MOTION SEGMENTATION IN VIDEOS FROM TIME OF FLIGHT CAMERAS

Sajid Ghuffar[1]*, *Nicole Brosch*[2]†, *Norbert Pfeifer*[1], *Margrit Gelautz*[2]

[1] Institute of Photogrammetry and Remote Sensing
[2] Institute of Software Technology and Interactive Systems
Vienna University of Technology, Austria
{sajid.ghuffar,nicole.brosch,norbert.pfeifer,margrit.gelautz}@tuwien.ac.at

ABSTRACT

This paper investigates motion estimation and segmentation of independently moving objects in video sequences from a range camera. Specifically, we present a motion estimation algorithm which is based on fusion of range flow and optical flow constraint equations. The flow fields are used to derive long-term point trajectories. A segmentation technique groups the trajectories according to their motion and depth similarity into spatio-temporal objects. We show results on a real world scene captured with a time of flight camera.

Index Terms— range flow, segmentation, ToF camera

1. INTRODUCTION

In this paper we analyze short- and long-term motion in video sequences recorded by a Time of Flight (ToF) camera. We estimate motion between two frames of a video by integrating range flow and optical flow. We use this motion information to derive dense point trajectories and group them into spatio-temporal segments. The derived trajectories can provide vital cues for higher-level scene interpretation. Furthermore, this information can be used for image registration by evaluating motion consistency among different segments.

ToF cameras provide direct range (depth) and synchronized intensity information in real-time. However, state-of-the-art ToF cameras have limited spatial resolution and low precision. Moreover, measurements from range cameras suffer from distortions caused by factors like object reflectivity, object distances [1] and internal light scattering [2]. Consequently, there is a need to study motion estimation specifically in the context of ToF camera measurements.

1.1. State of the art

Analogously to optical flow, which describes the motion between two intensity images, range flow [3] refers to motion vectors derived from a sequence of range images. Initial work on the estimation of range flow over deformable surfaces can be found in [4]. Spies et al. [3] present solutions for range flow in cases with different types of 3D structure. Barron and Spies [5] integrate range flow and optical flow in a least squares framework. Using range and intensity information can result in denser flow fields. However, it is not guaranteed that flow vectors are computed for each pixel. To solve this problem, [3] use an iterative regularization algorithm for computing dense flow fields. Schmidt et al. [6] incorporate the sensor observations from a ToF camera directly into the range flow constraint equation using a pinhole camera model.

Common movement and spatial proximity are strong cues for grouping [7]. Such knowledge has been incorporated differently in various segmentation approaches. These approaches can be divided into two groups, according to the properties of their results. Approaches which result in a dense segmentation are typically based on local motion cues (e.g., optical flow [8]). However, as the motion of two objects can be locally similar, these methods can benefit from additional information such as long-term motion cues [9], color [8] or depth. A second class of segmentation approaches takes on a more global view of the problem. They use tracking to derive sparse point trajectories, which are grouped according to global motion similarity (e.g., [9]). However, as the trajectories are sparse, so too is the result of the segmentation. In order to assign the remaining points to segments, additional information (e.g., color [9]) must be incorporated. Accordingly, various segmentation approaches [9, 10] take advantage of denser point trajectories [11, 12]. These trajectories are accurate, but do not guarantee that each pixel will be assigned to a trajectory (or, subsequently, a segment).

In this paper we aim to group all points of a range video into segments of coherent motion. In addition to depth (obtained by a ToF camera), we incorporate long-term motion information. Similarly to [12], we derive trajectories from previously computed motion vectors (Section 2.1). In contrast to the previous approaches, we attempt to derive a trajectory for each pixel of the range video (Section 2.2). In Section 2.3, we use an efficient graph-based segmentation technique [13, 8] to

*This work was supported by the Doctoral College on Computation Perception at TU Vienna.

