Object Flow

A per-object dense motion descriptor

Juan Manuel Pérez Rúa

Technicolor SA

A Thesis Submitted for the Degree of
MSc Erasmus Mundus in Vision and Robotics (VIBOT)

· 2014 ·

Abstract

Motion analysis in image sequences has undoubtedly shown good progress in terms of its two main research branches. Optical flow estimation and object visual tracking have been mostly studied as isolated problems, and high accuracy algorithms are available when needed as independent bricks. This paper presents a framework for combining object tracking techniques with optical flow methods aiming towards a precise motion description for objects in video sequences. Firstly, we introduce a method to extend max-flow min-cut based segmentation techniques to videos, without adding the computational load of performing a graph cut optimization approach for 3-dimensional (volume images) graphs. This is done by exploiting the inherent foreground-background separation hints given by object trackers, and the novel concept of superpixel flow, which is used to perform background regions tracking. Then, it is shown that long-motion awareness obtained from object tracking, together with a per frame object segmentation can improve the precision of the object motion description in comparison to several optical flow techniques. The proposed approach may be called Object flow as it offers a dense and semantic aware description of the current aparent motion state of the studied object.

Todo arde si le aplicas la chispa adecuada....

Enrique Bunbury

Contents

List of Figures

Chapter 1

Introduction

Object tracking and optical flow are two of the main components in the computer vision toolbox, and have been focus of great research efforts, leading to significant progress in the last years [16] [17]. The object tracking problem in videos consists on estimating the position of a target in every frame, given an initial position. On the other hand, the optical flow between a pair of frames consists on finding a displacement vector for each pixel of the first image, namely a *dense motion or displacement field*. Even though for several applications a complete (i.e. for every pixel) motion-field is needed, other applications like human-computer interaction, object editing in video or structure-from-motion, may only focus on an interest object and thus, only a subset of motion vectors is required. In such scenarios combining optical flow and object tracking in a unified framework appears useful and the precision of the object motion description could be enhanced. For instance, even with modern optical flow approaches, the long term dense motion estimation remains a challenge [20] [22]. At large, object trackers provide a more robust, longer term motion estimation featuring a global description of an object, specially after recent works based on tracking-by-detection approaches [16] [23] [24]. On the other side, they lack the (sub) pixel precision of dense optical flow estimators, as well as a deeper use of contextual information for bundle motion vector estimation. Even more, object trackers and optical flow give precious hints for other fundamental tools such as object segmentation in video. Nevertheless, these two techniques were not deeply studied in the literature as a unified problem. Though optical flow has been widely used as a motion feature for object tracking [25], feeding a dense motion estimator with tracking information is not being fully exploited. This being said, we introduce a new problem which we call object flow. Thus, for a given object of interest, the object flow is the set of displacement vectors for every pixel that belong to the target in a first frame, towards another frame of the sequence. In other words, a dense displacement field constrained to the spatial support of the object. Note that by definition this induces a segmentation of the target

and of the motion field.

1.1 Problem definition

We can define more precisely the object flow by starting with an image sequence, say $I_t, t : 0..N-1$, and an initial position of the interest object in the first frame of this sequence. Let $\mathcal{R} \in \Omega$ be the region corresponding to the support of the object in the bi-dimensional grid Ω. Then, the object flow is $\mathcal{O}(x) = d_{0,t}(x), \forall x \in \mathcal{R}$. We can define more precisely the object flow by starting with an image sequence (See Fig. 1.1 for a simple diagram) and an initial position of the interest object in the first frame of this sequence, and letting $\mathcal{R}$ be the region corresponding to the support of the object in 2D, such that $\mathcal{R} \subset \Omega$. If Ω is the set of all the possible grid positions, the object flow problem consist in finding the displacement vectors $d_{0,t}(x)$ from the image I_0 to I_t, $\forall x \in \mathcal{R}$.

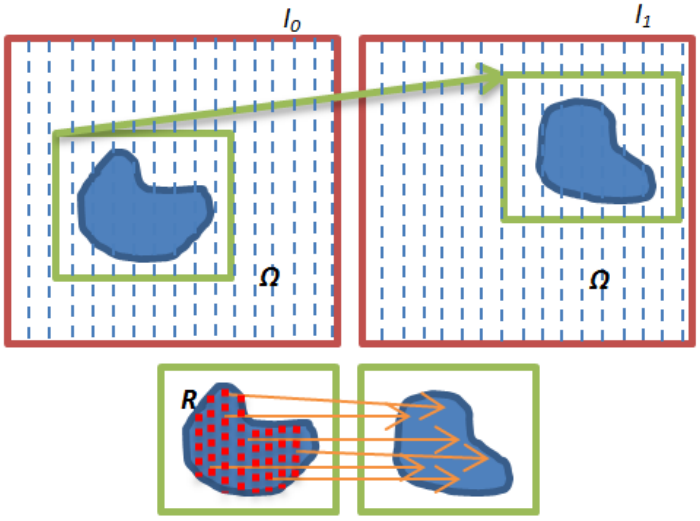

Figure 1.1: Object flow definition diagram.

A straightforward solution to this problem would be to compute the optical flow motion field, and apply a segmentation mask to recover the desired motion vectors. Nevertheless, this approach carries several problems. For example, a globally computed optical flow method can affect small objects motion, because of the common use of heavy regularization priors. Moreover, even if the segmentation mask is extracted from a tracker window by a graph-cut based segmentation method, is likely that this mask is not going to be well suited for the interest object and some extra user interaction would be needed to refine this process. We propose an

approach to reduce these problems.

1.2 Objectives

The main goal of this work is to define a dense motion descriptor for objects in video sequences. In order to achieve this goal, some specific objectives are defined:

- To show experimentally that the idea of mixing tracking techniques with optical flow methods enhance the precision of per-object motion description.
- To define an automatic segmentation approach for target objects in video sequences.
- To establish a flow estimation method that uses the segmentation mask provided and the region of interest given by the tracker.
- To show experimentally that the proposed object flow method is indeed more precise than regular optical flow techniques.

1.3 Document organization

The section 2 starts by introducing the reader in the important topics and concepts that support the rest of this work: object tracking, optical flow and object segmentation. After this, in the section 3, the object flow pipeline is explained in deep, together with the proposed algorithms. Finally, a list of results and implementation details are discussed in 4.

Chapter 2

Background and Related Work

2.1 Introduction

The background of the object flow concept is mainly related to object tracker techniques and optical flow methods. A short state of the art review is presented in following sections, without going too deeply in any specific approach, since those are widely diverse. Of course, another important point is the object based segmentation in videos problem. Several works have involved this or similar problem in the state of the art, and only the more related approaches are presented. On the other hand, as the object flow itself is a novel method, the related work is not large. However, as, in terms of applications, some works have done towards specific oncomings to this concept. Some of these works are superficially explained.

2.2 Object Tracking

As one of the most studied computer vision problems, object tracking has been constantly evolving since its early approaches. As the techniques were getting better, the benchmarking has been evolving too. The last global work in online object tracking was the work of Wu et al. [16] in 2013. Several remarks can be extracted from the deep benchmarking analysis of this work. For instance, it seems to be more clear that background information is a key hint towards better tracking methods. Some kind of modelling of the background should be an important step in the tracking pipeline. Whether this background modelling is implicit like in [23] or explicit like in [22], where is used as context, is just matter of design. Of course, this background modelling have to be accompanied with local model to account with variations (occlusions, deformations) in the interest object. It seems that these two are the reasons why

Figure 2.1: Sequences used to evaluate object trackers in [16].

tracking by detection and learning approaches are among the most successful ones. Usually, these methods account with local and background modelling ([22] [23] [24] [25] [26]). However, even when motion or dynamical models is crucial for object tracking, very few of the last state of the art techniques focus on this element. The Fig. 2.2 shows results for the top 15 state of the art trackers as presented in [16], taking into account the variability of different initialization. According to this data, the tracking problem is still open and further exploration can be done. Moreover, it has to be observed that tracking by detection methods are indeed in the top of results in stability in initialization and peak performance. The Struck tracker [23], for instance, seems to hold the higher position w.r.t stability. Another curious observation is the fact that even traditionally good color based particle filter approaches are left behind with last years object trackers.

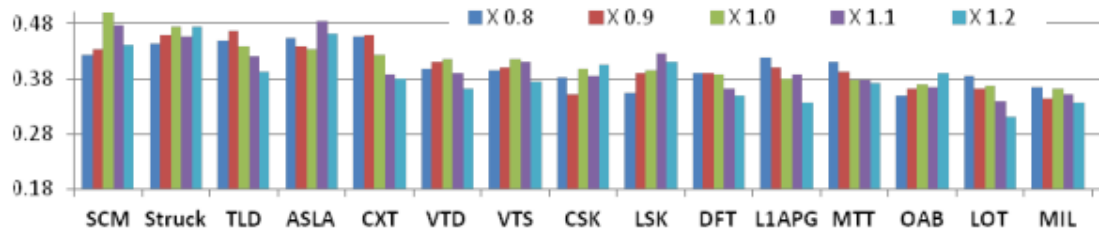

Figure 2.2: Performance summary for the top 15 trackers benchmarked in [16], initialized with different size of bounding box.

