Hierarchical Representation of Videos with Spatio-Temporal Fibers

Ratnesh Kumar Guillaume Charpiat Monique Thonnat
INRIA, Sophia-Antipolis, France
firstname.lastname@inria.fr

Abstract

We propose a new representation of videos, as spatio-temporal fibers. These fibers are clusters of trajectories that are meshed spatially in the image domain. They form a hierarchical partition of the video into regions that are coherent in time and space. They can be seen as dense, spatially-organized, long-term optical flow. Their robustness to noise and ambiguities is ensured by taking into account the reliability of each source of information.

As fibers allow users to handle easily moving objects in videos, they prove useful for video editing, as demonstrated in a video inpainting example.

1. Introduction

Owing to the huge amount of video data being recorded daily nowadays, there is an increasing demand for tools which can cater to video processing tasks such as video editing, compression, or understanding. These tasks are fundamentally challenging and require a good video representation algorithm providing reliable **dense** and **long term** correspondences. To be of practical usage, video processing tools should also require as little user interaction as possible, and be computationally fast. In this paper our goal is to develop a video representation algorithm which will facilitate the extraction of long term spatio-temporal statistics from video, and at the same time being efficient in computational requirements.

Optical flow estimation and video segmentation are often regarded as two different tasks in computer vision. While optical flow focuses on point-wise correspondences between frames, video segmentation is the extraction of temporally coherent regions from a video, without point-to-point temporal correspondences. The emerging need for long-term trajectories as well as for more detailed video segmentation is asking for the unification of the two fields, and requesting suitable structures to bridge the gap. The combination of these two kinds of information is needed to analyze finely videos and is a prerequisite to precise video understanding. For instance, relevant information for gesture recognition can be extracted from the identification of body parts, from their motion as well as their shape variations [1].

Current work on video segmentation and optical flow can be broadly classified into two categories : frame-by-frame based approaches and volume-based approaches. Frame-by-frame approaches take as input one or two successive frames of a video, while volume-based approaches consider many successive frames of a video at once. Frame-by-frame approaches have the advantages of low memory requirement while volume-based approaches in recent years have demonstrated good coherency in object labeling over time. Recently with the increase in computers' memory, volume-based approaches have gained importance, as jointly processing all frames of a video brings more information and helps maintaining segmentation or trajectory coherency over time. However the accuracy of current video segmentation algorithms still needs to be improved. Our approach combines cues from multiple frames over time and is based on the representation of videos as 2D+T volumes.

Each moving part of an object carves a different subvolume in the 2D+T cube formed by the video. Approaches [2, 16] aim at extracting meaningful patterns formed onto 2D slices by modeling 3D curves in a video-cube. However they face a high computational cost, and are restricted by assumptions such as linear camera motion [2] or scene consisting of few objects [16], which prohibits these approaches to be applied to current day video data.

In [10], a 3D graph of the full video is built, to perform hierarchical segmentation. Whether nodes of this graph are merged or not depends on a scale parameter. This algorithm performs the best in the evaluation methodology presented by Xu *et al*. [19]. The metric used to merge nodes is based on local color and motion variations and hence there is no incorporation of fine long term pixel correspondences. In [4], video segmentation is based on tracking regions : boundaries are matched within successive frames of a video, using a modified dynamic time warping, to obtain a region segmentation over time. However this relies heavily on low level image processing algorithms to obtain good contours, which prevents it from a practical usage for long

term video analysis in usual situations of temporal lighting variations. Moreover, boundaries of moving objects (*e.g.* humans) in activity or action recognition videos vary significantly with time, making it difficult for a boundary matching algorithm to work consistently throughout the video.

The approaches above, as well as other ones [12], output a set of 3D (2D+T) sub-volumes for a spatio-temporally coherent object (or its sub-parts), and **lack detailed correspondences inside sub-volumes**, required for high-level applications like activity or gesture recognition [1].

Another way of approaching the problem of video segmentation is to build long-term coherent tracks and use them to propagate information to other regions of the video. This is somehow similar to the colorization problem in the graphics community [20], where color information is propagated from user-placed seeds. In order to avoid manual intervention, one can first find informative regions in a video (seeds), and then use these seeds to propagate information to the rest of the video. Point tracks across a video-cube are built this way in [17, 6]. Clustering of these tracks can also be performed [6, 15], leading to a segmentation consisting of sparse point tracks which covers only 3% of the video. On the contrary, we intend to cover full videos with long term correspondences for every pixel. Different approaches [13, 14] incorporate long-term motion into dense discovery of objects in videos.

