OBJECT FLOW

A per-object dense motion descriptor

Juan Manuel Pérez Rúa

Technicolor SA - University of Burgundy

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

List of Tables

Acknowledgments

...

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 [**?**] [**?**]. The object tracking problem consist on estimating the position of the target in future frames, given an initialization. In the other hand, the optical flow between a pair of frames consist on finding a motion vector for each pixel of interest in the initial image. Even though for several applications a full 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 motion vectors within its space may be of interest. In such scenarions combining optical flow and object tracking in a unified framework would become useful and the precision of the object motion description could be enhanced. For instance, even with modern optical flow approaches, the long term motion problem remains a challenge. However, the problem is more bearable for object tracking techniques. In contrast, object trackers are more global in motion description, and its information can be completed by optical flow subpixel precision. Moreover, even when object trackers and optical flow could give good hints for object segmentation in video, these elements are not deeply studied in the literature as a unified problem. We introduce the object flow problem as the computation of dense motion flow fields of the set of pixels that belong to an interest object. In other words, the object flow by definition induces the segmentation of the target and its motion field.

We can define more precisely the object flow by starting with an image sequence 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 vector $d_{0,t}(x)$ from the image I_0 to I_t, $\forall x \in \mathcal{R}$.

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 position by a graph-cut based 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, starting by the segmentation.

Among the state of the art segmentation methods for objects in video sequences, point trajectories based ones stand for its performance and reliability, even when only sparse trajectories are known because of computational reasons. 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. We propose to mix these two ideas together with the tracking of background regions via the novel concept of superpixel flow for reliable object segmentation through video. We show how this extra information can be used to complement the graph-cuts based techniques for an efficient foregorund-background segmentation.

Discussion on the implementation details and experiments showing the performance of the proposed algorithms (the superpixel flow, background regions tracking, video based object segmentation and the object flow) are presented in subsequent sections.

Chapter 2

Background

2.1 Introduction

Many scientific subject areas (i.e. medicine, remote sensing) have to deal with the problem of image correspondence (also known as image registration or image matching). This problem arises when there is a need to extract information about an object by comparing a set of different images in which this object appears. Most of the time this comparison cannot be performed easily because of differences between images such as viewpoint changes, use of different sensors, images taken at different times, and changes in general imaging conditions. Given such differences, a way to align different images of the same object is needed. In medical applications, and specifically in mammography, radiologists have to deal with all the above problems. For example, a way to detect abnormal structures in the breast is to compare temporal (images of the same breast taken at different times) or contralateral (using left-right breast) mammograms.

As images are taken under different conditions (i.e. film sensitivity, radiation exposure, breast compression and patient movement) and/or time intervals (e.g. screening interval of three years) the structure of the breast is likely to suffer some changes, although an overall similarity will be maintained. A way to detect those changes is to align the images and compare the results using, for instance, simple image subtraction. In addition, as described in the previous chapter, different imaging modalities are being used to provide a better understanding of a region of interest. The first step to incorporate information from those modalities is to align them to be able to establish areas of correspondence.

An image correspondence method is based on finding a mapping function ($f(x, y)$) that maps each coordinate of one image (A) into another image (B).

$$B(x', y') = A(f(x, y)) \tag{2.1}$$

which maps spatial coordinates (x, y) in the reference image A to coordinates (x', y') in the warped (also referred to as target) image B. Usually the function f is expressed as two (or three in three dimensions) separate functions f_x and f_y (and f_z). This review describes image registration in two dimensions. However extrapolation to 3D is often straightforward and is described in detail where appropriate.

A general methodology for image alignment typically follows those steps:

1. *Selection and extraction of features.* A registration method can be based on matching different image primitives. For instance points, edges, ridges, surfaces, or a whole image. The selection of a suitable primitive is a trade off between the information it provides and its complexity. For instance, raw pixels with only intensity information provide a straightforward comparison. On the other hand, taking the whole image provides more (and more complex) information. Primitives will be described by a set of features extracted from them. That is the case, for instance, of shape features extracted from regions or curvature values extracted from linear structures. There is an extensive literature about feature extraction, the reader is referred to Chapter 4 where a review on that subject is given.

