Object flow

Anonymous ACCV 2014 submission

Paper ID ***

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 graphs. This is done by exploiting the inherent foreground-background separation hints given by object trackers, and the novel concept of superpixel flow. Then, we show 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. We may call the proposed approach *Object flow* as it offers a dense and semantic aware description of the current motion state of the studied object.

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.

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, $\mathcal{O}(x)$, is defined as $\mathcal{O}(x) = d_{0,t}(x), \forall x \in \mathcal{R}$.

A straightforward solution to this problem would be to compute the optical flow field between a pair of frames, and to 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 prior. Moreover, finding the pixels that belong to the interest object in several frames is a difficult problem. We propose an approach to reduce these problems.

2 The object flow estimation pipeline

Following the definition of the problem given above we have devised a system for computing the object flow. The Fig. 1 shows a simplified block diagram of the proposed system. It is important to recognize the two main components of our work-flow: object tracking and segmentation, and pixel-wise flow computation on the pixels of interest. The scheme is completed by feeding back the flow information to improve the tracker as depicted in the figure. For instance, one can make the most of dense displacement vectors to refine the motion of the target, and the segmentation information can be used to improve the sampling process of the learning stage in several trackers by detection methods [22]. Thus, the tracker and motion flow algorithm can work for mutual enhancement.

Tracking. The object tracking in the object flow pipeline can be selected according to a specific need for a given application. Here we prefer recent methods based on tracking-by-detection given that they are in the top of modern benchmarks for object tracking [16] in terms of accuracy and stability to initialization changes. Even more, as previously mentioned tracking-by-detection techniques ca benefit from the background-foreground separation of the next stage in the object flow pipeline. For instance, the sampling process in the *Struck* tracker can be improved by selecting positive samples only inside the region of the target object.

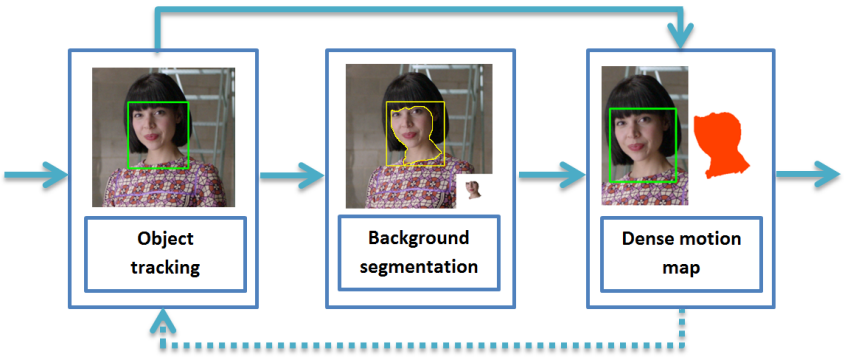

Fig. 1. Block diagram of the object flow pipeline.

Object Support Extraction. Determining the spatial support of the object in a given frame benefits from the output of the tracker. Appearance cues alone, learned inside and outside the tracking window can result in a misleading modeling of the foreground and the background. In contrast, we propose to perform foreground-background segmentation by tracking background pixels surrounding the target, thanks to the tracker output. Thus, the pixels that are initially outside the tracker window, are followed through the sequence and as long as they enter the tracked region, they can be safely labeled as background. This idea can be observed in the Fig. 2, where the object window given by the tracker (green) loosely separates the foreground from the background. Points outside the tracker (blue) are labeled as background in previous frames, and as they enter the tracker window (red points), they can be used to improve the modeling of the foreground and the background. The red window is used to save computational power by avoiding to track points that are too far from the interest object.

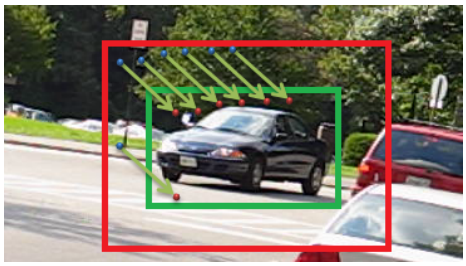

Fig. 2. Example image of points entering a tracking region (green) due to object motion in a video sequence.