Similarly, we first seek reliable sources of information in videos, and then propagate to other regions. A key difference of our approach from such work however is the incorporation of statistics over neighborhoods in form of fibers. Furthermore, we present a generic video representation useful for many applications. Another recent work is by [8], wherein the authors first extract superpixels in each frame and use dense trajectories from [6] to obtain affinities for superpixels, thereafter using spectral clustering to segment a video. Our work differs from [8] in following manner. Firstly since optical flow is unreliable at homogeneous locations, we search for initial *fibers* (defined in section 2) near corners. Secondly we build fibers **jointly** in space and time rather than using point trajectories.

With respect to this literature, our contributions are:

- a new point of view on **video representation**, with a structure handling together point trajectories and hierarchical segmentation with object meshing : **fibers**,
- an iterative process to build these fibers, which can be seen as an **optical flow robustifier**, making it **reliably dense and long-term**,
- a new approach to **video segmentation**, with small complexity (quasi-linear) and thus **near real-time**.

We formalize the problem in section 2, build fibers in sections 3, 4, 5, and finally show experiments (section 6).

2. Fibers : Definition and Approach

A video cube is the stack of successive video frames. Fig. 1 shows slices of a video cube, displayed as 3D volumic scans in medical imaging. A pair of *red* colored lines mark the correspondences in the slices. The top-left image displays a standard 2D image frame from the stack, while two other spatio-temporal cuts, in planes (t, y) and (x, t), are shown in the top-right and bottom left. In this 3D representation, points on the static background form straight lines of homogeneous intensity over time, while points on moving objects form curved lines. Analogically to fibers in MRI images of human brains, we term *fibers* these straight or curved lines.We are interested in a dense estimation of fibers, involving all pixels of a video.

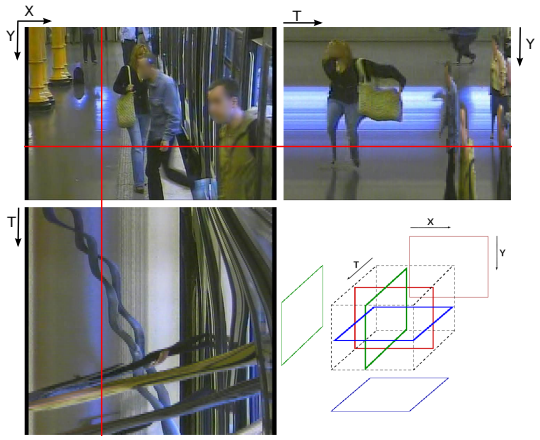

Figure 1. 2D+T video cube represented as 3D data, as in medical imaging: frontal, sagittal and horizontal slices correspond to cuts along planes (x, y), (y, t) and (x, t) respectively. The red lines indicate the values of x and y chosen for the cut. This video shows people exiting a train in a train station. Note in particular the lines in the (x, t) slice, formed by people trajectories, by the train or the background. We name these sets of lines *fibers*.

2.1. Formalization

A **video cube** $V = (I_t)_{t \in [1,n]}$ is a stack of n successive image frames I_t, each of which is defined over a same domain $\Omega \subset \mathbb{R}^2$. It can be seen as 3D data, parameterized by $(x, y, t) \in \Omega \times [1, n]$.

A **fiber** $\mathbf{F} = \left(\{T_i\}_{i\in[1,m]}\ ,\ \mathcal{M}\right)$ is a set of m trajectories T_i, spatially connected with a triangular mesh $\mathcal{M}$. Each trajectory T_i is a sequence of locations $\mathbf{x}_i^t \in \Omega$ during a time span $[t_s^i, t_e^i] \subset \mathbb{N}$, and thus writes $T_i = (\mathbf{x}_i^t)_{t\in[t_s^i,t_e^i]}$. The mesh $\mathcal{M}$ is a planar graph whose vertices are trajectories, and whose edges connect spatially-close trajectories, in a triangulated way.

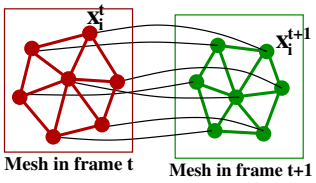

Figure 2. A mesh of trajectories induces a mesh of points in each frame.

A **regular fiber** satisfies moreover that its trajectories (T_i) do not cross each other. More precisely, the trajectory mesh $\mathcal{M}$ induces at any time t, on the set of locations $\{\mathbf{x}_i^t\}_{i\in[1,m]}$, a triangular mesh $\mathcal{M}_t$ over the region occupied by the fiber in image I_t. These meshes $\mathcal{M}_t$ are required to be spatially coherent in time, *i.e.* no triangle should be flipped from a frame to the next one.

Note that fibers may have subpixelic spatial precision. Motions from a frame to the next one are usually not congruent with the pixel grid indeed, and discretizing them spatially would make trajectories suffer from aliasing. Thus the frame domain Ω is considered as continuous.