2. *Similarity metric.* A way to measure the similarity between the extracted features is needed. Many different similarity metrics have been proposed and their suitability depends on the chosen feature. The reader is referred to Chapter 5 where similarity measures are reviewed and a novel measure is proposed.

3. *Selection of the mapping function and estimation of its parameters.* The complexity of a mapping function should be determined depending on the application field and type of misalignment we are dealing with. Once a mapping function is selected its parameters need to be estimated. This requires a definition of an optimum search space and search strategy which is often related to optimisation of a cost function related to a similarity measure previously mentioned.

4. *Alignment of images using the mapping function.* When the function parameters are known we are able to transform an image in order to minimise the initial misalignment.

2.1.1 Interpolation

Usually transformed coordinates (x', y') in the warped image do not match a given coordinate grid (i.e. they are non-integer values) and interpolation is needed. Different interpolation

methods have been used in image registration: nearest neighbour, tri-linear and partial volume distribution. The interpolation problem is graphically represented in Figure 2.1.

Figure 2.1: Interpolation of corresponding coordinates

Nearest neighbour interpolation assigns the intensity value of $m = (x', y')$ to its nearest point:

$$B(m) = B(\_\forall imin(d_E[m, n_i])) \tag{2.2}$$

where d_E indicates Euclidean distance and n_i are the nearest neighbours. The nearest neighbour approach provides good accuracy and it is easy to compute.

Trilinear interpolation assigns to m the value of the weighted (w_i) sum of the neighbouring pixels.

$$B(m) = \sum_i w_i B(n_i) \tag{2.3}$$

where each weight w_i is related to the distance from m (see Figure 2.1)

$$w_1 = (1 - dx)(1 - dy) \quad w_2 = (1 - dx)dy \quad w_3 = dx(1 - dy) \quad w_4 = dxdy \tag{2.4}$$

This interpolation creates a new intensity value, which could introduce undesirable effects in the intensity distribution. This fact is regarded as its main drawback, and can be solved using a partial volume distribution. This method adds the weight of each neighbour (w_i values) at the intensity value in the histogram distribution, without creating an additional intensity value.

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 [**?**], 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* [**?**] or MIL [**?**], 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 [**?**]. Superpixels are traditionally used as performance booster for several other techniques. However, it is still mostly related to single frame processing [**?**] [**?**] [**?**]. In the search for consistency in

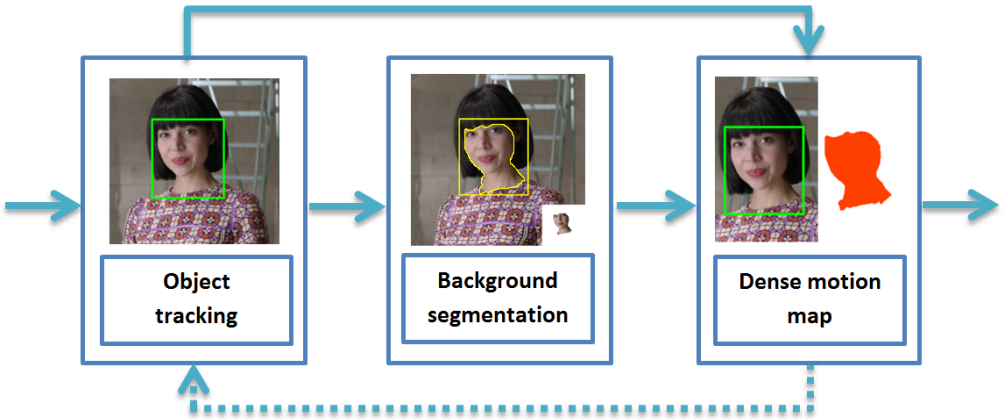

Figure 3.1: Block diagram of the proposed pipeline.

superpixel labelling through video, some authors have proposed different techniques, which go from simple extension to supervoxels [**?**] [**?**], to more complicated approaches [**?**]. 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 [**?**]. 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 [**?**] [**?**]. For the experiments presented in this work, the SLIC method [**?**] 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 [**?**] [**?**] [**?**], 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 neighborhood 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 [**?**] [**?**]. 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 neighborhood. 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 [**?**] [**?**]. However, some of the approaches restrict the construction of the spatial term, and/or enforce limitations in the number of labels [**?**]. Because of the high amount of possible labels for each superpixel in the proposed approach, the use of the Fusion Moves [**?**] technique seems to be well suited. This algorithm employs the Quadratic Pseudo-Boolean Optimization (QPBO), to combine incremental sets of proposal labelings, resulting in a semi-globally-optimal solution [**?**]. 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) [**?**] [**?**] 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.