2.3 Optical Flow

Figure 2.3: Frame pairs used to evaluate optical flow methods in [17], Ground truth flow coded with the proposed flow coding.

Optical flow is another traditional problem in computer vision, and it may be argued that its under-determined nature makes it one of the most difficult ones. Several well known databases and benchmarks have been proposed to account for the different intrinsic complications of the problem. The last and larger benchmarking work for this problem may be Baker et al. [17], but other useful studies are available [27]. These works reveal that most of the existing methods establish the problem as the optimization of an energy function that takes into account two terms. The first one, E_{data} measures data consistency over input frames, and E_{prior} which favours flow with certain characteristics, e.g. smoothness in the flow. It seems that one of the major challenges remains to be the best pair of objective function and optimization method. A number of combinations have been proposed [29] [30] [32]. However, totally different approaches had been also proposed, e.g. by using polynomial expansions [28].

$$E_{global} = E_{data} + \lambda E_{prior} \tag{2.1}$$

According to the update of 2011 of the work in [17], the first 15 optical flow methods by interpolation error are shown in the Fig. 2.4. It has to be noted that the performance in several of the dataset is already very good for the most of these methods.

Method	Avg	Avg IE by dataset							
		Mequ.	Scheffl.	Urban	Teddy	Backyd.	Basktb.	Dumptr.	Evergr.
CBF	3.5	2.3	5.3	1.3	3.3	4.0	3.0	3.7	5.3
Aniso. Huber-L1	4.6	4.0	11.3	2.3	4.0	8.3	1.7	1.0	4.0
Second-order prior	5.5	3.3	8.0	6.0	3.0	6.3	4.3	3.0	9.7
Brox et al.	6.3	5.7	3.0	4.7	3.3	2.3	14.0	16.3	1.0
F-TV-L1	7.1	14.7	11.0	5.0	7.7	4.0	2.7	5.7	6.0
Filter Flow	9.7	10.7	16.0	9.0	9.3	5.3	9.7	7.0	10.3
Fusion	10.0	4.7	2.0	6.3	6.7	13.3	21.3	10.0	16.0
Black & Anandan	10.1	12.7	17.7	15.7	12.3	4.0	7.7	7.7	3.0
DPOF	10.2	15.0	1.0	15.3	6.7	13.7	9.0	8.7	12.0
2D-CLG	11.0	8.0	15.7	9.7	12.3	17.3	6.7	13.3	5.0
Horn & Schunck	11.1	9.0	20.0	13.7	16.3	4.7	5.3	13.0	7.0
Adaptive	12.5	11.7	16.7	7.0	12.0	14.3	14.3	12.0	12.0
Complementary OF	12.5	13.3	4.3	19.0	13.0	14.7	9.0	11.0	15.3
TV-L1-improved	12.8	8.3	15.3	11.0	5.0	11.7	18.3	18.3	14.3
Graph Cuts	13.0	17.0	5.3	14.0	12.0	15.7	10.3	15.7	13.7

Figure 2.4: Performance summary for the top 15 opt. flow methods benchmarked in [17] by interpolation error.

2.4 Object segmentation in video

Among the state of the art segmentation methods for objects in video sequences, point trajectories based ones stand for its performance and reliability [33], even when only sparse trajectories are known because of computational reasons [34]. In the other hand, for the problem of extracting out a preselected object in still frames, max-flow min-cut based approaches have demostrated to be a powerful tool [14] [18].

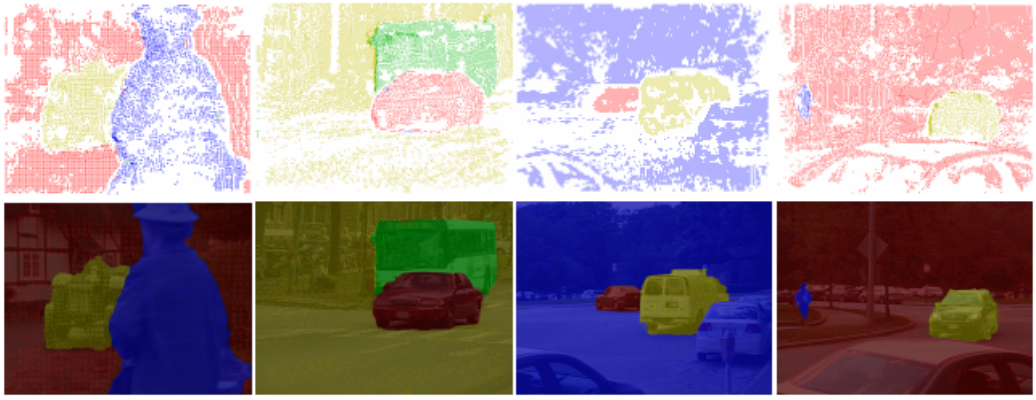

Figure 2.5: Sparse motion trajectories for segmentation. Results in several datasets [34].

The Fig. 2.5 shows good accuracy results when using sparse motion trajectories [34]. However, to overcome the sparsity of the flow the authors proposed a variational method to obtain density. The information propagation is done by presenting the problem as a non-linear diffusion process that takes superpixels into account. It is expected, then, that this method can be precise, but slow. In this work we propose to mix single frame graph based approaches with the idea of using background point trajectories. However, the sparsity of point trajectories is overcame by using superpixels. Therefore, instead of tracking points, background regions are tracked via the novel concept of superpixel flow. We show in future sections how this extra information can be used to complement the graph-cuts based techniques for an efficient foreground-background segmentation.

2.5 Related work

As previously mentioned, the related work is commented mostly from an application-wise point-of-view, due to the fact that the object flow concept is novel. The most used approaches in the literature to mix motion estimation with scene semantics are related to the use of this motion estimation to perform segmentation of scene [33] [34]. These works, however, depend intrinsically in the accuracy of the motion estimation to perform a good segmentation. Other authors have proposed to use simple low-order parametric motion models to model background movement, and by incorporating radial maps, obtain a frame-by-frame moving object segmentation [36], in contrast with the methods that combine motion awareness with appearance information [35].

In the other hand, other authors had used the optical flow constraints to track objects, which is, up to some extent, the inverse problem proposed for the object flow [37]. Some authors went further in this sense, to create specialized trackers for deformable objects by combining model based tracking with the optical flow constraint. Even when a work like the last one is very interesting for video editing tasks, the approach is limited by the mesh model availability. In this sense, the object flow proposal seems to be more generic, and thus, more adequate for this kind of applications. From other point of view, the use of superpixels seems to provide a reliable hint for large displacement optical flow. The work presented in [39] provide an interesting method to combine the idea of optical flow with superpixels, to obtain an optical flow method which can perform well in difficult datasets [27]. The full set of these ideas can lead to some conclusions towards the object flow pipeline proposal. For instance, authors have already exploited the optical flow to improve tracking methods. It remains unexplored to complete the cycle by improving the optical flow with the state of the tracker. Moreover, the use of superpixels can be combined with motion estimation to obtain both an object based segmentation and the enhancement of the motion itself.

2.6 Simple Flow

As the Simple Flow [21] is the optical flow technique that is used as base of the proposed object flow method, it is presented in more detail. The main characteristic of this method is its efficient approach, which tries to concentrate in the zones where there is evidence of motion, and use linear interpolation for excluded zones. Moreover, the problem is not solved in the usual way by minimizing a variation of (2.1), but using a simpler likelihood model that follows the constant-color assumption, without including explicitly the pairwise terms that account for smoothing. The smoothness prior it is taken into account, however, by implementation of local filters that uses pixel weighting to take into account pixel proximity (w_d) and color similarity (w_c)., leading to a equation of the type:

$$E(x_0, y_0, u, v) = \sum_{(x,y)\in\mathcal{N}_0} w_d w_c ||I_0(x, y) - I_1(x + u, y + v)||^2 \tag{2.2}$$

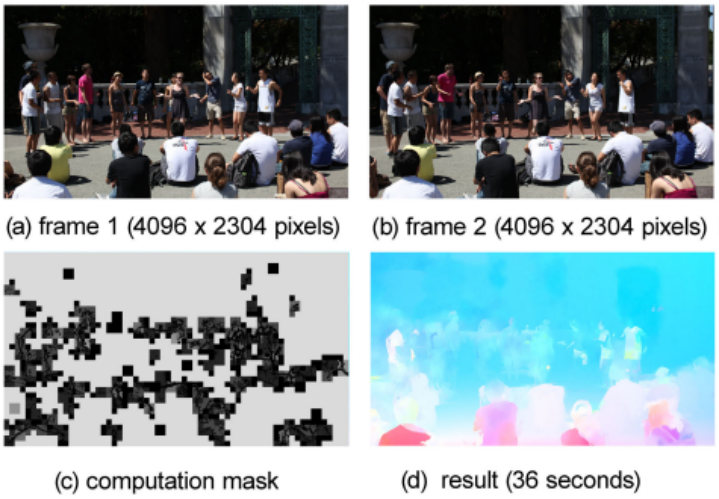

Figure 2.6: Results of the Simple Flow method in several datasets.