Tracking pixels as independent points, however, carries several problems. First, to track all the background points means to compute the optical flow between the image pair, which can become very inneficient when taking into account large objects. Second, the point tracking techniques causes some drift in the trajectories, which can lead to wrong labelling as background. Finally, the background labelling could be more dense, and less complex if superpixels are tracked instead of pixels. After several frames, the labeled superpixels will almost completely cover the unwanted areas in a dinamyc scene. We call this process background segments tracking. Fig. 3 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 runs, the labeled superpixels flow inside the window, propagating the background mask. The segmentation is then refined applying the grab-cut method ([18][15]) guided by the background labeling, which works a an initialization, replacing the usual user interaction.

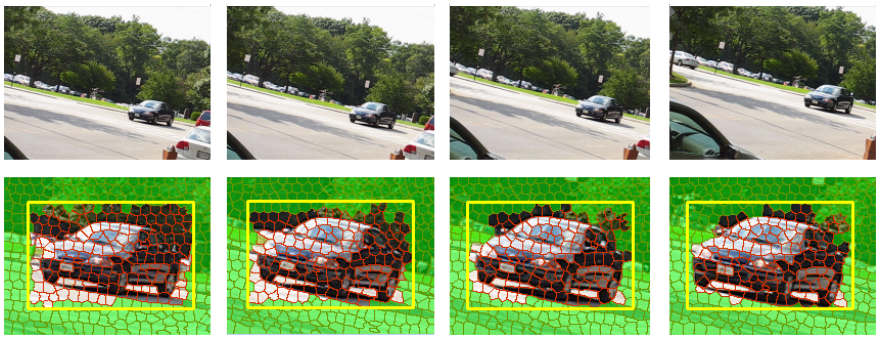

Fig. 3. Background segments automatic labeling and propagation, the flow goes from left to right. The superpixels that are outside the tracker window (yellow) are labelled as background (green) in the first frame. These labels are propagated when the sequence runs.

In order to perform the background segments tracking, we propose a superpixel matching technique, which we call superpixel flow. The objective of the superpixel flow is to find the best match for every superpixel p in the first frame with one (p') in the next frame, while holding a global flow-like behaviour.

Thus, the superpixelization should maintain certain size homogeneity within a single frame. Some super pixel techniques can cope with this requirement [9][10]. For the experiments presented in this work, we prefer the SLIC method [9], which gives good results in terms of size homogeneity and compactness of the superpixelization.

Inspired by a large number of optical flow and stereo techniques [7][12][13], the superpixel flow is modelled with a pairwise Markov Random Field. The matching is performed via maximum-a-posteriori (MAP) inference on the labeling l, which is equivalent to the minimization of an energy function of the form:

$$E(l) = \sum_{p \in \Gamma} D_p(l_p; I_0, I_1) + \sum_{(p,q): q \in \mathcal{N}_r} S_{p,q}(l_p, l_q). \quad (1)$$

With l the set of labels of the superpixels in I_0, that match with those in I_1, in the super pixel space Γ. $\mathcal{N}_r$ is a neighbourhood of radius r of the superpixel p. The terms D, and S in (1) stand for data term and spatial smoothness term. The first one determines how accurate is the labeling in terms of consistency with the measured data (color, shape,etc.). In the classical optical flow equivalent 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 appearance based feature is a low dimensional color histogram (with N bins). So, here, D is the Hellinger distance between the histograms:

$$D_p(l_p; I_0, I_1) = \sqrt{1 - \frac{1}{\sqrt{\bar{\boldsymbol{h}}(p)\bar{\boldsymbol{h}}(p')N^2}} \sum_i \sqrt{\boldsymbol{h}_i(p)\boldsymbol{h}_i(p')}}. \quad (2)$$

Where $\mathbf{h}(p)$ and $\mathbf{h}(p')$ are the color histograms of the superpixel p and its correspondent superpixel in the second frame I_1. For the experiments we used RGB color histogram with N=3 bins per color. Note that the low dimensional histogram gives certain robustness against noise, and slowly changing colors between frames.