Our aim is to search for a **partition** of the video cube into a set of regular fibers, optimizing the criteria below.

2.2. Criteria for a good representation

The following three traits were given in [11] to characterize a reliable segmentation:

(C_1) **Region homogeneity.** The segmentation should provide regions that are homogeneous w.r.t. one or more properties, *i.e.* the variation of measurements within the regions should be considerably less than variation at the borders.

(C_2) **Pixelic precision on edges.** The position of the borders should coincide with local maxima, ridges and saddle points of the local gradient measurements.

(C_3) **No oversegmentation.** Areas that perceptually form one region should not be split into several parts.

to which we add:

(C_4) **Time coherency.** The video representation should provide high coherency in time *i.e.* the identities of object should not change while moving across time.

(C_5) **Robustness and reliability.** The representation should not be very sensitive to noise, and its reliability should be estimated, in order to know which parts can be safely trusted and which are debatable.

Criterion (C_1) expresses the homogeneity of each fiber, *e.g.* internal color or motion coherency. Criterion (C_2) stresses that differential information, such as intensity gradients, is useful locally to reach pixelic precision. We will make use of structure tensors and of corner detectors to estimate reliable and precise correspondences. Criterion (C_3) will be dealt with by merging neighboring fibers of similar color or trajectory. Criterion (C_4) asks for regular fibers with long time-spans. Last but not least, criterion (C_5) encourages statistics over neighborhoods instead of considering single local value only, and promotes reliability estimation, which may actually be expressed from such statistics. Fibers are well-designed for this, as their meshed-trajectories structure can be seen as spatial & temporal neighborhoods, upon which statistics can be easily computed. The instantiation of these 5 criteria will be detailed in the next sections.

2.3. Approach Outline

We propose to search for the best partition of the video cube iteratively, by:

1. **suggesting candidate fibers** at locations where motion estimation is reliable (*e.g.* corners), **selecting** the best ones (most coherent in time, longest), while **improving them** by making them regular (no triangle flip) (section 3),

2. **extending them** jointly (to cover the full video domain) in such a way that each video point is assigned a **valid trajectory** (section 4),

3. **merging** similar fibers hierarchically (section 5).

The next sections of the article follow this order.

3. Sparse Reliable Fibers

Fibers can be detected by finding correspondences across the video volume. Many existing techniques, such as optical flow or descriptor matching, can serve this purpose. However all methods are unreliable in homogeneous areas and suffer from the notorious aperture problem on boundaries. Hence we first identify video regions where the correspondences are likely to be reliable (corners), then we check the quality of the trajectories in these areas while simultaneously improving their regularity. Each fiber thus built is then associated with a reliability factor.

3.1. Initiating fibers at corners

Since algorithms computing correspondences are more reliable in high structure variation regions (corners and edges) than in homogeneous areas (*e.g.* Brox & Malik and Werlberger *et al.* flows [5, 18]), we detect corners and build fibers there, using *cornerness* as a reliability factor, defined as :

$$\lambda = \exp\left(\frac{-\gamma}{||\lambda_1 - \lambda_2||_2}\right) \qquad (1)$$

where λ_1 and λ_2 are the eigenvalues of the structure tensor $\nabla I \times \nabla I$ averaged over the area, and γ is a constant. However, corners are often located on the external boundary of objects, in an area thus involving several different overlapping objects with different trajectories. We consequently segment this area into as many fibers as needed. For this, we mesh the area and segment this mesh into submeshes based on color. We apply k-means in color space and segment mesh vertices. To achieve spatial consistency in segmentation we use graph cuts [3] with an MRF prior.

Each of the meshes obtained is now a separate candidate fiber. Candidate trajectories for each mesh vertex are built time step after time step by simultaneously following the optical flow, estimating motion coherency in space and color coherency in time, and correcting the flow if necessary, as described below. Fibers which are too heterogeneous will be split, stopped or removed. We ensure a minimal density of candidate fibers by selecting the best corners in each sufficiently-large hole in the video coverage by fibers. Thus fibers that are stopped let place in the next frame for new candidates.

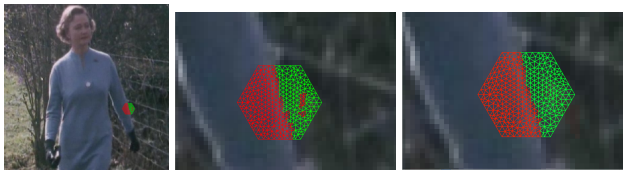

Figure 3. **Left** : Corner and its neighborhood. **Center** : k-means in color space. **Right** : Spatial coherency using graph cuts.