The practical implementation of this equation requires a cross-bilateral filter to compute E, as it takes into account pixel (x_0, y_0) centred nxn windows between I_0 and I_1, producing n^2-dimensional vector for each pixel. The flow is the vector (u_0, v_0) that minimized E, producing an integer value. The precision can be enhanced by fitting parabolas at (u_0, v_0) and extracting the minimum. A final bilateral filtering is applied to the flow field, discarding occluded pixels (determined by cross-matching using the computed forward and backward flows). To recover large motions, instead of using large nxn windows, a multi scale approach is followed. If more

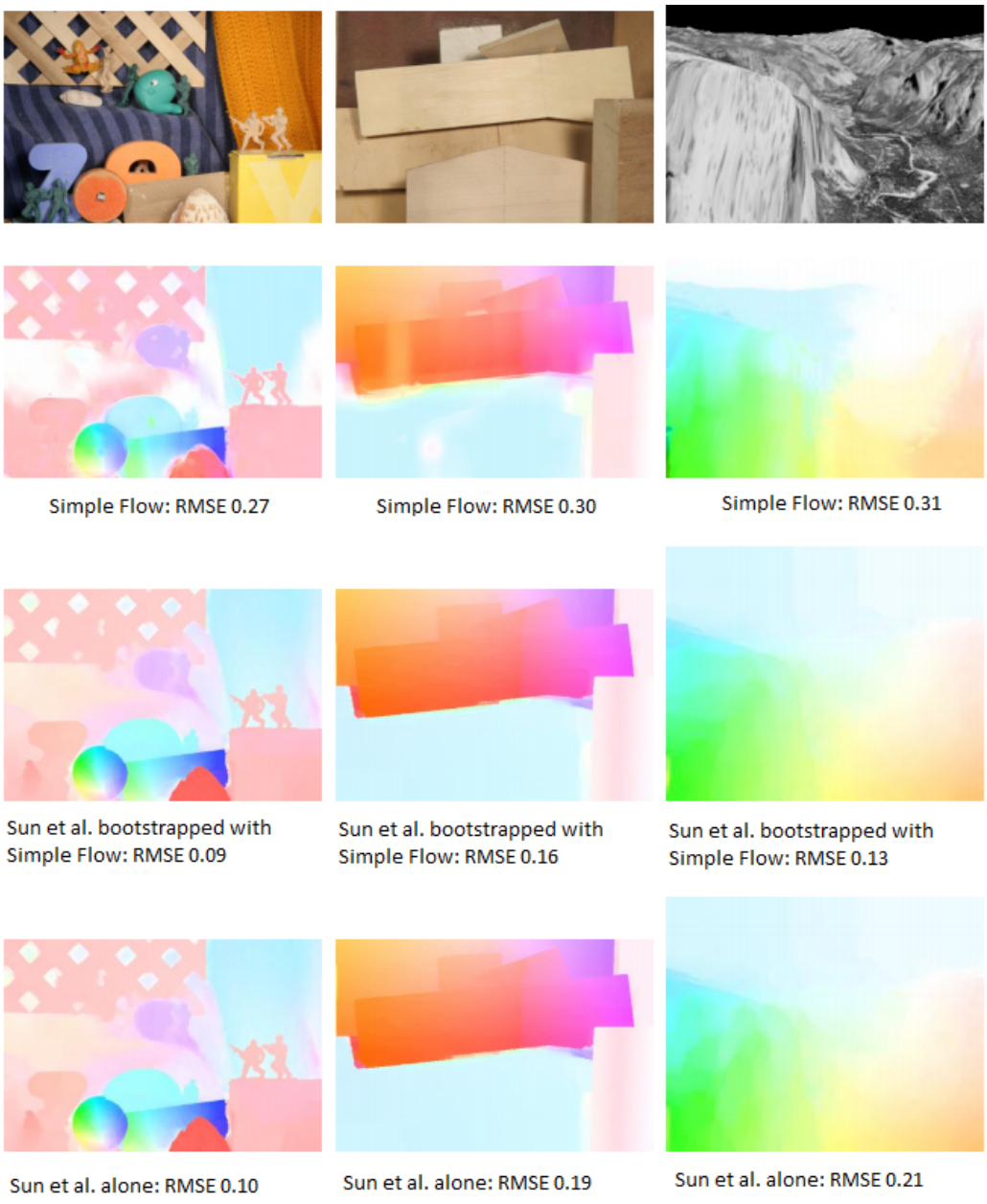

Figure 2.7: Results of the Simple Flow method in several datasets.

accuracy is desired, a more global optimization can follow, for instance, connecting the global approach used in Sun et al. [40]. This simple approach for computing optical flow ranked in a very good position in the Middlebury evaluation [17], and some results can be appreciated in the Fig. 2.7.

Chapter 3

Object Flow Pipeline

3.1 Algorithm description

The Fig. 3.1 shows a simplified block diagram of the proposed system. Two details are important, the use of the tracker window to initialize a segmentation procedure, and the use of this segmentation over the tracked window to perform a more precise motion flow computation in the interest pixels. The dotted line represents the possible interaction between precise flow information and the next tracker state. For instance, the current object flow can work as direction hint, and the segmentation information can be used to improve the sampling process of the learning stage in several trackers by detection methods [22], and thus the tracker and motion flow algorithm can work for mutual enhancement.

The first step in the object flow pipeline can be selected according to specific need for a given application. We prefer, in general, tracking-by-detection methods like *Struck* [22] or MIL [23], but other approaches could be followed. In the second place, for the object segmentation in video we propose the use of labelled background regions through the concept of superpixel flow, which is explained in the next section.

3.2 Superpixel flow

3.2.1 Problem definition

Superpixels and over segmentation techniques became a widely used pre-processing stage for a large number of machine vision applications, after the original concept was introduced [1]. Superpixels are traditionally used as performance booster for several other techniques. However, it is still mostly related to single frame processing [1] [10] [11]. In the search for consistency

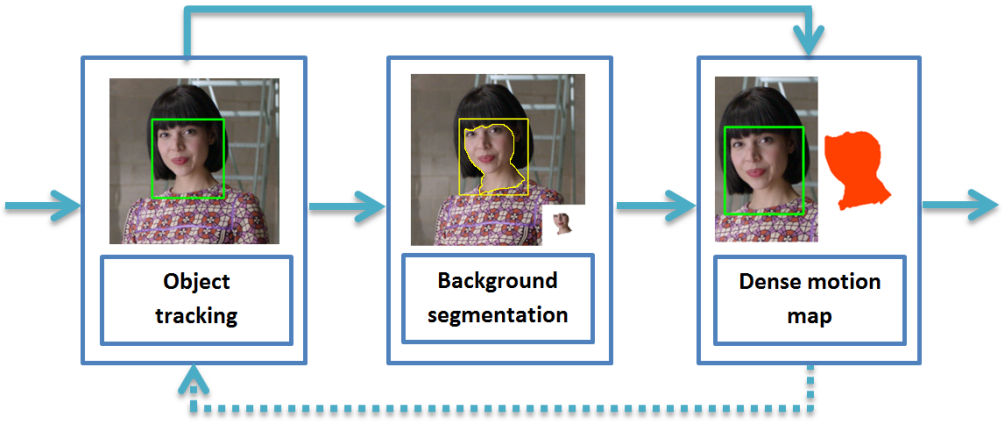

Figure 3.1: Block diagram of the proposed pipeline.

in superpixel labelling through video, some authors have proposed different techniques, which go from simple extension to supervoxels [9] [11], to more complicated approaches [8]. These approaches, nonetheless, usually require a global processing and knowledge of all (or several of) the video frames beforehand.

As a preprocessing step in the object flow pipeline, we propose a superpixel matching technique which assumes a flow-like behaviour in the image sequences (natural video), which can be used to track superpixels. Some previous work have been done towards a superpixel based image comparison using the Earth Mover's Distance, by taking superpixels as bins of a global histogram [2]. The label propagation or superpixel flow can be achieved with this technique as a by-product, by selecting the superpixel in the second frame that maximize the EMD flow from each superpixel in the first frame. By taking into account superpixels computed separately in images, so the video process can be performed with only two frames at a time, we move towards a more time efficient approach. This matching, however, has to comply with a set of constraints. Firstly, two correspondent superpixels should be similar in terms of some appearance feature, which most likely depends on the way the superpixelization was performed (color, texture, shape). Also, the superpixel flow should maintain certain global regularity (at least for superpixels that belong to the same object). In this sense, it seems natural that the problem of superpixel flow could be solved with a discrete energy minimization procedure. If the size compactness of the superpixels is maintained, it actually seems to share some of the properties of the optical flow problem, with the difference that the smoothness is usually a very strong constraint for the last one. The strength of this smoothness prior relies not only in the nature of the problem, but also because it gives better cues towards an easier-to-minimize

global approach.