On the other hand, the spatial term is a penalty function for spatial diference of the displacement vectors between neighboring superpixels, where a displacement vector has origin in the centroid of the superpixel of the first frame and end in the centroid of the superpixel of the second frame.

$$S_{p,q}(l_p, l_q) = \lambda(p) \sqrt{\frac{|u_{p_c} - u_{q_c}|}{\|p_c - q_c\|} + \frac{|v_{p_c} - v_{q_c}|}{\|p_c - q_c\|}}, \quad (3)$$

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

The operator ρ is the Hellinger distance as used in the data term (2). The histogram distance is nonetheless computed between adjacent superpixels p and q, which belong to the first image. 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 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.

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. The candidate matches are generated by assuming a proximity prior. This means, every possible match should be inside a seach radius in the second frame. Fig. 4 shows matching results for several datasets. Observe that the matches are correct even in difficult cases (bottom right).

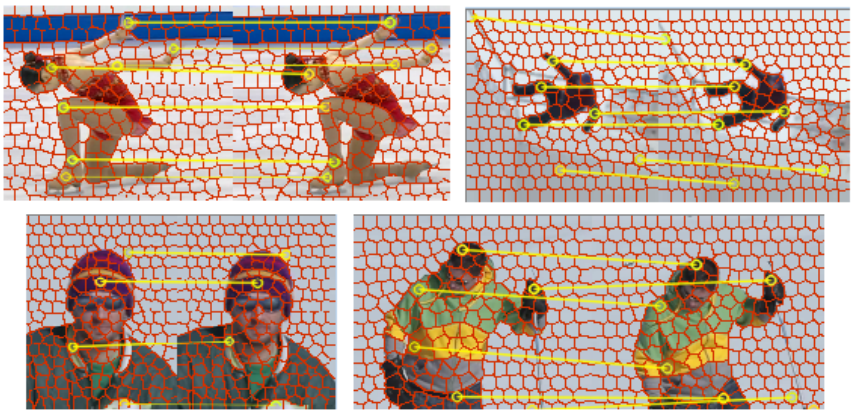

Fig. 4. The yellow lines show selected superpixel matching between pairs of images in several datasets.

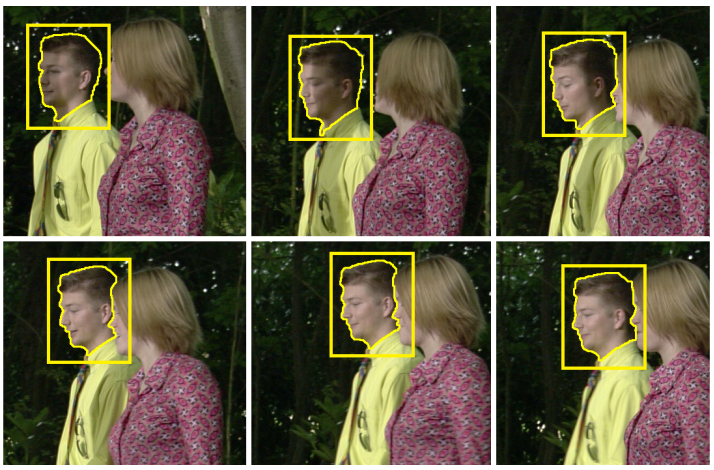

Fig. 5. Segmentation through the sequence Walking Couple (Yellow contour) initialized in the mans head. The yellow box correspond to the tracker output. The labeled background superpixel are not shown for clarity.

Fig. 5 shows segmentation 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 method is tested in the Walking Couple sequence, by allowing only a small amount of iterations in the grab-cut 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, despite the similar (skin) color between foreground and background regions.

In order to understand the effect of including superpixel propagation in a video sequence for object segmentation, some results are shown in the Fig. 6. For these experiments only one iteration is allowed in the graph-cut based methods. The top row frames (Fig. 6) 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.