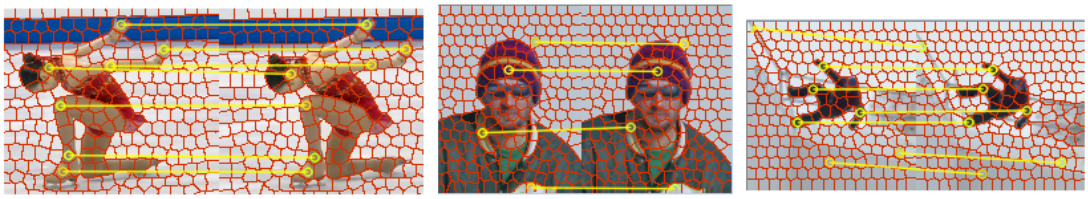

Figure 3.2: The yellow lines show selected superpixel matching between pairs of consecutive frames in a video with the proposed method. The video frames go from right to left.

3.2.4 Matching Results

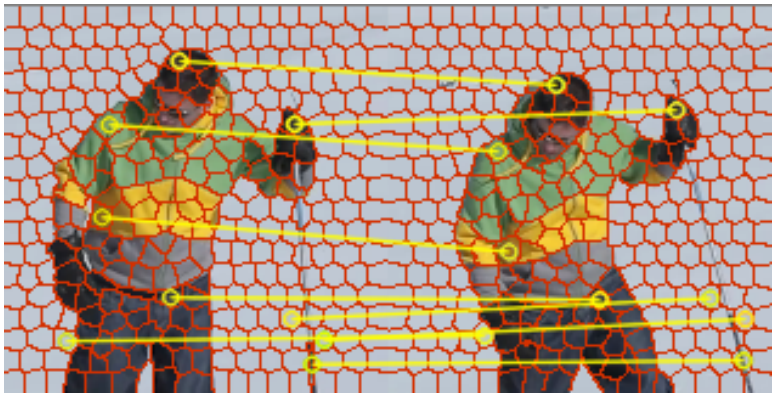

Figure 3.3: The yellow lines show selected superpixel matching between a pair of distant frames in the Snow Shoes sequence.

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 [**?**], 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 ([**?**]), 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 [**?**] and [**?**] 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 [**?**]. 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. **??**).

To save computational power, the tracked superpixels are limited to the ones that fall inside a control region (red box in the Fig. **??**). 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. **??** 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

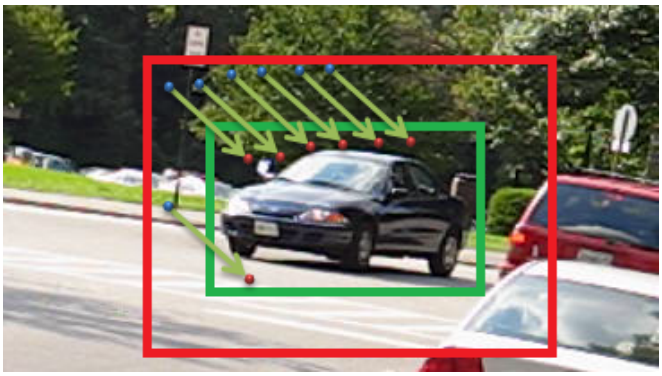

Figure 3.4: Example image of points entering a tracking region (green) due to object motion in a video sequence.

