion estimation in GOPs

Classical tracking methods compute the contour c_{i+1} in frame F_{i+1} from contour c_i in the previous frame. Instead, we suggest to compute simultaneously the contours $c_i, c_{i+1} \ldots c_{i+m-1}$ of GOPs composed of m consecutive frames F_i, $F_{i+1} \ldots F_{i+m-1}$. It is reasonable to assume that the object motion is stationary within a GOP (or conversely, the GOP size m is chosen such that this assumption is reasonable). For a given GOP, let us suppose that the contour in the first frame, called the reference contour, is known. It is obviously true for the first GOP: the reference contour has been hand-edited in its first frame. Only the tracks intersecting the interior domain of the reference contour will be considered for the current GOP. Matching points are extracted from these tracks. The motion M is estimated from the set of matching points (see Section 2.3) and applied j times to the sampling points of the reference contour to compute the contour c_{i+j} of frame F_{i+j}. The contour c_{i+m-1} of the last frame of the GOP will be used as the reference contour of the next GOP. The tracks are used to select successive pairs of matching points, allowing to perform this GOP processing. Without the tracks coherence, the pairs of matching points would be independent, isolated and, as such, might correspond to feature points occluded during most of the GOP or not linked to the interior domain of the reference contour. Note that there is a trade-off between the necessity to keep the GOP size small enough to respect the motion stationary condition and the need to have enough matching points to allow a robust motion estimation. The global scheme of the proposed method has been presented. We will now discuss the motion model and the method to estimate it.

2.3 Motion estimation

Given $S = \{\{p_1, p'_1\}, \{p_2, p'_2\}, ..., \{p_k, p'_k\}\}$, a set of k pairs of matching points, the motion estimation consists in computing the model M which transforms p_i into p'_i: $p'_i = M.p_i,\ i \in [1, k]$. Let $\widehat{M}$ be the estimation of M computed by minimizing the residual r given by

$$r_i(M) = p'_i - M.p_i,\ \forall i \in [1, k].$$

Note that p_i and p'_i are given in homogeneous coordinates. Let $\widehat{p'}_i$ be the estimation of p'_i given by $\widehat{p'}_i = \widehat{M}.p_i,\ i \in [1, k]$. The misestimation of M produces an error on p'_i given by

$$\begin{aligned} p'_i - \widehat{p'}_i &= (M - \widehat{M}).p_i \\ &= \begin{pmatrix} \alpha\ \beta\ \chi \\ \delta\ \epsilon\ \phi \\ 0\ 0\ 1 \end{pmatrix} . \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} \alpha.x_i + \beta.y_i + \chi \\ \delta.x_i + \epsilon.y_i + \phi \\ 1 \end{pmatrix} . \end{aligned} \tag{1}$$

According to (1), errors on the estimation of the rotation and the scaling parameters (*i.e.* α, β, δ and ϵ) imply an error on p'_i depending linearly on the coordinates of p_i. On the opposite, errors on the translation parameters (*i.e.* χ and ϕ) imply an error independent of p_i. Therefore, a translation model will be less sensitive to noise in the observations than rotation or scaling. The following model was used:

$$M = \begin{pmatrix} S_x & 0 & T_x \\ 0 & S_y & T_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

It represents a translation and a non-uniform scaling. It is a trade-off between fidelity to the data (which requires more parameters) and well-posedness.