3.2. Color coherency in time

Discontinuity in color at occlusion vicinity are important cues to avoid fibers switching objects. Successive slices of a reliable fiber should have similar color; this can be measured with :

$$\frac{1}{|\mathcal{M}_t|}\int_{\mathcal{M}_t}\left\|\partial_t I_t(\mathbf{x}(t))\right\|^2 d\mathbf{x} \simeq \sum_{i\in\mathcal{M}} w_i \left\|I_t(\mathbf{x}_i^t) - I_{t+1}(\mathbf{x}_i^{t+1})\right\|^2$$

where the integral over the mesh $\mathcal{M}_t$ makes the color variation estimation robust to image noise. Here w_i is a vertex weight standing for the element area (relative area of the neighborhood). The possibility of considering such statistical criteria is an important advantage of fibers, while point tracks alone would suffer from pixelic noise.

3.3. Motion coherency in space

Similarly, motion should be coherent within a mesh at all times. Rather than asking the motion variance within $\mathcal{M}_t$ to be small, which would penalize wider non-rigid fibers, we ask the motion to be continuous, and penalize its spatial variation :

$$\frac{1}{|\mathcal{M}_t|}\int_{\mathcal{M}_t}\left\|\nabla_{\mathbf{x}}\,\mathbf{m}^t(\mathbf{x})\right\|^2 d\mathbf{x} \;\simeq\; \sum_{i\sim j} w_{ij}\left\|\mathbf{m}_i^t - \mathbf{m}_j^t\right\|^2$$

where $\mathbf{m}_i^t = \mathbf{x}_i^{t+1} - \mathbf{x}_i^t$ is the estimated motion at vertex i. Once again, the mesh provided by our fiber representation proves useful to express such kinds of criteria.

3.4. Regularizing the flow

While self-occlusions are frequent at the object level (*e.g.* person walking), trajectories in a same small candidate fiber (*e.g.* the knee) should not intersect each other. This is equivalent to the requirement that no triangle of the mesh $\mathcal{M}_t$ can be flipped in $\mathcal{M}_{t+1}$. If triangles are found to be flipped between frames consecutive frames and that they are not located at the boundary, then we smooth the vertices locations in $\mathcal{M}_{t+1}$ while keeping the mesh boundary constant, until triangles are unflipped. This spatially regularizes the trajectories. If flipped areas include part of the boundary, we compute instead the best affine movement A that sends $\mathcal{M}_t$ as close as possible to $\mathcal{M}_{t+1}$ for the L_2 norm (closed-form solution for $\inf_A \int_{\mathcal{M}} \|A\,\mathbf{x}(t) - \mathbf{x}(t+1)\|^2\, d\mathbf{x}$), and then replace the original motion with it.

3.5. Fiber termination or split

A lack of color or motion coherency of the mesh $\mathcal{M}_t$ at time step t indicates heterogeneity of a fiber. This can be due to an occlusion, or to a drift of the flow across the boundary of an object. Such drift and discontinuity problems have been reported by [17] while concatenating optical flow vectors for long sequences. The advantage of our approach is that, as we estimate fiber color and flow statistics at each time step, we detect these potential issues.

When a lack of coherency is detected, we split the fiber into two homogeneous parts (typically, the two sides of a boundary) if possible, and pursue the work for each of the two fibers independently, provided their spatial size is significant enough (if not, they are just ignored). Note that in this case the fibers are fully split, on their full time span, which is possible as we know their trajectories back in time. This is useful when different objects of similar color or motion behave coherently during a while before becoming distinguishable, as then by propagating the information back in time, we are able to distinguish them at all times.

To cover also drifts that happen too slowly to be noticeable between two successive frames, we add another criterion: the elasticity of the mesh. A too big variation of edge length between any two mesh vertices at any two times (not necessarily consecutive) will call the fiber splitter.

A fiber is stopped if it cannot be split into temporally coherent sub-fibers (*e.g.* full occlusion). It is furthermore

deleted if its time-span thus obtained is too short, as the duration of trajectories is a good indicator of their quality.

With the employment of the above steps for all corners in a video, we obtain a set of reliable fibers at high structure variation regions of a video. The full representation of video in terms of fibers will now need the extension of these quality fibers to the rest of the video.

4. Full video coverage

This section explains how we extend the fibers previously found to the rest of the video. For this, we first find zones for possible fiber extensions, and then rely on trajectory coherency to choose among extension possibilities.