†This work was funded by the Doctoral College on Computational Perception at TU Vienna and the Austrian Science Fund (FWF): P19797-N13.

group trajectories according to their spatial and motion similarity. Section 3 demonstrates the capabilities of our algorithm on a video recorded in our lab and compares the results to a state-of-the-art segmentation method [8]. The results obtained in the different processing steps are illustrated and discussed.

2. ALGORITHM DESCRIPTION

In our work we use an SR3000 ToF camera[1]. It computes range by emitting an amplitude-modulated continuous wave signal and evaluating the round-trip time by measuring the phase difference between the emitted and the received signal. In addition to range values, the SR3000 delivers the amplitude of the returned signal. The raw amplitude images are affected by the inverse-square law and illumination fall-off. Since the optical flow constraint equation is based on the brightness constancy assumption, we correct the raw amplitude images for these effects. The corrected images are referred to as *intensity images*. Our range flow computation (Section 2.1), which incorporates optical flow, is based on intensity and depth. The trajectory generation step (Section 2.2) takes the previously estimated flow fields and the intensities as input. The segmentation algorithm groups trajectories according to depth and motion similarity (Section 2.3). Below, we discuss these steps in detail.

2.1. Range Flow

Depth can be viewed as a function of space (X, Y) and time (T) $Z = f(X, Y, T)$. Taking a total derivative with respect to time gives the range flow constraint equation: [3]

$$Z_X U + Z_Y V - W + Z_T = 0, \tag{1}$$

where U, V and W are flow velocities. Z_X, Z_Y and Z_T are the spatial and temporal derivatives of range. Analogously to range flow, the optical flow constraint equation is given by:

$$I_X U + I_Y V + I_T = 0, \tag{2}$$

where I_X, I_Y and I_T are the spatial and temporal derivatives of intensity. Specifically, we compute the derivatives of range and intensity images as described in [5]. Since range and intensity images from the SR3000 depend on the scene, one or the other might be dominant in a least squares solution. We weight range and intensity equally using the average range and intensity gradient magnitudes [5]. When applying the constraints mentioned above in a $n \times n$ neighborhood, we obtain $2 \times n^2$ equations of the form $Ax = 0$. Their least squares solution is given by:

$$A^T A x = \lambda x, \tag{3}$$

where $\mathbf{x} = \begin{bmatrix} U & V & W & 1 \end{bmatrix}^T$ is an eigenvector and λ is an eigenvalue of $A^T A$. To compute these three unknowns, three constraints are needed. For areas in which we have enough constraints, the lowest eigenvalue of the matrix $A^T A$ gives the quality of fit and the full flow can be computed. When using only depth information, linear structures will result in two constraints and planar structures will result in a single constraint. For both cases, appropriate flow vectors are computed using eigenvectors of non-vanishing eigenvalues [3].

At pixel neighborhoods with multiple motions, range flow cannot be represented by a single value. As a result, the smallest eigenvalue will not be close to zero. This is also the case in the presence of high noise. Hence it is essential to define a measure which identifies these areas. We compute confidence values for each flow vector as described in [3] and set a threshold τ_1. Whenever the lowest eigenvalue of $A^T A$ exceeds τ_1, we set the pixels' confidence value and its corresponding flow vector to zero. Additionally, we compute only flow vectors where the trace of $A^T A$ exceeds a second threshold τ_2 and exclude areas with an insufficient gradient [3].

When using range and intensity as well as a local neighborhood, it is not guaranteed that flow can be computed at each pixel. For the purpose of motion segmentation, however, dense flow fields are essential. Hence we apply the regularization procedure described in [3]. As a contribution over [3], we use weights and masking to avoid smoothing across depth discontinuities. Both are computed from depth similarity.

2.2. Flow Trajectories

It has been shown that long-term motion analysis can be beneficial for segmentation [9, 10]. In contrast to local motion, such as range flow, trajectories take the entire movement of a point into account and are able to disambiguate locally similar motion (see Fig. 1). We obtain dense point trajectories by applying a flow vector-based tracker similar to [12]. Contrary to [12], we do not remove points which are difficult to track, but attempt to cover the entire video.