The objective of the superpixel flow is therefore to find the best labeling l for every superpixel p (with $l_p \in 0, 1, ...N-1$) between a pair of frames (I_0,I_1), but holding a flow-like behavior.

Thus, the superpixelization should maintain certain size homegenity within a single frame. Some super pixel techniques can cope with this requirement [9] [10]. For the experiments presented in this work, the SLIC method [9] is prefered, because it usually gives good results in terms of homegenity of the superpixelization across the sequence. The proposed steps to solve the propagation problem assume this requirement is hold. For other kind of the techniques, other approaches should be followed.

3.2.2 Energy Formulation

Inspired by a large number of optical flow and stereo techniques [7] [12] [13], the superpixel flow can be modeled with pairwise Markov Random Fields. If the matching is performed with MAP inference, its posterior probability is:

$$P(l|I_0, I_1) = \prod_{p\in\Omega} \mathrm{e}^{-D_p(l_p;I_0,I_1)} \prod_{p,q\in\mathcal{N}} \mathrm{e}^{-S_{p,q}(l_p;l_q)} \tag{3.1}$$

With l the set of labels of the super pixels in I_0, that match with those in I_1. $\mathcal{N}$ is a neighbourhood of the superpixel p, which defines its adjacency. Given this posterior probability, the equivalent energy function can be directly obtained by extracting the negative logarithm of the posterior,

$$E(l) = \sum_{p\in\Omega} D_p(l_p; I_0, I_1) + \sum_{p,q\in\mathcal{N}} S_{p,q}(l_p, l_q) \tag{3.2}$$

The terms D, and S in (3.2) stand for data term and spatial smoothness terms as they are popularly known in the MRF literature. The first one determines how accurate is the labelling in terms of consistency of the measured data (color, shape, etc.). In the classical optical flow formulation of this equation, the data term corresponds to the pixel brightness conservation [7] [5]. However, as superpixels are a set of similar (or somehow homogeneous) pixels, an adequate color based feature can be a low dimensional color histogram. So D can be written more precisely as the Hellinger distance between the histograms:

$$D_p(l_p; I_0, I_1) = \sqrt{1 - \frac{1}{\sqrt{\bar{h}(p)\bar{h}(p')N^2}} \sum_i \sqrt{h_i(p)h_i(p')}} \tag{3.3}$$

Where $h(p)$ and $h(p')$ are the histograms of the superpixel p and its correspondent superpixel in the second frame I_1. Note that the low dimensional histogram gives certain robustness against

noise, and slowly changing colors between frames.

In the other hand, the spatial term is a penalty function for horizontal and vertical changes of the vectors that have origin in the centroid of the superpixel of the first frame and end in the centroid of the superpixel of the second frame.

$$S_{p,q}(l_p, l_q) = \lambda(p)\sqrt{\frac{|u_{p_c} - u_{q_c}|}{\|p_c - q_c\|} + \frac{|v_{p_c} - v_{q_c}|}{\|p_c - q_c\|}} \tag{3.4}$$

$$\text{where, } \lambda(p) = (1 + \rho(h(p), h(q)))^2$$

In (3.4) the operator ρ is the Hellinger distance as used in the data term (3.3). The histogram distance is nonetheless computed between superpixels p and q, which belong to the same neighbourhood. The superpixels centroids are noted as q_c and p_c, and u and v are the horizontal and vertical changes between centroids. This term is usual in the MRF formulation and has a smoothing effect in superpixels that belong to the same object. It has to be observed that when two close superpixels are different, thus, more probable to belong to different objects within the image, the term λ allows them to have matches that do not hold the smoothness prior with the same strength. It has to be noted that the proposed energy function is highly non-convex.

3.2.3 Energy Minimization

A fair amount of work has been dedicated to discrete optimization techniques in computer vision, leading to well-defined and widely tested approaches to solve pairwise MRF [3] [4]. However, some of the approaches restrict the construction of the spatial term, and/or enforce limitations in the number of labels [3]. Because of the high amount of possible labels for each superpixel in the proposed approach, the use of the Fusion Moves [7] technique seems to be well suited. This algorithm employs the Quadratic Pseudo-Boolean Optimization (QPBO), to combine incremental sets of proposal labellings, resulting in a semi-globally-optimal solution [4]. Thus, the minimization starts by proposing a set of possible solutions, and iteratively merges them with the QPBO technique.

The candidate solutions depend on the problem to be solved. For example, in stereo superpixel matching, some assumptions related to the cameras layout can be made to generate solutions. In a more generic sense, other assumptions can be made towards candidate generation. The Quadratic Pseudo-Boolean Optimization (QPBO) [3] [4] is used to minimize the proposed energy function, by merging a set of candidate matches for every superpixel in the first frame. For instance, for a given superpixel in the initial frame, the corresponding matching would be the most similar one in terms of color, shape, or the spatial distance. More candidate solutions can be added by defining a neighbourhood in the second frame and select random pairs

from every neighbourhood of every superpixel in the first frame. This is suitable for problems where the images are extracted from the same video sequence. To speed-up the minimization procedure, the QBPO properties can be exploited. For instance, the fusion of the proposed solutions is always guaranteed to be of lowest or equal energy than the two proposals. Thus, one could split the fusion procedure in several cores and build a hierarchical chain as fusions of proposal are subsequently fused.

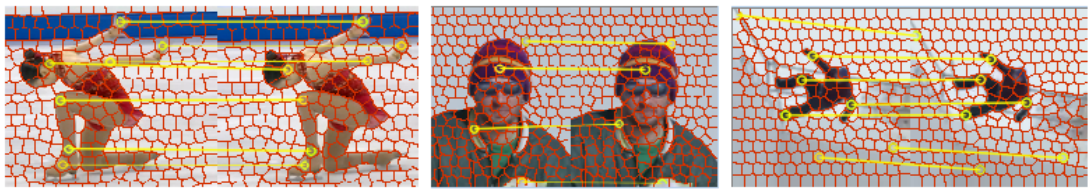

Figure 3.2: The yellow lines show selected superpixel matching between pairs of consecutive frames in a video with the proposed method. The video frames go from right to left.

3.2.4 Matching Results

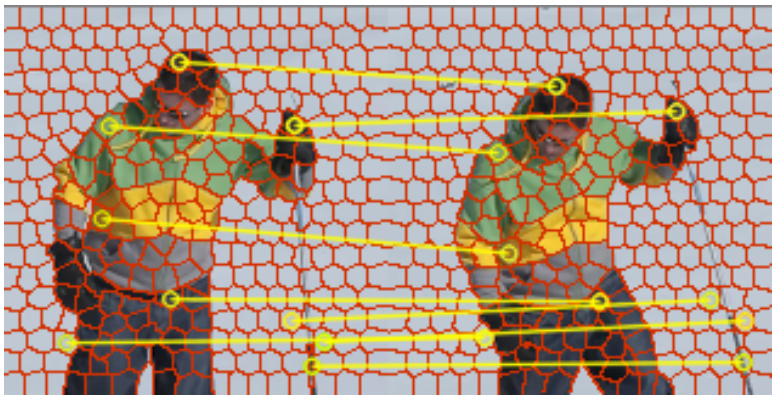

Figure 3.3: The yellow lines show selected superpixel matching between a pair of distant frames in the Snow Shoes sequence.

The Fig. 3.2 shows some examples of superpixel matching between subsequent frames with the presented method. It can be seen that the matching performs well even in difficult cases, like the hands in the top row. It has to be noted as well that even in superpixels where there is a lack of texture, there is correct matching. This seems to be the effect of enforcing the regularization between superpixels that are close, but are also similar to each other.

Moreover, unlike most of the optical flow methods, superpixel flow extends naturally for more distant frames. The Fig. 3.3 shows results for large separations between frames, without tweaking or adjusting any parameters. For this case, however, the matches in the texture-less part of the scene are mostly invalids. Though this is expected because of the aperture problem and heavy occlusions.

3.3 Background regions tracking and segmentation

The algorithm proposed in [18], offers a good deal in terms of background-foreground separation from user interaction. A technique like this, however, performs very well in still images, but it may not be well adapted for sequential videos. Extensions to this method, like the GrabCut algorithm ([14]), work by implementing an iterative graph-cut based minimization to separate regions according to appearance information from a loosely drawn rectangle around the object, and small user-interaction-based hints. Given the tracker state for every frame, the minimization procedure of the methods in [18] and [14] could be extended to video. However, a lot of details in the segmentation contour may be lose if no fine hints are given. These hints usually depend on on-the-fly supervised methods. However, this need could be minimized in videos, given the extra information that offers the flow of the sequence. Some authors had approached the graph-cut based segmentation techniques in sequential videos to propagate a consistent segmentation [15]. However, some more work on reducing user interaction given the extra flow-like information that video sequences offer is still needed. We propose to combine the presented superpixel flow as an automatic initialization method for the desired segmentation method.