4.1. Geodesics between fibers and rest of the video

The reliable fibers found so far do not cover all pixels of the video. We would like to extend them to the full video by associating to each pixel one of the closest fibers in term of color and motion similarity. This is done by using Dijkstra's algorithm on the graph of all pixels of the video, with multiple sources (the fibers). The local cost considered between adjacent pixels p and q is :

$$\exp\left(\frac{-\alpha}{\|I(p)-I(q)\|_2}\right)+\lambda_{p,q}\exp\left(\frac{-\beta}{\|\mathbf{m}(p)-\mathbf{m}(q)\|_2}\right)$$

where I and $\mathbf{m}$ denote the local color and optical flow, and where α, $\beta \in \mathbb{R}^+$ are constants set to the desirable standard deviation of color and motion allowed for mesh segmentation (*c.f.* 3.1). The cornerness λ (defined in Eq. (1)) expresses the local trust on the optical flow.

The initial distance of the sources is set to be the opposite of their reliability, $-\lambda$, in order to facilitate the extension of reliable fibers. During the shortest path computations, we keep track of the original sources so that we know to which fiber each pixel is the closest, and even to which pixel of that fiber. We assign to each pixel p a trajectory T_p^F by copying the one of the closest pixel of the closest fiber F. Thus we obtain a coverage of the video with possible extension zones for each fiber, with associated trajectories.

4.2. Enforcing trajectory coherency

We would like now to assess the quality of the proposed extensions and trajectories, and ensure their coherency. For each fiber F, we consider a representative reference frame, chosen in the middle of its time span. We project the 2D+T possible extension zone of this fiber on this 2D reference frame by following the previously assigned trajectories, as in Figure 4. The quality of a trajectory T_p^F is expressed as its color coherency, more precisely as the amount of work needed to re-arrange its color histogram H_p into a single Dirac peak. The infinimum of the Earth Mover Distance over all possible color Dirac peaks:

$$\mathrm{coh(p)} := \min_i \; EMD(H_p\,,\,\delta_i) \qquad (2)$$

as well as its $argmin$, denoted by $col(p)$, can be computed in linear time. In order to ensure not only trajectory quality

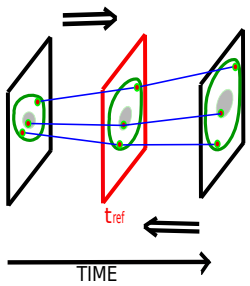

Figure 4. The possible extension zone (in green) of a fiber (in filled gray) is projected on a reference frame (in red), following trajectories (in blue) associated coherently with the fiber.

but also spatial and color coherency within one fiber, we recompute geodesic distances in the reference frame starting from the original fiber, with a different metric, where the cost of moving from a pixel p to an adjacent one q is

$$\exp\Big(\frac{-\alpha_1}{\mathrm{coh(q)}}\Big)+\exp\Big(\frac{-\alpha_2}{||\mathrm{col(p)}-\mathrm{col(q)}||_2}\Big) \qquad (3)$$

where α_1 and $\alpha_2 \in \mathbb{R}^+$ are constants. We thus obtain a geodesic distance map G_f in each reference frame f.

The cost, for any pixel p of the video, of choosing a fiber f, is then defined as the corresponding geodesic distance $G_f(\mathrm{proj}_f(p))$, i.e. as the value of this map G_f at the projection $\mathrm{proj}_f(p)$ of the pixel on the reference frame of that fiber (according to its associated trajectory T_p^F). Among possible fiber extensions, each pixel then chooses the one with the lowest cost.

The total time complexity of the fiber extension stage is **quasilinear**: $O(V \log \Omega)$, where Ω stands for an image frame size. In particular, for a given frame size, **the computation time increases only linearly with the duration of the video**.

5. Hierarchical Representation

At this step we have a very fine representation of the video in term of fibers. This fine representation can be used by algorithms requiring reliable dense long term optical flow *e.g.* action recognition or video compression. Often in many computer vision applications a coarser representation of video is required, *e.g.* for background subtraction or activity recognition. This calls for the need of criteria to merge fibers. Recent works on object segmentation in videos from sparse set of input trajectories [6, 7] use graph spectral clustering to obtain a fixed number of labels relating to number of objects in the scene. Note here that as

opposed to these algorithms we have a trajectory associated to all pixels of the video, and we describe below a simple procedure to obtain a hierarchical representation of a full video in terms of fibers.

In order to obtain hierarchical representation, we merge fibers based on their speed similarity. Alternatively a different cost function could be used, *e.g.* as in [6] incorporating both spatial and speed distances between the trajectories. With the cost function (4), one can expect to hierarchically merge two fibers without considering their spatial positions.