Starting with the first frame, we build trajectories by iteratively following the motion vector v_{it} of each pixel x_{it} to its new position in the next frame $x_{it+1} = v_{it} + x_{it}$. A trajectory ends when a pixel without valid motion information is reached (leaves scene, gets occluded). We detect occlusions by flow divergence [11] and a tolerant consistency check of forward flow and backward flow [12]. Since these checks do not account for mismatches or occluding objects with similar movement as the background, we additionally observe a point's intensity change over time [11]. In order to assign each pixel one trajectory, we start a new trajectory whenever a pixel is not assigned to a trajectory which comes from a previous frame (e.g., due to disocclusion, see Fig. 1).

[1] http://www.mesa-imaging.ch

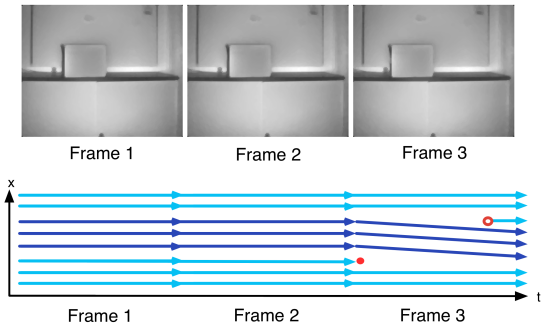

Fig. 1. Intensity images (*top*) and horizontal movement (*bottom*). Locally similar motion (*Frame 1, 2*) can be disambiguated by long-term motion (*Frame 1-3*). Trajectories stop at occlusions (*dot*) and are initialized at disocclusions (*circle*).

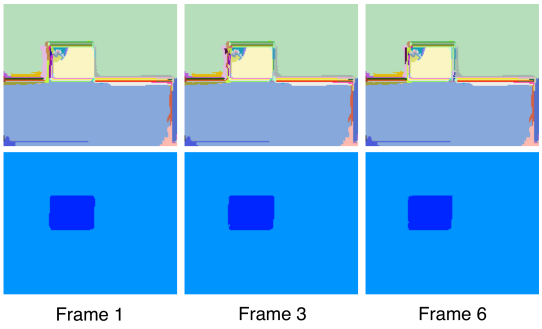

Fig. 2. Segmentation result for three example frames of a video sequence. Over-segmentation of point trajectories according to motion and depth similarity (*top row*) and corresponding frames after iterative region merging (*bottom row*).

2.3. Motion Segmentation

Having applied the trajectory generation step, every pixel belongs to a respective single trajectory and all pixels on a trajectory belong to the same object. Our goal is to group these trajectories to spatio-temporal segments that correspond to objects or groups of objects which move together. To this end, we define a pairwise similarity measure w_{ij} which takes common movement (*law of common fate* [7]) and spatial proximity (*law of proximity* [7]) into account:

$$w_{ij} = max(d_{ij})^2 \sum_{i \cap j} \frac{||x_{it} - x_{jt}||^2 (||v_{it} - v_{jt}|| + 1)^2}{|i \cap j|}. \quad (4)$$

Here, two spatially neighboring trajectories i and j are compared in the frames in which both trajectories exist, i.e., $i \cap j$. $|i \cap j|$ is the number of the trajectories' common frames. The product $max(d_{ij})^2$, the square of the maximal velocity multiplied by the square of the maximal spatial distance, is weighted by the average distance between time-corresponding spatial positions x_t of i and j and their velocity values v_t. As a result, only spatially proximal trajectories with similar movement in every common frame can cause small values. Note that the described measure is similar to the measures used in [9, 10]. However, our spatial differences include a depth component. Contrary to [9, 10], we increase the velocity difference by one. Consequently, we do not neglect the spatial distance of points with the same velocity.

Having defined a similarity measure, the next step is to group trajectories accordingly. For this purpose, we use an efficient graph-based segmentation technique proposed in [13, 8]. We apply the segmentation algorithm to trajectories which consist of more than one pixel and therefore have valid motion information. Then, we merge short (one pixel) trajectories and groups of trajectories that are smaller than a given minimum segment size [8]. The process described above results in an over-segmentation (see Fig. 2, *top*).