The main idea to perform object segmentation consist in tracking (or more exactly, matching) superpixels that are labelled as background, thanks to an object tracker initialization. Thus, the superpixels that are initially outside the tracker region of interest, can be propagated through the sequence, and if they fall into the window on a subsequent frame, they can be safely labelled as background (Fig. 3.4).

To save computational power, the tracked superpixels are limited to the ones that fall inside a control region (red box in the Fig. 3.4). Usually, after several frames, the labeled superpixels will almost completely cover the unwanted areas in a dinamyc scene. We call this process background segments tracking. The Fig. 3.5 shows this idea in a real scenario. From left to right, initially the superpixels with elements outside the bounding box are labeled as background (green), then, as the sequence changes, the labeled superpixels flow inside the window, giving hints for the model initialization in the background-foreground separation algorithm. At this point, some generic segmentation technique can be connected to the pipeline to refine the

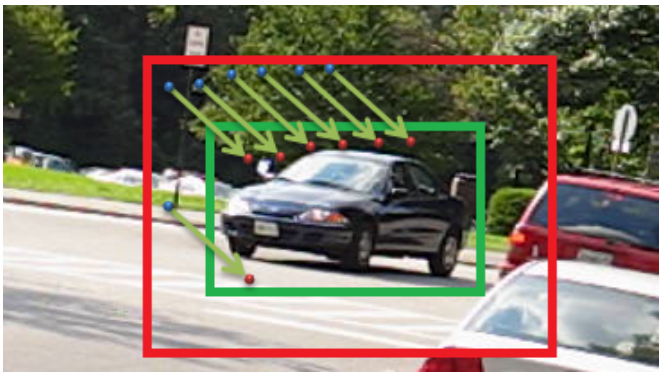

Figure 3.4: Example image of points entering a tracking region (green) due to object motion in a video sequence.

segmentation (e.g. region growing). We prefer, however graph based segmentation methods ([18] [15]) because the usual user interaction can be replaced by the tracked background regions.

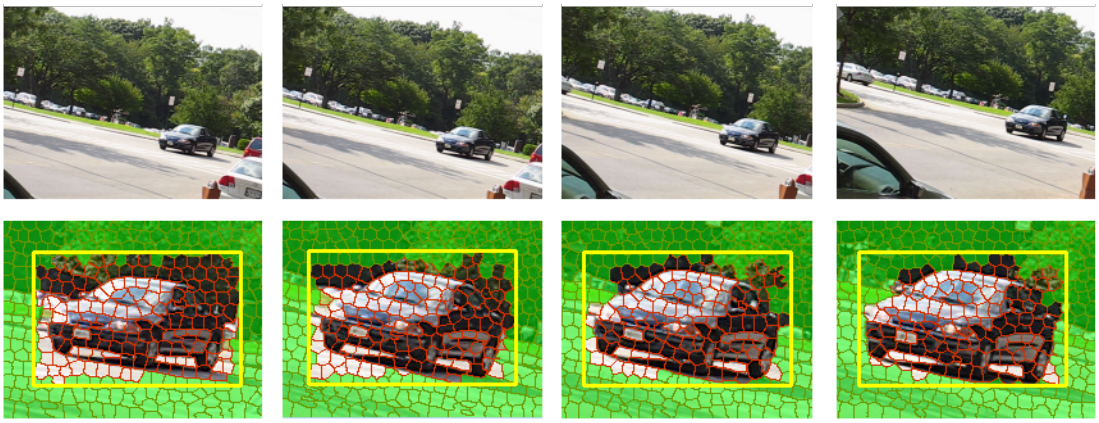

Figure 3.5: Background segments automatic labeling and propagation, the flow goes from left to right.

3.3.1 Segmentation results

Fig. 3.6 shows the results for an image sequence where the interest object is the head of a person. The head tracker and the superpixel flow provide information for better background-foreground separation. The background-foreground models are updated as the frames go on, giving more robustness for sequential propagation of the segmentation. The method is tested in the Walking Couple sequence, by allowing only a small amount of iterations in the graph based segmentation. Observe how the contour in the man's head is correctly delineated when another

person's head occludes part of it. In this case, the superpixels that belong to the womans face were correctly propagated and thus, labeled as background.

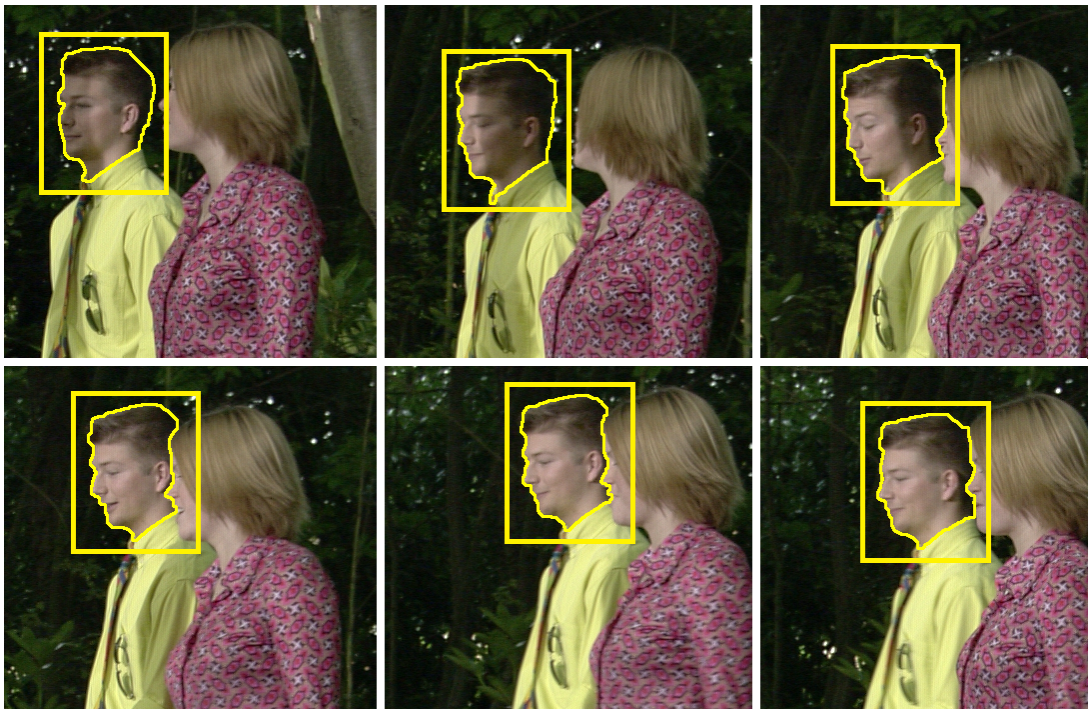

Figure 3.6: Segmentation through the sequence Walking Couple (Yellow contour) initialized in the mans head. The yellow box correspond to the tracker output. The labeled background superpixel are not shown for clarity.

In order to understand the effect of including superpixel propagation in a video sequence for object segmentation, some results are shown in the Fig. 3.7. For these experiments only one iteration is allowed in the graph-cut based methods. The top row frames (Fig. 3.7) were initialized only with the tracker, and the bottom row was initialized with the superpixel tracking technique. Observe that in general, the contour delineated is usually better in terms of precision and stability for the later one.

3.4 Flow estimation

The object flow consist on computing the motion field for an object of interest through an image sequence. The most usual approach to solve a problem like this is to implement some of the available optical flow techniques through the complete sequence and perform the flow