We compute barycentric trajectories for fibers and compare them using (4), where $O[F_i, F_j]$ defines the overlap time span for two fibers F_i and F_j, and $\mathbf{m}^t_{G,i}$ represents the barycentric motion of fiber i at time t:

$$d(F_i, F_j) \propto \frac{1}{O[F_i, F_j]} \sum_{t \,\in\, O[F_i,F_j]} \left\|\mathbf{m}^t_{G,i} - \mathbf{m}^t_{G,j}\right\|^2. \quad (4)$$

We consider the graph of all fibers, initially fully disconnected, where fibers are represented by their barycenter trajectories, and set an initial threshold τ. At each hierarchy level, we connect the fiber nodes of the graph for which the cost (4) is less than an associated threshold τ. The hierarchy level is changed by multiplying τ with a constant scale factor s (not depending on experiments). This approach builds a hierarchy tree, ensuring that finer motion details are preserved at the lowest hierarchy while much coarser motion segmentation is at the higher hierarchy. Our results demonstrate the robustness of the incorporation of motion detail by fibers, as the background merging is almost perfect in these challenging moving camera scenarios.

6. Experiments and Complexity

Dense groundtruth annotation of video datasets is rarely available as it is an enormous task. Since we analyze long term motion and color coherency, we considered the datasets containing long sequences of around 100 frames provided by the authors of [6, 10, 17]. The dataset proposed by [10] consists of videos from Hollywood, and has a few long sequences. Its downside is that it is highly processed with artificial effects in some frames, and the scene changes are quite rapid compared to natural motion, with frames skipped unregularly, making it impractical for long term coherency check. Moreover this dataset and the one proposed by [17] do not have any groundtruth annotations. One of the datasets used by [19] (`xiph.org`) consists of groundtruth annotations. However this dataset doesn't reflect the current day video usage, as the frame resolution is mere 240x160 pixels, and as most video pixels remain static from the first frame to last one.

Thus, finally, we consider the videos from [6, 17]. We present our video segmentation, with spatio-temporal slices, to *display* the label coherency in time, in Fig. 5, and more classically with frames in Fig 6 and 8. Darker regions show higher costs of association of the corresponding pixels to the fibers, *i.e.* lower reliability. In Fig. 7 we show the result, on the same video as in Fig. 5, obtained by the state-of-the-art in video segmentation [10], using also optical flow, but not trajectories. This shows that motion is indeed vital for segmentation and that local optical flow does not help sufficiently, hence the interest of our approach. On the other side, motion segmentation algorithms do usually not provide a full dense segmentation, but we do, with trajectories for all pixels.

All results are obtained with the same parameter values. The total computational times including flow computation for typical videos from [6] ranges from 70∼140 seconds for 20 frames, which is **very fast** compared to usual approaches and can still be easily improved [1]. We use the GPU-based optical flow [18] on a basic Nvidia graphics card.

We now show the usefulness of fibers in practical usage such as video editing. The hierarchical representation of fibers allows the selection of moving objects in videos very efficiently. In Figure 9 we perform a video inpainting task, for which video zones to remove or to keep are selected in only **very few clicks**. This is to be compared with the state of the art [9] which requires manual segmentation of all frames.

Figure 9. Inpainting task. Left : original video (top) and xt slice (bottom) showing trajectories. Right : our result (no artifact!). Clusters of fibers were computed and selected with only 7 mouse clicks to distinguish the disturbing girl from the reporter and background. The girl was removed and the hole was filled by extending the background fibers in time.

7. Discussion and Conclusion

We presented a novel representation of videos in terms of **fibers**, practical to handle jointly temporal aspects (such as motions and trajectories) and spatial aspects (such as meshes and segmentation into regions). We build these fibers in **quasi-linear complexity** $O(V \log(\Omega))$, which makes them affordable in practice for real applications.

To the contrary of other approaches, we do not rely on a perfect optical flow, but robustify it instead, by checking

[1] Refer to `http://www-sop.inria.fr/stars/Documents/fibers/` for more details and examples on fibers.