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

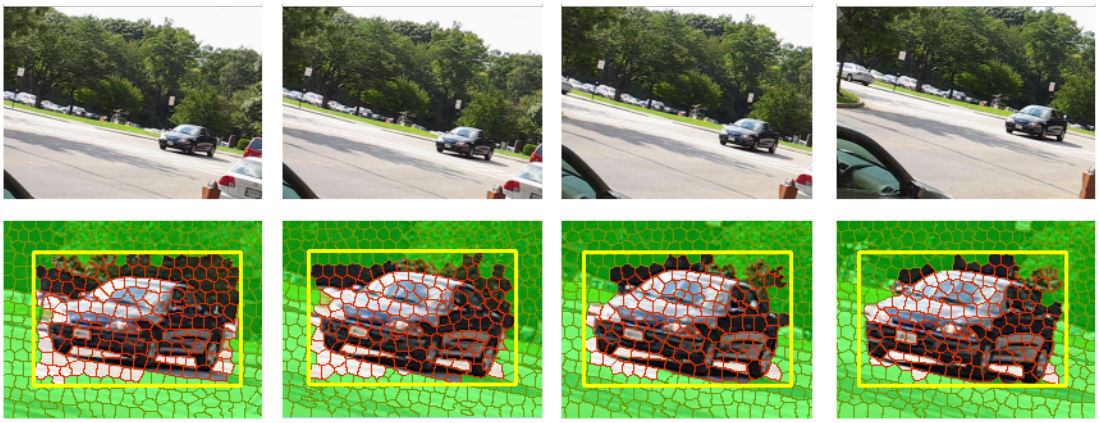

Figure 3.5: Background segments automatic labeling and propagation, the flow goes from left to right.

3.3.1 Segmentation results

Fig. **??** 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.

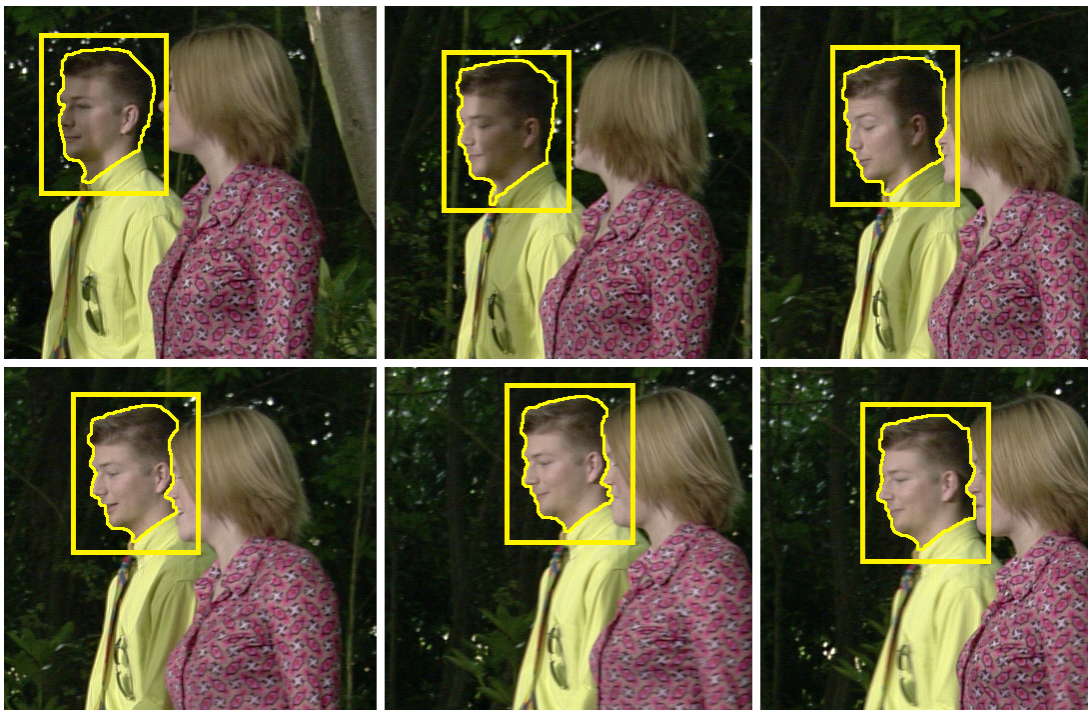

Figure 3.6: Segmentation through the sequence Walking Couple (Yellow contour) initialized in the mans head. The yellow box correspond to the tracker output. The labeled background superpixel are not shown for clarity.

In order to understand the effect of including superpixel propagation in a video sequence for object segmentation, some results are shown in the Fig. **??**. For these experiments only one iteration is allowed in the graph-cut based methods. The top row frames (Fig. **??**) 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.