As [8] point out, applying the segmentation algorithm mentioned above hierarchically leads to more robust results. Thus we iteratively re-segment the result from the previous merging iteration. For that purpose we define new edge weights wr_{ij} which express the similarity of two groups of trajectories i and j in their common time window. The weights are based on the χ^2 distance of per-frame-flow-histograms [8] $d_v \in [0, 1]$ and the Euclidean distance of the average depth per frame $d_d \in [0, 1]$:

$$wr = (1 - (1 - d_v)(1 - d_d))^2. \quad (5)$$

The segmentation procedure described in [13] is applied iteratively [8] until the desired level of generalization is reached (see Fig. 2, *bottom*).

3. EXPERIMENTAL RESULTS

To the best of our knowledge, there is no publicly available ToF dataset with motion and segmentation ground truth. Therefore, we present a qualitative analysis and comparison of our results on a video recorded by us using an SR3000 ToF camera. Fig. 3 shows intensities (*first row*), depth (*second row*) and estimated flow vectors (*third row*) for nine frames of the video. It can be seen that the persons' movements are well determined by our algorithm. However, when both range and intensity gradients diminish and the signal to noise ratio is low, motion cannot be captured. This is visible in the lower portion of the body as shown in Fig. 4. Fig. 4 shows the derivatives of range and intensity images in horizontal direction and further demonstrates the advantage of fusing depth and intensity information. The intensity gradient in the lower portion of the person in the front diminishes. The depth gradient diminishes towards the lower portion of the person in the background. Therefore, when using only range flow or optical flow, movement in these areas is not captured due to low determinability of flow vectors (see Fig. 5, *a.*, *b.*). In these areas, fusion of depth and intensity increases the

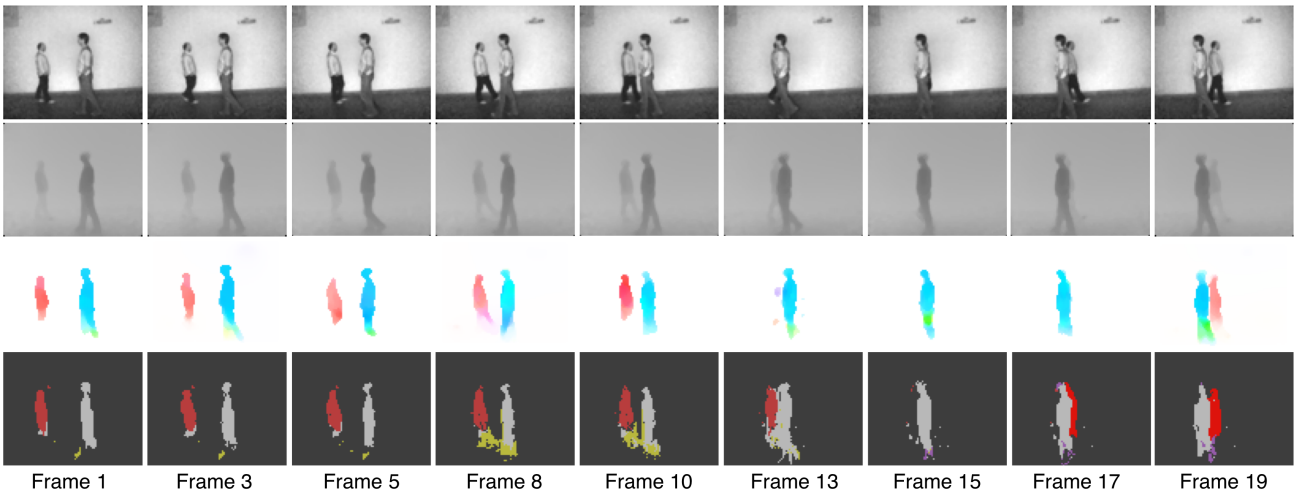

Fig. 3. Frames of a range video (intensity *first row*, depth *second row*, bright: front, dark: back), corresponding range flow fields (*third row*, hue encodes orientation and saturation magnitude) and segmentation result (*forth row*, color coded labels).