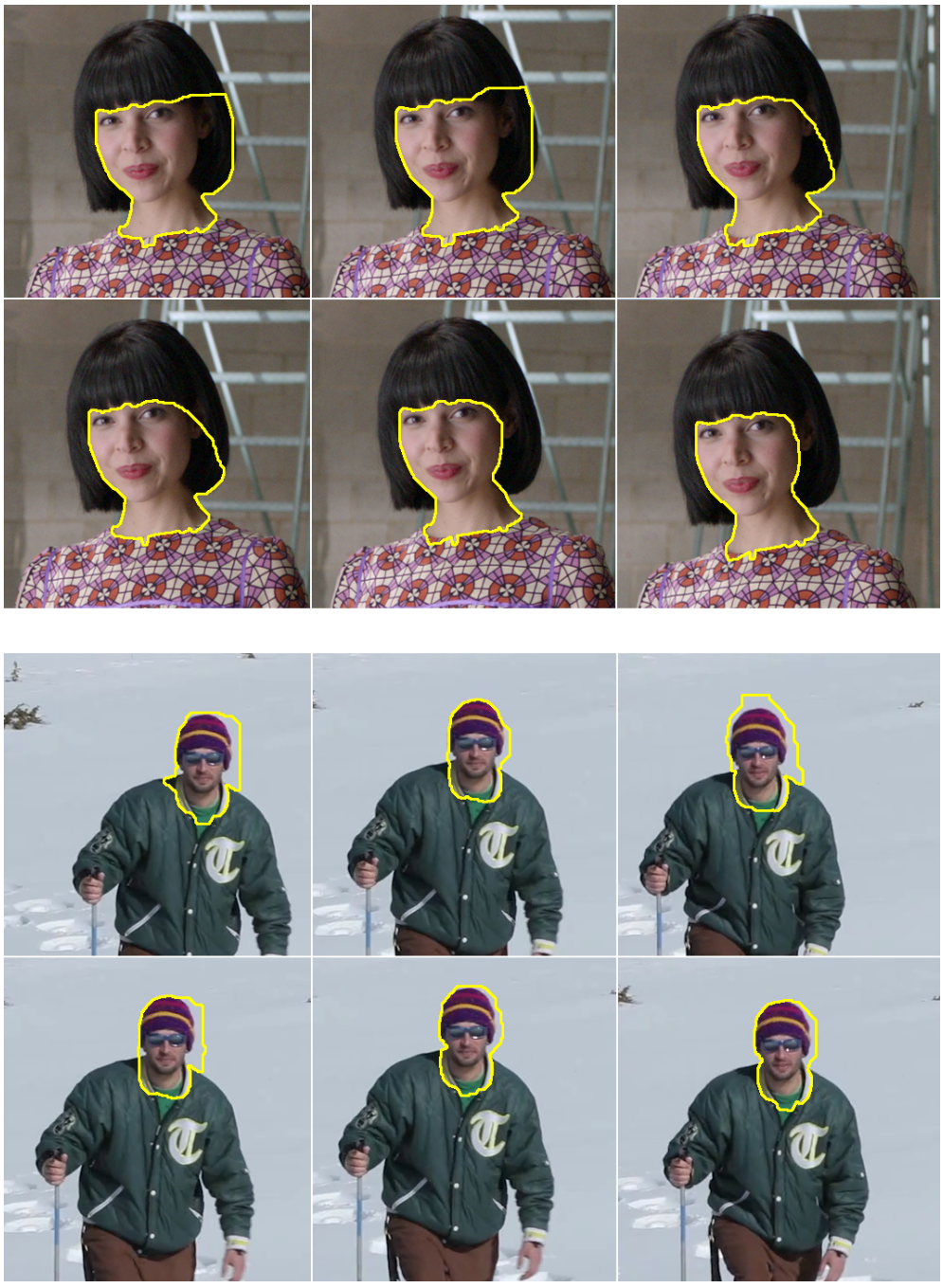

Figure 3.7: Face segmentation in the Amelie Retro and the Snow shoes sequences in three different frames. For each group, the Top Row: One-iteration window-based graph-cuts; and the Bottom Row: One-iteration graph-cuts initialized with superpixel tracking.

integration. However, this process results in high levels of motion drift [18] [19] and usually the motion of the interest object is affected by a global regularization. In some extreme cases, the interest object motion may be totally blurred and other techniques have to be incorporated. Moreover, the diversity of natural video sequences makes difficult the choice of one technique over another, even when specialized databases are at hand [17], because currently no single method can achieve a strong performance in every of the available datasets. Most of these methods consist in the minimization of an energy function with two terms (As was previously mentioned in the Sec. 3.2). The data term is mostly shared between different approaches, but the prior or spatial term is different, and basically states under what conditions the optical flow smoothness should be maintained or not. In a global approach, however, this is a difficult concept to define. Most of these smoothness terms rely in appearance differences or gradients. All these meaning that, unavoidably, some methods may be more reliable for some cases but weaker for others. It can be argued that this behaviour may be caused because most of the techniques do not count with a way to identify firmly where exactly this smoothness prior can be applied.

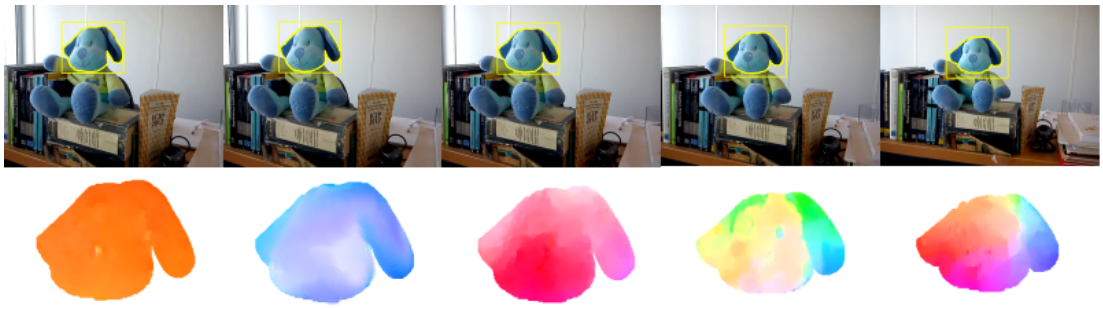

Figure 3.8: Object flow with the color code of [17] (bottom) for frames in the Puppy sequence (up).

The main idea behind the object flow is that given the availability of several robust tracking techniques, and the proposed segmentation method for video, the optical flow computation can be refined by computing it successively between pairs of tracked windows. The basic proposal to perform this refinement consist on considering the segmentation limits as reliable smoothness boundaries. This is, of course, under the assumption that the motion is indeed smooth within the object region. This is assumption is not far from reality in most scenes with an interest object. Naturally, as the object tracker is included, is expected that the object flow should be more robust to rapid motions than the optical flow. Thus, the full motion is split in two, the long range motion, given by the tracker window, and the precision part, given by the targeted optical flow. The Fig. 3.8 shows the object flow for a frame in the Puppy sequence. Observe the motion vectors are computed only inside the object of interest, preserving a strong smoothing

prior, but also allowing internal variations in the flow.

As a first approximation to the object flow, the Simple Flow technique [21] is taken as core base. This is because of its scalability to higher resolutions and because its specialization to the concept of object flow is only natural. This is because in the Simple Flow pipeline the smoothness localization can be easily specified through computation masks. More specifically, the initial computation mask is derived from the segmentation performed as prior step. The resulting flow is then filtered only inside the mask limits to enhance precision and fastening the implementation. However, direct modifications in other optical flow methods can be further studied. For instance, in graph-cut based minimization approaches, the regularity constraints can be precisely targeted by disconnecting foreground pixels from background ones.

Chapter 4

Results and Implementation Details

To evaluate the performance of the object flow in comparison with optical flow techniques, we performed a number of experiments on several video sequences. We annotated an initial bounding box for the videos, and a segmentation contour of the interest object for every frame. The experiment measures the ability of the method to extrapolate an image from the initial frame and the integrated flow. For every pair of frames the video sequence, the PSNR between the annotated current state of the object and the extrapolated images is computed. The Fig. 4.1 is a sample of the performed experiment, each column is an image generated from the given flow. Two types integration are evaluated, $From - the - reference$, or forward integration, and $To - the - reference$, or backward integration, as discused in [20]. So, for each row in the Fig. 4.1, two columns correspond to the object flow, and two columns correspond to optical flow, with both types of integration.

The Fig. 4.2 shows PSNR graphics for 4 different sequences. For every pair of frames an image is extrapolated, and the PSNR with the ground-truth object is computed. The results are shown with both, Euler integration (Labelled as $forward$ in the figs.) of the used flow, and using the integration method described in [20], labeled as *backward* in the figures. The results show that the object flow methods are generally more precise than its optical flow counterparts. Moreover, the object flow method with backward integration usually performs much better than any other combination of techniques. For this experiment, the object flow is compared with the simple-flow optical-flow method. This experiment directly shows that the object flow concept is indeed capable of increasing accuracy of a given optical flow method.