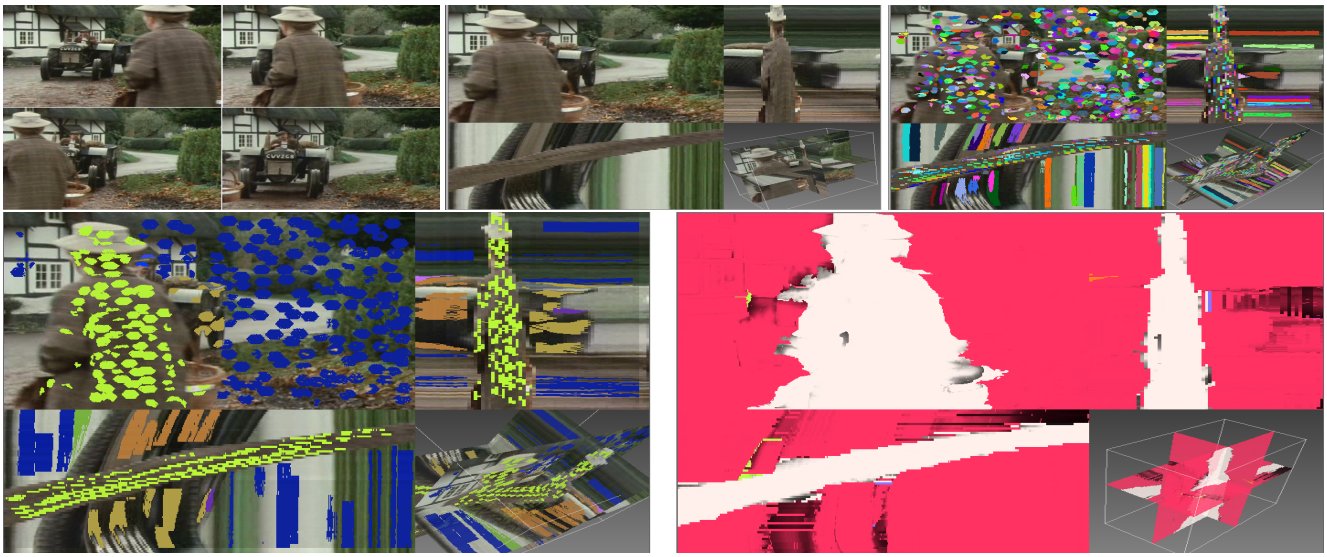

Figure 5. **Top Row**: Left image displays four images of a sequence from [6]. Center image displays the same video as a 3D volume as in Fig. 1, with an additional bottom right corner showing the 3D position of the xt and yt slices. Rightmost image displays all fibers found, in different colors. **Bottom Row** : Left image displays a high level of their hierarchical clustering. Right Image displays the highest level of the hierarchical clustering, with fiber extension. This result compares favorably to the state-of-the-art of video segmentation in Fig 7.

Figure 6. **First Row** : Sample Input image frames (12,24,46,53,66) from marple13 sequence by [6]. **Bottom Row** : After 5 steps of hierarchical merging. Parts of the foreground and background are indistinguishable during the first frames of the video (same color). Yet, the two objects follow later significantly different trajectories, which enables us, when propagating this information back in time, to separate them as different fibers in all frames (*c.f.* section 3.1).

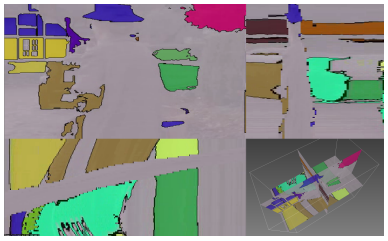

Figure 7. Result on the same sequence as Fig. 5 obtained by the state-of-the-art segmentation of videos [10], using also optical flow, for comparison. The main foreground object and the background are already merged at a relatively low hierarchical level.

the coherency of trajectories, and modifying them at different levels (mesh unflipping and fiber extension). The advantage of our approach over classical segmentations is that **we associate not only a label to each pixel, but also a coherent trajectory**. Thus the segmentation is more robust to noise. Moreover, we provide, in plus of the segmentation, a reliability map of our result.

Meshes of trajectories prove useful in many places. First, they allow to define vertex-dependent quantities, such as motion or depth, while ensuring the continuity of their variation. Furthermore, they provide a dense, organized video coverage, to the contrary of most approaches offering only sparse tracks, short tracks, or frame-by-frame estimations. The range of possible criteria to optimize with our representation is much greater, as it allows us to express statistics, both in time and in space, making estimated quantities more robust to video noise.

Another strength of this framework is the incorporation of **hierarchical clustering, with meshes**. For action recognition applications it is often desired to keep the finest representation in term of fibers (long term dense optical flow) while for domains like background segmentation or foreground estimation (in freely moving cameras), a much coarser representation can be selected. This proved very useful in the video editing task. Fibers are a middle-level entity bringing the gap between low-level pixels and high-level activity recognition: usual practical problems in computer vision, like lighting variations, shadows or occlusions, are difficult to face at the pixel level, and require more semantic information from the scene. Shadows or occlusions will result in several bits of homogeneous fibers, that can easily be merged later at a higher level, based on global tra-