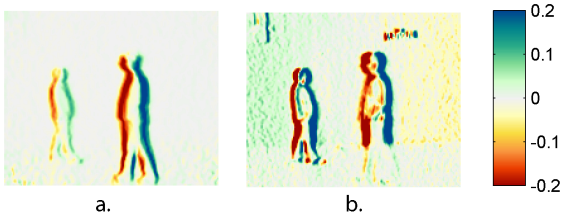

Fig. 4. Derivatives in horizontal direction in *a.* depth image (in meters) and *b.* intensity image. Mean of intensity derivatives has been scaled to match the mean of depth derivatives.

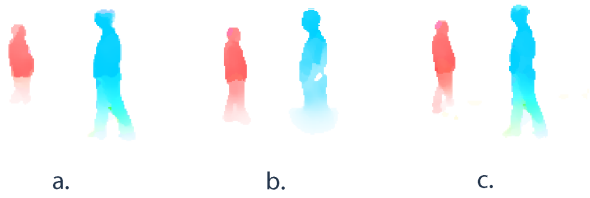

Fig. 5. Flow vectors computed from *a.* depth images, *b.* intensity images and *c.* both, depth images and intensity images. Hue encodes orientation and saturation magnitude.

determinability of flow vectors and results in more reliable flow vectors (see Fig. 5, *c.*).

Our results indicate absence of noise in the static areas of flow fields (see Fig. 3). The main reason for that are two thresholds. More precisely, flow vectors are only computed for areas with a sufficient gradient (see Fig. 6, *a.*). Threshold τ_2 excludes areas with dominant noise that might otherwise result in incorrect flow vectors. Fig. 6, *b.* gives an example for selecting a low τ_2, which results in an inaccurate flow field. Another important factor is the use of a confidence value based on threshold τ_1, which determines the quality of the least squares solution. It detects and excludes multiple motions and occlusions that can lead to inaccuracies in the flow vectors. Furthermore, applying a median filter to the flow fields before the regularization step improves the results. We additionally noticed improvements in the results when using an iterative weighted least squares instead of the ordinary least squares solution (Eq. (3)). Iteratively re-weighting observations based on residuals reduces the influence of outliers. This solution adds robustness to the motion estimation and results in smoother flow vectors.

Fig. 2 and Fig. 3 show that the obtained segmentations are temporally coherent. The objects follow the previously estimated flow vectors (e.g. Fig. 3, *third row*). In fact, the derived trajectories as well as the segmentation strongly depend on the flow vectors and the occlusion detection. Fig. 3 gives an example for the latter case. In frame 15, occlusions that are not detected correctly cause improper trajectories, which are visible as isolated points near the person (light gray).

As an additional experiment and comparison, Fig. 7 shows a result which was obtained by applying a state-of-the-art segmentation method [8] to our test video (see Fig. 3). This segmentation technique was originally developed to segment conventional videos. It is based on local color and local motion similarity and results in a dense segmentation. In this experiment, we use [8] to segment depth (see Fig. 7, *a.*) and intensity (see Fig. 7, *b.*). It can be seen (Fig. 7) that distortions in the ToF measurements affect the segmentation results. This is especially visible in the intensity segmentation, in which pixels are wrongly grouped due to noise. This effect is not visible in our segmentation result (see Fig. 3). The main reason for that is a conceptual difference of our

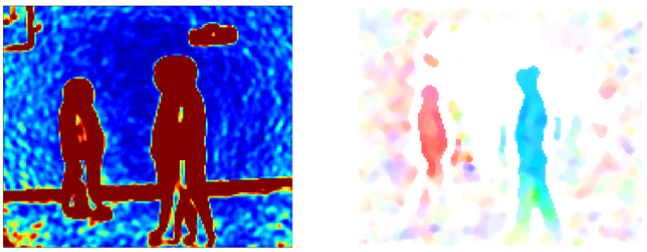

Fig. 6. *a.* Trace of $A^T A$. Cold to hot colors indicate an increasing trace. *b.* Flow vectors using a low τ_2. Hue encodes orientation and saturation encodes magnitude.