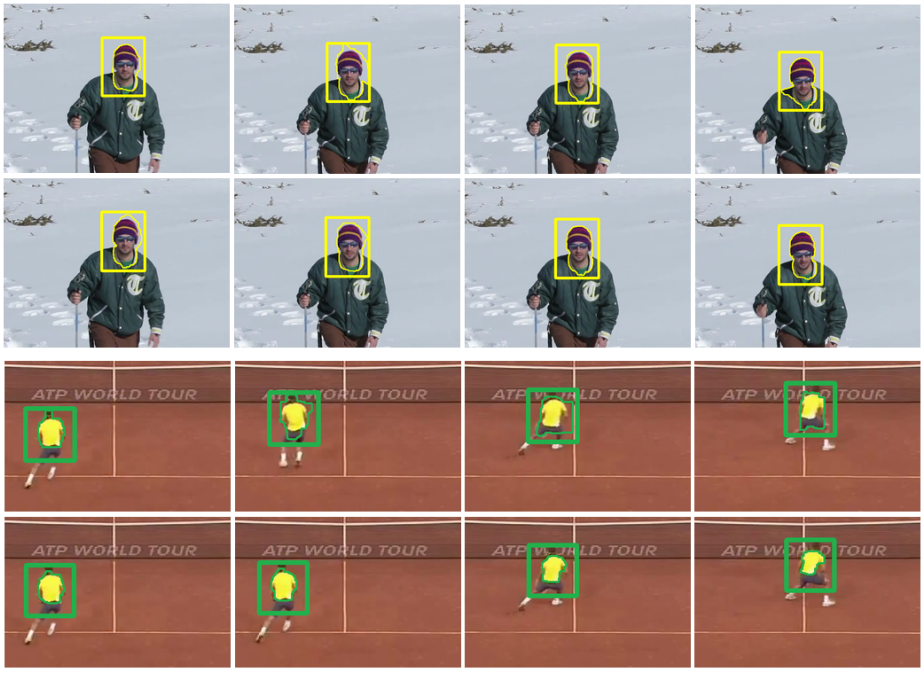

Fig. 6. Face segmentation in the Snow Shoes sequence and T-shirt extraction from Tennis sequence in several frames. For each group, the Top Row: One-iteration grab-cut initialized with tracker window; and the Bottom Row: One-iteration grab-cut initialized with the background regions tracking.

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 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 all of the available datasets. Most of these methods consist in the minimization of an energy function with two terms (As was previously mentioned in this section). The data term is mostly shared between different approaches, but the prior or spatial term is different, and it 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.

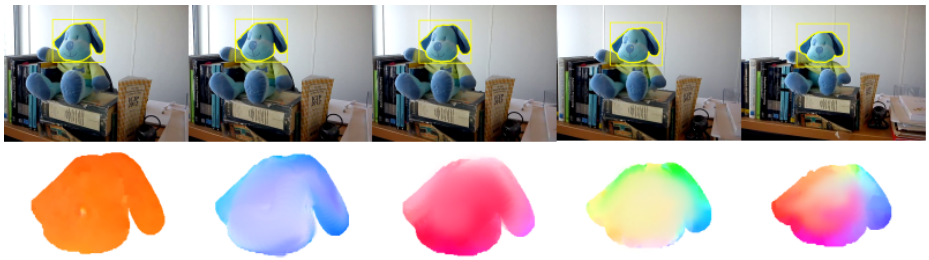

Fig. 7. Object flow with the color code of [17] (bottom) for frames in the Puppy sequence (up).

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 taking into account the segmentation mask within the 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. 7 shows the object flow for several frames 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.

3 Experimental results

Fig. 8. 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.

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. 8 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. 8, two columns correspond to the object flow, and two columns correspond to optical flow, with both types of integration.

The Fig. 9 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.