Now, in order to study how the object flow concept compares to several state of the art

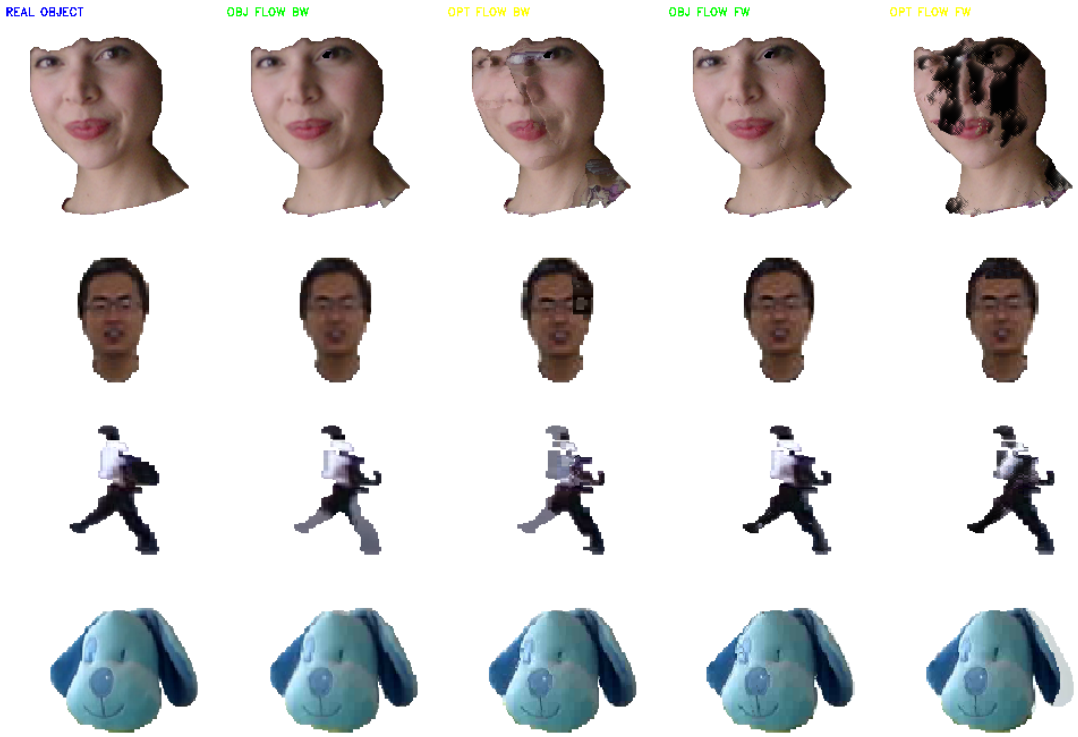

Figure 4.1: Extrapolation results from integrated flow in 4 sequences. In descending order: Amelia Retro, Boy, Walking, Puppy. From Left to Right: Annotated object, Backward object flow, Backward optical flow, Forward object flow, Backward optical flow.

optical flow methods, another extrapolation experiment is performed. The Fig. 4.3 presents a visual comparison between the object flow and several optical flow techniques in the Amelia sequence for object extrapolation, and the involved frames (the first and last used frames in the sequence). The Fig. 4.4 shows the PSNR results for every extrapolated frame in the full sequence, the object flow performs better than all the studied optical flow techniques.

Observe that the object details are lost in comparison with the ground-truth object image (Fig. 4.3). For example, the closed eyes detail is missing in the most of the optical flow methods. Furthermore, several of the methods lost any significance, and the output barely holds any resemblance with the original image. This is possibly the result of a couple of details that are naturally better attacked with the object flow: long motion (the tracker always centers the object in any given frame), and smoothness prior (the segmentation mask is a good delimitation to establish this prior).

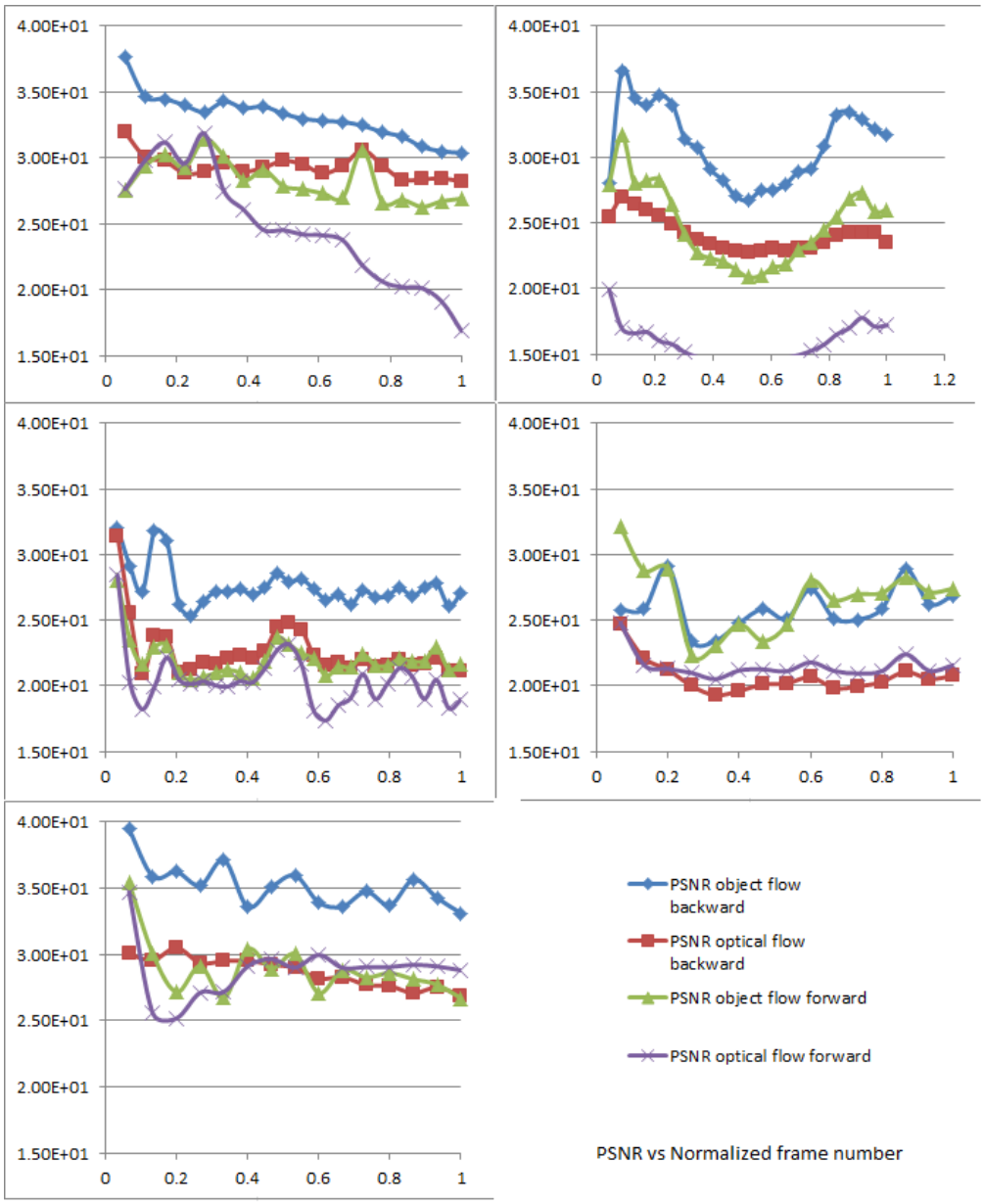

Figure 4.2: PSNR graphs for extrapolated images using Object flow and the Simple Optical Flow for 4 sequences. From left to right and up to bottom: Puppy Seq.; Amelie Retro Seq.; Boy Seq.; Walking Seq.

Figure 4.3: Top: The first frame and the accumulated flows are used to extrapolate objects in the frame number 30. The used methods from left to right: Groundtruth object, Object flow, TVL1, Block Matching, Brox, Farneback and Simple Flow. Bottom: First and frame#30. The extrapolations are performed using backward accumulation of the flows.

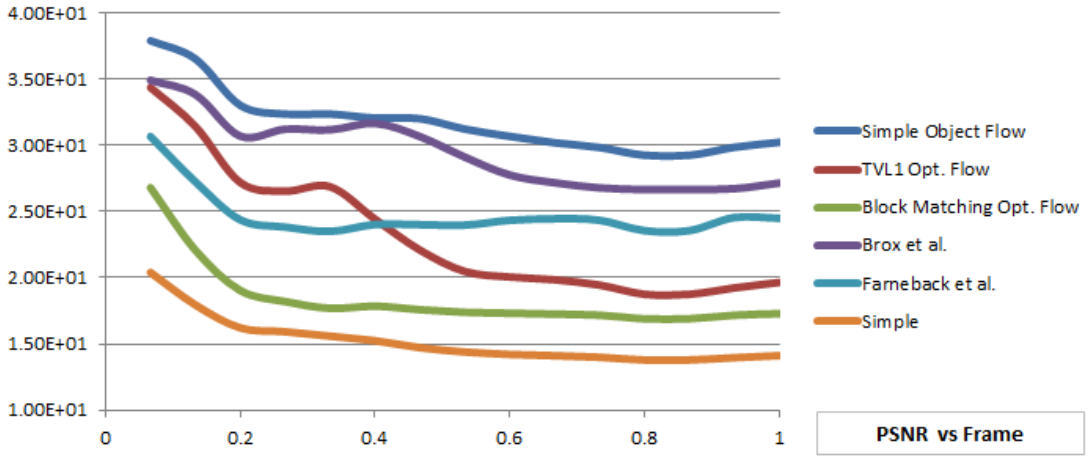

Figure 4.4: PSNR graphs for extrapolated images using Object flow and the different Optical Flow techniques for the Amelia sequence.

Chapter 5

Applications

A couple of applications for the object flow are presented. Firstly, in complex video edition and augmented reality, and secondly as part of the object-based structure-from-motion. Only visual results are provided in both cases.