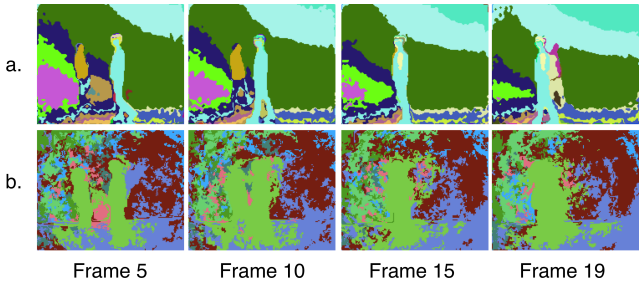

Fig. 7. Segmentation results using [8] with the presented range flow. *a.* Segmentation of depth. *b.* Segmentation of intensity. Segments labels are color coded.

algorithm with respect to [8]. In contrast to [8], our algorithm integrates long-term motion information in an early stage of the algorithm, i.e. groups trajectories according to common movement and spatial proximity. This additional motion information can compensate for noise in the range video.

4. CONCLUSION

In this paper we presented a framework for motion segmentation of video sequences from ToF cameras. We extracted motion information using fusion of range flow and optical flow. We showed that a confidence measure in range flow estimation helps to distinguish between reliable flow vectors and noise. This is especially advantageous in areas with low signal to noise ratio. As a second contribution we derive long-term point trajectories and group trajectories of coherent motion. Future work will concentrate on more advanced techniques for fusion of range and intensity data, based on factors like texture, geometry and noise levels.

5. REFERENCES

[1] W. Karel, "Integrated range camera calibration using image sequences from hand held operation," in *Proceedings of the 21st Congress of the International Society for Photogrammetry and Remote Sensing*, 2008, vol. 37, pp. 945–951.

[2] W. Karel, S. Ghuffar, and N. Pfeifer, "Modelling and compensating internal light scattering in time of flight range cameras," *The Photogrammetric Record*, To appear.

[3] H. Spies, B. Jähne, and J.L. Barron, "Regularised range flow," in *Proceedings of European Conference on Computer Vision*, 2000, pp. 785–799.

[4] M. Yamamoto, P. Boulanger, J. Beraldin, and M. Rioux, "Direct estimation of range flow on deformable shape from a video rate range camera," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 82–89, 1993.

[5] J.L. Barron and H. Spies, "The fusion of image and range flow," *Multi-Image Analysis, Lecture Notes in Computer Science*, vol. 2032, pp. 171–189, 2001.

[6] M. Schmidt, M. Jehle, and B. Jähne, "Range flow estimation based on photonic mixing device data," *International Journal of Intelligent Systems Technologies and Applications*, vol. 52, pp. 380–392, 2008.

[7] K. Koffka, *Principles of Gestalt Psychology*, Routledge Chapman and Hall, 1999.

[8] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *Proceedings of the 23rd Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1–14.

[9] J. Lezama, K. Alahari, J. Sivic, and I. Laptev, "Track to the future: Spatio-temporal video segmentation with long-range motion cues," in *Proceedings of the 24th Conference on Computer Vision and Pattern Recognition*, 2011.

[10] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *Proceedings of the 11th European Conference on Computer Vision: Part V*, 2010, pp. 282–295.

[11] P. Sand and S. Teller, "Particle video: Long-range motion estimation using point trajectories," in *Proceedings of the 19th Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2195–2202.

[12] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by gpu-accelerated large displacement optical flow," in *Proceedings of the 11th European Conference on Computer Vision: Part I*, 2010, pp. 438–451.

[13] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, pp. 167–181, 2004.