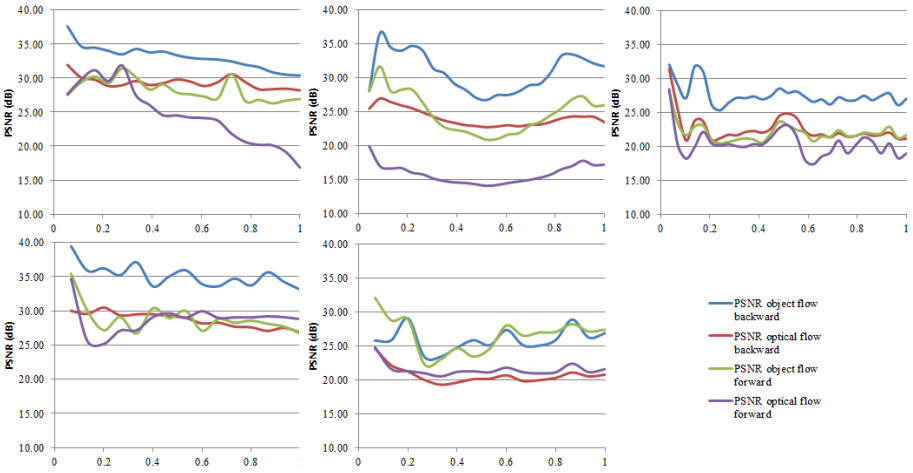

Fig. 9. PSNR graphs for extrapolated images using Object flow and the Simple Optical Flow for 4 sequences. In descendent order: Puppy Seq.; Amelie Retro Seq.; Boy Seq.; Walking Seq.

The Fig. 10 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. 11 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. 10). 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.

Fig. 10. Top: Comparison between extrapolated objects using several methods: Groundtruth object, Object flow, TVL1, Block Matching, Brox, Farneback and Simple Flow. Bottom: First and current frame. The extrapolation is performed using backward accumulation of the flow.

3.1 Object flow for video edition

Patch replacement is a common task in video post-production, and it usually requires am exhausting and time consuming frame-by-frame edition. The time to make such modification can be widely reduced by implementing the object flow, and taking the interest zone as the target object. The edition can be done only in a couple of keyframes, and then, an automatic expansion to the rest of the frames can be performed. Fig. 12 shows the proposed method. The top row contains the automatically edited frames (logo insertion), and the bottom row shows the original frames. Pay attention to the difficult non-rigid transformations.

4 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

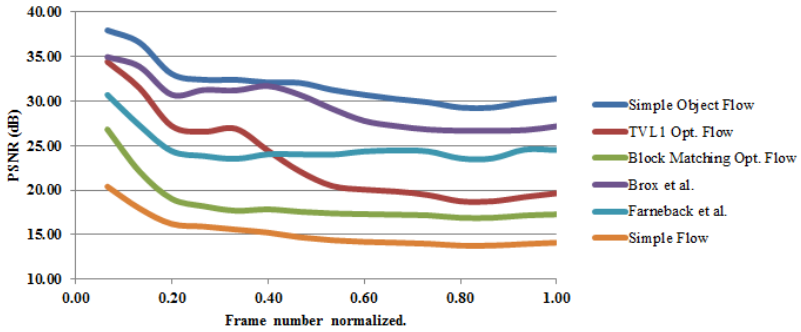

Fig. 11. PSNR graphs for extrapolated images using Object flow and the different Optical Flow techniques for the Amelia sequence.

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.

References

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.

Fig. 12. An application of the object flow for augmented reality/video edition.

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. Brox, A. Bruhn, N. Papenberg, J. Weickert. High accuracy optical flow estimation based on a theory for warping. *European Conference in Computer Vision.* 2004.
23. S. Hare, A. Saffari, and P.H.S. Torr, Struck: Structured Output Tracking with Kernels. *International Conference on Computer Vision.* 2011.
24. B. Babenko, M.H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. *Computer Vision and Pattern Recognition.* 2009.
25. J. Shi and C. Tomasi. Good features to track. *Conference on Computer Vision and Pattern Recognition* , 1994.