5.1 Video edition

The video edition that is approached here is the insertion or modification of an element in a sequence. For instance, during post-production it is possible that an element of a sequence presents inadequate content for the expected audience, and usually removing these elements require a hard manual art-edition work. The object flow can be exploited in order to reduce drastically the amount of manual work that has to be done.

The proposed approach is to edit manually the initial frame. The elements of this edition can be propagated through a number of subsequent frames with the object flow. Moreover, for complex scenes, with heavy occlusions and sudden illumination changes, several instances of the object flow can be used, to take into account the difficult cases. Thus, the overall edition work is greatly reduced.

5.2 Structure from motion

The idea of the structure from motion problem (*SfM*) is to recover the shape of objects or scenes from a sequence of images obtained from a camera that follows certain motion. Usually, it is assumed that the scene contains rigid objects undergoing an Euclidean motion.

The most common way to solve the *SfM* is to find a set of correspondences between the

acquired frames by using a feature matching method, and use the epipolar geometry constraint for two images ($x_1^T E x_2 = 0$). Usually, the *8 points algorithm* is used to obtain a relative rotation and translation of the camera by factorizing the essential matrix (E). However, given the simplicity of the approach, the algorithm is very sensitive to noise and several other approaches have been proposed. One of the major weakness of such methods is still the feature matching (w.r.t precision and sparsity).

When matching is replaced by optical flow, the scene structure and the camera motion are tied together by the equation:

$$\mathbf{u}(x) = A(x)\mathbf{v}/Z + B(x)\omega \tag{5.1}$$

$$\text{where, } A = \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \end{bmatrix}$$

$$\text{and, } B = \begin{bmatrix} -xy & (1+x^2) & -y \\ -(1+y^2) & xy & x \end{bmatrix}$$

The optical flow, however, has been traditionally inaccurate for long term point tracking, and due to the fact that a least square optimal structure can be obtained from camera velocities, the research was mainly focused on capturing camera ego-motion.

As it has been demonstrated that the object flow increases accuracy of the point trajectories estimation, it can be connected to a *SfM* pipeline. Some results can be appreciated in following images.

Chapter 6

Conclusions

A framework to combine tracking and optical flow methods to improve object based dense motion description is presented. The pipeline is composed of three main steps, object tracking, segmentation and flow estimation. For the segmentation step a new promising video object segmentation algorithm was proposed, and, to the best of our knowledge, the introduced superpixel flow is the first energy based algorithm for superpixel matching. For the last step, we presented a flow estimation method based on a modification of the simple-flow method to use the obtained segmentation mask. The experiments showed that this object based flow estimation improves the dense motion estimation in comparisson to optical flow techniques. Future work can be further explored in the use of the object flow as feedback hint for tracking-by-detection methods. Also, several kind of applications of the object flow can be more deeply approached. For instance, in the structure-from-motion pipeline, video based rendering, automatic video edition, and video inpainting among others.

Bibliography

[1] J. Malik and X. Ren, Learning a classification model for segmentation, *Computer Vision, International Conference*, 2003.

[2] S. Boltz; F. Nielsen and S. Soatto, Earth mover distance on superpixels, *International Conference on Image Processing*, 2010.

[3] E. Boros; P. Hammer and G. Tavares, Preprocessing of unconstrained quadratic binary optimization, *RUTCOR*, 2010

[4] E. Boros and P. Hammer, Pseudo-boolean optimization, *Discrete applied Mathematics.*, 2002

[5] B. Horn and B. Schunck, Determining Optical Flow, *Artificial Intelligence*, 1981

[6] H. Ishikawa and P. Bouthemy, Multimodal estimation of discontinous optical flow using Markov random fields, *TPAMI.*, 1993

[7] V. Lempitsky, S. Roth and C. Rother, Fusion Flow: Discrete-Continuos optimization for optical flow estimation, *Computer Vision and Pattern Recognition*, 2008

[8] M. Reso and J. Jachalsky, Temporally Consistent Superpixels, *International Conference Computer Vision.*, 2011

[9] R. Achanta; A. Shaji; K. Smith; Aurelien Lucchi; P. Fua and S. Susstrunk SLIC, Superpixels compared to state of the art superpixel methods, *Discrete applied Mathematics.*, 2002

[10] F. Perbet and A. Maki, Homogeneus superpixels from random walks, *MVA.*, 2011

[11] C. Xu and J.J. Corso. Evaluation of super-voxel methods for early video proccesing, *Computer Vision and Pattern Recognition.* 2012.

[12] A. Shekhovtsov, I. Kovtun and V. Hlavac. Efficient MRF deformation model for non-rigid image matching, *Computer Vision and Pattern Recognition*. 2007.

[13] J. Sun, N.N Shen and H.Y. Shum. Stereo matching using propagation belief, *TPAMI*. 2003.

[14] C. Rother, V. Kolmogorov and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts, *SIGGRAPH*. 2004.

[15] L. Yang, Y. Guo, X. Wu and X. Wang. A new video segmentation approach: Grabcut in local window, *Soft Computing and Pattern Recognition*. 2011.

[16] Y. Wu, J. Lim and M.-H. Yang. Online object tracking: A benchmark, *Computer Vision and Pattern Recognition*. 2013.

[17] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black and R. Szeliski. A Database and Evaluation Methodology for Optical Flow, *International Journal Computer Vision*. 2013.

[18] Y. Boykov, M-P. Jolly. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D images, *International Conference on Computer Vision*. 2013.

[19] W. Li, D. Cosker and M. Brown. An anchor patch based optimization framework for reducing optical flow drift in long image sequences, *Asian Conference on Computer Vision*. 2012.

[20] T. Crivelli, P.-H. Conze, P. Robert, M. Fradet and P. Perez. Multi-step flow fusion: towards accurate and dense correpondence in long video shots, *British Conference Machine Vision*. 2012.

[21] M. Tao, J. Bai, P. Kohli, and S. Paris. SimpleFlow: A Non-iterative, Sublinear Optical Flow Algorithm, *Computer Graphics Forum, Eurographics*. 2012.

[22] T.B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring Suporters and Distracters in Unconstrained Environments. *Computer Vision and Pattern Recognition* . 2011.

[23] S. Hare, A. Saffari, and P.H.S. Torr, Struck: Structured Output Tracking with Kernels. *International Conference on Computer Vision*. 2011.

[24] X. Jia, H. Lu, and M.H. Yang. Visual Tracking via Adaptive Structural Local Sparse Appearance Model. *Computer Vision and Pattern Recognition* . 2012.

[25] B. Babenko, M.H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. *Computer Vision and Pattern Recognition*. 2009.

[26] Z. Kalal, J. Matas, and K.Mikolajczyk. P-N Learning: Bootstrapping Binary Classiers by Structural Constraints. *Computer Vision and Pattern Recognition*. 2010.

[27] D.J. Butler, J. Wulff, G.B. Stanley, and M.J. Black. A naturalistic open source movie for optical flow evaluation.*European Conference in Computer Vision*. 2012.

[28] G. Farneback. Two-Frame Motion Estimation Based on Polynomial Expansion. *Scandinavian Conference, SCIA* . 2003.

[29] Brox, A. Bruhn, N. Papenberg, J. Weickert. High accuracy optical flow estimation based on a theory for warping. *European Conference in Computer Vision*. 2004.

[30] Berthold K.P. Horn and Brian G. Schunck. Determining Optical Flow. *Artificial Intelligence* . 1981.

[31] Lucas, B., and Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision, *International Joint Conference on Artificial Intelligence* . 1981.

[32] Zach, T. Pock and H. Bischof. A Duality Based Approach for Realtime TV-L1 Optical Flow, *Proceedings of Pattern Recognition (DAGM)* . 2007.

[33] T. Brox and J. Malik. Object Segmentation by Long Term Analysis of Point Trajectories, *European Conference in Computer Vision*. 2010.

[34] P. Ochs and T. Brox. Object Segmentation in Video: A Hierarchical Variational Approach for Turning Point Trajectories into Dense Regions, *International Conference in Computer Vision* . 2011.

[35] A. Bugeau and P. Perez. Detection and segmentation of moving objects in highly dynamic scenes, *Computer Vision and Pattern Recognition* . 2007.

[36] S.M. Smith. ASSET-2: Real-Time Motion Segmentation and Shape Tracking. *Computer Vision*. 1995.

[37] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. *Carnegie Mellon University Technical Report CMU-CS-91-132* , 1991.

[38] A. Hilsmann and P. Eisert. Deformable Object Tracking Using Optical Flow Constraints. *European Conference in Visual Media Production* , 2007.

[39] H-S. Chang and Y-C. Frank Wang. Superpixel-Based Large Displacement Optical Flow. *International Conference on Image Processing* , 2013.

[40] D. Sun, S. Roth J.P. Lewis, and M.J. Black. Learning Optical Flow. *European Conference on Compute Vision* , 2008.