Robust parameter estimation methods use minimization criteria based on the absolute value or on M-estimators [6, 7]. However, efficiency of these estimators drops when the distribution of the data is neither gaussian nor laplacian. Instead, we suggest to use a nonparametric approach [8, 9] in 2 steps: first, the distribution of the data is estimated using a kernel-based approach and second, the entropy of the distribution is estimated. Entropy is a convex function of the density

of the observations that coincides locally asymptotically with likelihood at its optimum. This suggests that an estimator minimizing the entropy of the residual should be efficient and robust to outliers.
Since entropy is invariant by translation, if M minimizes the residual then $M+t$, where t is a translation, minimizes the residual. To find a unique minimizer, the residual needs to be symmetrized. The entropy is computed using the Ahmad entropy estimator [10] as follows:

$$H_k(r(M)) = -\frac{1}{k}\sum_{i=1}^{k} \log[\frac{1}{2}(\hat{f}(r_i(M)) + \underbrace{\hat{f}(-r_i(M))}_{\text{symmetrization}})] \quad (3)$$

where $\hat{f}$ denotes the Parzen kernel density estimate of the residual r. The kernel bandwidth σ was set empirically. Finally, the motion $\widehat{M}$ in a GOP is computed by minimizing the criterion (3) with respect to M, *i.e.*,

$$\widehat{M} = \arg\min_{M} H_k(r(M)). \quad (4)$$

The minimization of (4) was performed using a the simplex method.

3 Rotoscoping application

In this section we propose a rotoscoping method based on the contour tracking method of the Section 2. Let *keyframe* be a frame where the contour was hand-edited by the animator and let *interframes* be the other frames. First, we present the proposed method with two keyframes. Second, we explain how user can refine the process by adding new keyframes.

3.1 Two keyframes

We divide the rotoscoping problem for a video of n images into a two directional contour tracking problem and a merging problem. The contour tracking method can be applied in the forward direction as seen before, but it can also be applied in the backward direction because the track building is independent of the temporal direction. The suggested method is as follows:

- compute the forward contours c_i^f $i{\in}[2,n]$ by using contour tracking from initial contour $c_1^f = c_1$ in the forward direction,
- compute the backward contours c_i^b $i{\in}[1,n-1]$ by using contour tracking from final contour $c_n^b = c_n$ in the backward direction,
- merge forward and backward contours by computing weighted average contours C_i as follows:

$$C_i = \frac{n-i}{n-1}.c_i^f + \frac{i-1}{n-1}.c_i^b,\ i \in [1,n]$$

Note that the contours C_1 and C_n on keyframes are equal to the hand-edited contours c_1 and c_n.

3.2 Refining by adding keyframes

The motion of the object may be quite complex and rotoscoping needs sometimes more than two keyframes to compute contours precisely enough. The *refinement* is the animator interaction on the contour C_m of an interframe F_m, $2 \leq m \leq n-1$, where C_m is too far from the actual object contour. The animator modifies C_m by pulling some sample points. F_m becomes a new keyframe and rotoscoping is performed again separately on intervals $[1, m]$ and $[m, n]$. Refinement is done until obtaining a natural matte of the object.

4 Experiments

In this section we propose an objective evaluation of the proposed rotoscoping method. The proposed method has been tested on several sequences. Four example are presented here. Note that refinement is not used in these experiments in order to evaluate the method accuracy and not the animator skill.

4.1 Measure of quality of a contour

Given a frame F_i, let A_i be the mask manually defined by an expert and $\widehat{A}_i$ be the mask defined by the estimated contour C_i. The masks are equal to 1 inside the contour and 0 outside. In [11], the segmentation error is measured by the following term

$$d_i = 100 \times \frac{\sum_{x,y} A_i(x,y) \oplus \widehat{A}_i(x,y)}{\sum_{x,y} A_i(x,y)}$$

where $\oplus$ is the xor operation. For rotoscoping, this error is averaged for all the interframes:

$$\overline{d} = \frac{1}{n-2} \sum_{i=2}^{n-1} d_i$$

4.2 Evaluation of proposed method

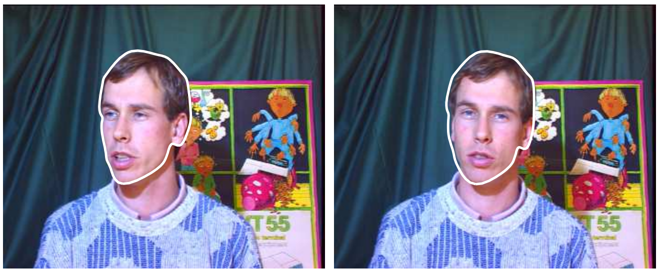
Rotoscoping allows to constrain the contour simultaneously in forward and backward direction by computing an average contour. Therefore rotoscoping is more precise than a pure tracking, up to four times better for the sequence *Bus* seen on Fig. 3 (4% of misclassified pixels for rotocoping and 16% for forward tracking). Therefore, the method will be evaluated only for the forward tracking.