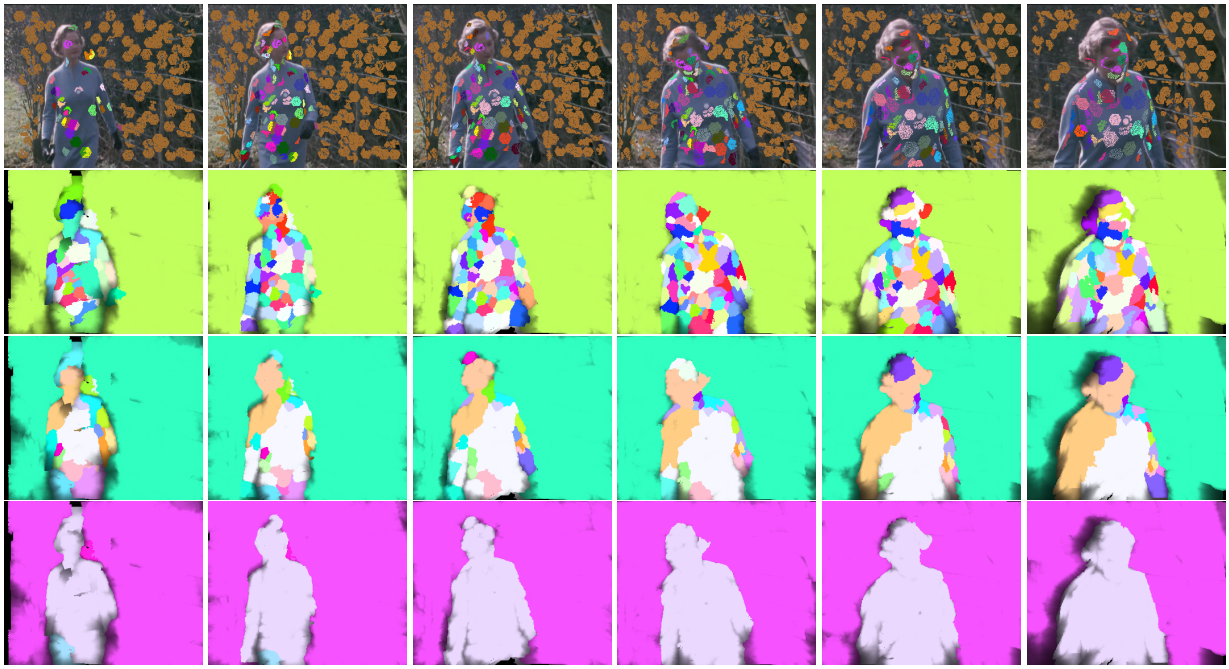

Figure 8. Example from the *marple3* sequence by [6], showing frames at a spacing of 10 frames till the 50th frame. **First Row** : Fiber merging at a lower hierarchy. Notice that the background belongs to one cluster now. **Second Row** : Merged Fibers at a lower hierarchy. **Third Row** : Penultimate hierarchy in merging fibers. **Fourth Row** : Final hierarchy in merging fibers.

jectory similarity or rules (*e.g*. a darker fiber at the bottom of a foreground object and following it is a shadow). We plan to implement this as well as to use fibers for activity recognition in long videos.

Acknowledgements

The research leading to these results has received funding from the European Communitys Seventh Framework Programme FP7/2007-2013 - under grant agreement nº 248907-VANAHEIM.

References

[1] J. K. Aggarwal and M. S. Ryoo. Human Activity Analysis : A Review. *ACM Computing Surveys '11*.

[2] N. Apostoloff and A. Fitzgibbon. Automatic video segmentation using spatiotemporal T-junctions. BMVC '06.

[3] Y. Boykov and G. Funka-Lea. Graph Cuts and Efficient N-D Image Segmentation. *IJCV '06*.

[4] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. ICCV'09.

[5] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *TPAMI' 11*.

[6] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. ECCV '10.

[7] K. Fragkiadaki and J. Shi. Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. CVPR '11.

[8] F. Galasso, R. Cipolla, and B. Schiele. Video Segmentation with Superpixels. ACCV '12.

[9] M. Granados, K. I. Kim, J. Tompkin, J. Kautz, and C. Theobalt. Background inpainting for videos with dynamic objects and a free-moving camera. ECCV'12.

[10] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. CVPR '10.

[11] U. Köthe. Primary image segmentation. DAGM '95.

[12] Y. J. Lee, J. Kim, and K. Grauman. Key-Segments for Video Object Segmentation. ICCV'11.

[13] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the Future: Spatio-temporal Video Segmentation with Long-range Motion Cues. CVPR '11.

[14] P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. ICCV '11.

[15] A. Ravichandran, C. Wang, M. Raptis, and S. Soatto. Super-Floxels: A Mid-level Representation for Video Sequences. ECCV Workshops '12.

[16] M. Ristivojevic. *Space-time image sequence analysis: object tunnels and occlusion volumes*. PhD thesis, ' 06.

[17] P. Sand and S. Teller. Particle Video: Long-Range Motion Estimation Using Point Trajectories. *IJCV '08*.

[18] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 Optical Flow. *BMVC '09*.

[19] C. Xu and J. J. Corso. Evaluation of super-voxel methods for early video processing. CVPR '12.

[20] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *TIP '06*.