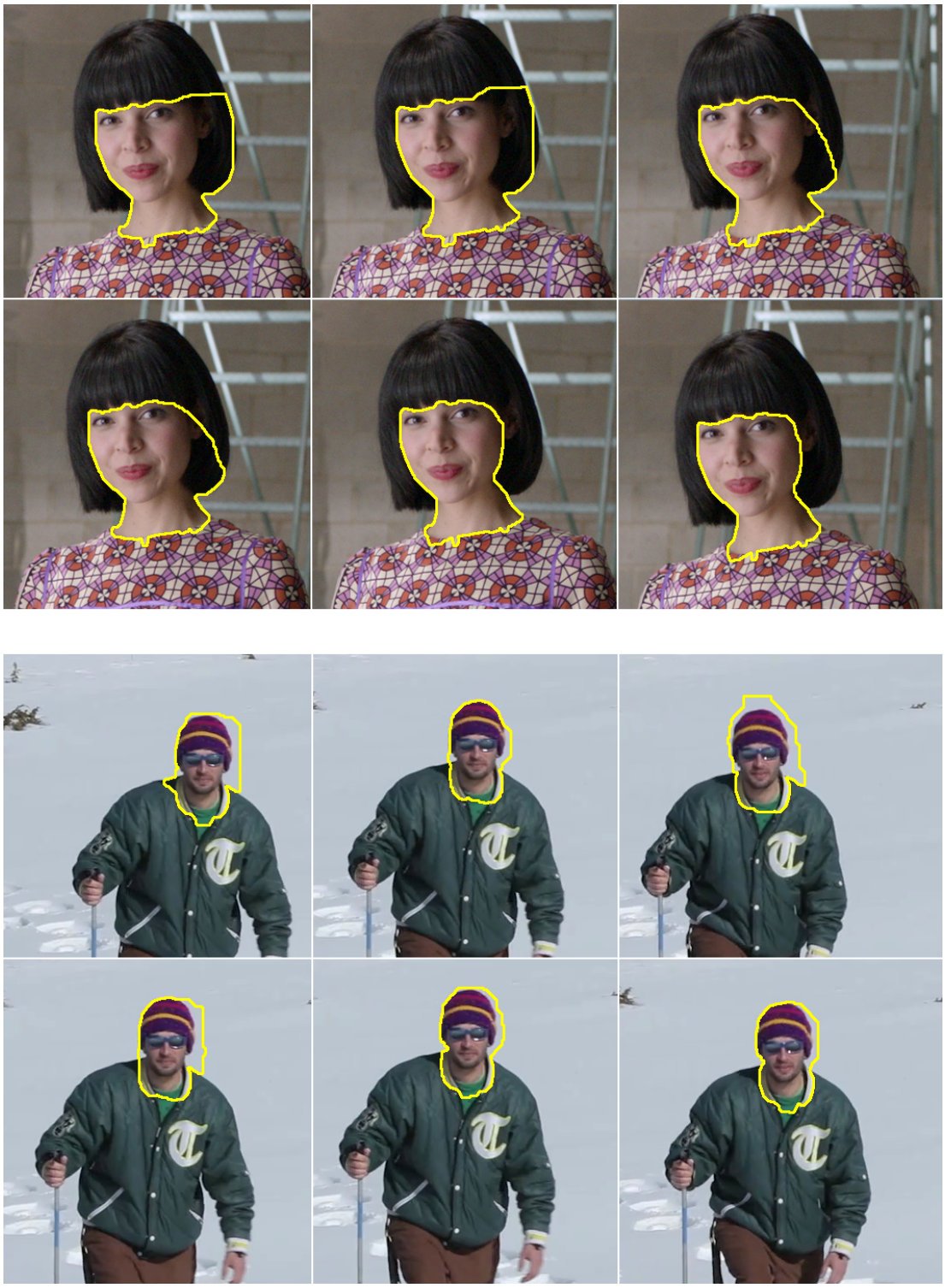

Figure 3.7: Face segmentation in the Amelie Retro and the Snow shoes sequences in three different frames. For each group, the Top Row: One-iteration window-based graph-cuts; and the Bottom Row: One-iteration graph-cuts initialized with superpixel tracking.

Appendix A

The first appendix

If you need to add any appendix, do it here... Etc.

Bibliography