The use of GOPs allows to increase the number of observations, providing a better context for robust motion estimation. However, the size of GOPs must not compromise the assumption of motion stationarity. Tab. 1 illustrates this phenomenon on sequence *Erik* (see on Fig. 1) and shows the usefulness and limit of GOPs. First, it can be noted that the number of observations increases considerably with the size of GOP. Second, the quality of the contour is optimal for a GOP of nine frames: with a smaller GOP, the estimation is less robust, and with a larger GOP, the stationarity of the motion is not verified. In other words, there is a trade-off between a good context for robust motion estimation and the assumption of motion stationarity within the GOPs. Fig. 1 shows that the contour sticks to the face over time. The segmentation error is less than 2.5% (in misclassified pixels) for GOPs of 9 frames.

	GOP 1	GOP 5	GOP 9	GOP 15
$\overline{d}$	19.7 %	18.5 %	**18.1** %	22.5 %
Observations	12.6	54.5	79.7	107

Table 1. Evolution of the contour quality (given in percentage of misclassified pixels) and the number of observations (given in tracks per GOP) with respect to the size of GOP for the forward tracking on sequence *Eric*.

The entropy-based criterion brings robustness to the motion estimation but its usefulness depends on the motion model used. Let M^T be the translation model and M^{T+S} be the translation + non-uniform scaling model (see Eq. (2)). In spite of its robustness (studied in Section 2.3), the model M^T is too simple to allow the tracking of an object with a motion more complicated than a "pure" translation as we can see on the standard sequence *Tempete* (see Fig. 2). The model M^{T+S} is more precise. As seen on Tab. 2, the use of entropy brings robustness to the tracking with M^{T+S} and increases the quality of the contour.

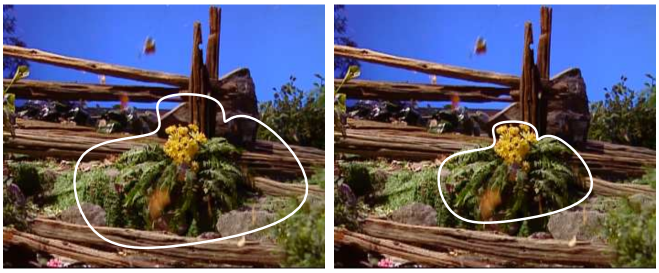


(a) Frame F_1

(b) Frame F_7

Fig. 1. Rotoscoping results on sequence *Erik*.

The sequence *Bus* (see Fig. 3) gives an example of occlusion. In this sequence, a bus on a highway is largely occluded by a billboard. The motion of the bus is a translation and a scaling. The occlusion causes a variation of the object statistics over time. The joint use of GOPs and entropy-based criterion allows to manage occlusions as seen in Fig. 3: the tracks following the billboard motion are considered as outliers. The segmentation error is less than 4%.



(a) Model M^T (b) Model M^{T+S}

Fig. 2. Tracking results on sequence *Tempete* using respectively the models M^T and M^{T+S} on frame F_{100}.

	GOP 2	GOP 6	GOP 8	GOP 10	GOP 14
SSD	65.8 %	52.9 %	48.9 %	66.3 %	22.5 %
Entropy-based	**56.8** %	**43.1** %	**14.7** %	**14.9** %	**16.9** %

Table 2. Evolution of the contour quality (given in percentage of misclassified pixels) with respect to the GOP size for the forward tracking on sequence *Bus*.

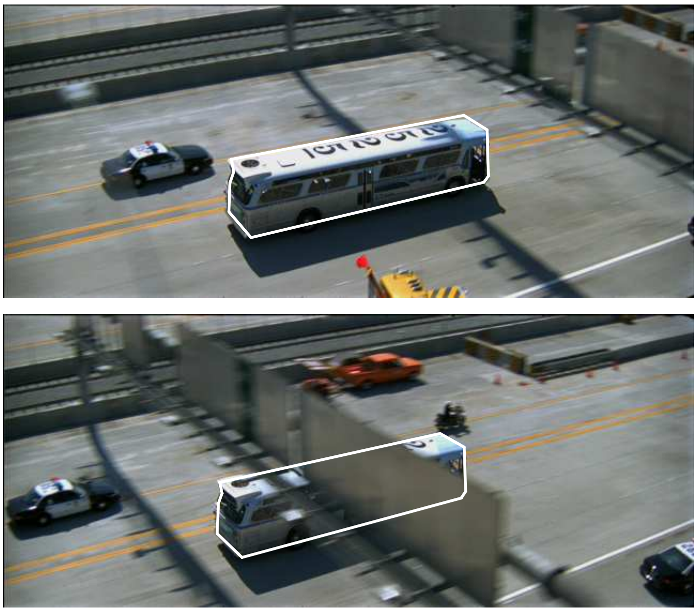


Fig. 3. Rotoscoping result on sequence *Bus* on frames F_{17} and F_{28}.

5 Conclusion and perspectives

We have proposed a rotoscoping method based on a new contour tracking. This tracking relies on a robust motion estimation performed in groups of pictures using trajectories using region-based information. The robustness to outliers is both due to the use of GOP and the use of entropy as a measure of error of the estimated motion. The rotoscoping processing is the result of the tracking in both the forward and backward directions. According to the experiments, the proposed rotoscoping method seems accurate and robust to large object occlusions.

The suggested model does not allow deformations more complex than affine deformations from frame to frame. In the future, we plan to define a local approach allowing natural deformations.

References

1. Zhang, S.: Object tracking in unmanned aerial vehicle (uav) videos using a combined approach. In: Proc. Intern. Conf. on Acoustics, Speech, and Signal Processing. (2005)
2. Venkatesh Babu, R., Pérez, P., Bouthemy, P.: Robust tracking with motion estimation and kenel-based color modelling. In: Proc. Int. Conf. on Image Processing (ICIP'05), Genova, Italy (2005)
3. Blake, A., M.Isard: Active Contours: The Application of Techniques from Graphics,Vision,Control Theory and Statistics to Visual Tracking of Shapes in Motion. Springer–Verlag, New-York, tats-Unis (1998)
4. Debreuve, É., Gastaud, M., Barlaud, M., Aubert, G.: A region-based joint motion computation and segmentation on a set of frames. (In: Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS))
5. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. of The Fourth Alvey Vision Conference, Manchester, UK (1988) 147–151
6. Black, M.J., Anandan, P.: The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. Comput. Vis. Image Underst. **63** (1996) 75–104
7. Charbonnier, P., Blanc-Féraud, L., Aubert, G., Barlaud, M.: Deterministic edge-preserving regularization in computed imaging. IEEE Trans. Image Processing **6** (1997) 298–311
8. Wolsztynski, E., Thierry, E., Pronzato, L.: Minimum entropy estimation in semi parametric models. Signal Processing **85** (2005) 937–949
9. Wang, H., Suter, D.: Mdpe: A very robust estimator for model fitting and range image segmentation. International Journal of Computer Vision **59** (2004) 139–166
10. Ahmad, I., Lin, P.: A nonparametric estimation of the entropy for absolutely continuous distributions. In: IEEE Trans. Inform. Theory. (1989)
11. Mech, R.: Description of COST 211 analysis model. COST 211^{quat} simulation group, Dublin. (1998)