Integration of Region Tracking and Optical Flow for Image Motion Estimation

Ulrich Neumann and Suya You
Computer Science Department
Integrated Media Systems Center
University of Southern California
{uneumann | suyay}@graphics.usc.edu

Abstract

Robust motion estimation is obtained by integrating optical flow and region tracking methods into a closed-loop architecture. To ensure robust tracking over long intervals in video image sequences, the system has mechanisms to predict and assess tracking performance. Our application is to constrain the estimation of the 3D motion of a camera.

1 Introduction

Many applications depend on a measurement of motion in image sequences, including dynamic scene analysis and understanding, image registration and stabilization, visual navigation and obstacle avoidance, and video data compression. The 2D image motion field is a projection of the 3D velocities of surface points onto the imaging plane. Our work extends motion estimation techniques to the video-based augmented reality [1] tracking problem. By extracting the 2D image motion field directly from a live video stream, we can estimate or constrain the real camera 3D motion parameters to maintain dynamic alignment between a real and virtual camera. Our goal is to compute the motion field without depending on artificial landmarks (fiducials) as many existing systems do, and thereby extend the application domain of augmented reality systems to natural outdoor environments.

Methods for computing motion fields often rely on spatial and temporal gradients of the image intensity [2-8]. The estimation of a pixel-density motion field is an ill-posed problem, so additional constraints are required. For example, a quadratic smoothness constraint [3] or a higher order spatial-temporal derivative constraint [4] make the problem solvable. Other approaches for motion estimation include feature-based methods that detect and track recognizable features throughout the image sequence [5]. Correlation-based approaches compare small patches of an image with nearby patches in neighboring frames [6]. Frequency-based methods resolve image velocities in a spatiotemporal transformation domain [7].

Most of the existing motion estimation approaches operate on the entire image, intending to compute a motion vector for every pixel. It is difficult, except in trivial cases, to recover accurate and dense motion fields in a natural environment. Difficulties arise from scene geometry and texture complexity; variations in lighting and shading; and occlusion. Our approach concedes the goal of dense motion field computation in exchange for sparse but reliable motion estimates. We compute confidence measures to predict and verify the most reliable regions of estimated motion [2, 3]. The problems we confront include: What regions of the image are most reliable for motion estimation? How can we actively select and control the reliable regions? Can we track these regions of high confidence image motion for use in 3D motion estimation?

In this paper, we present a robust approach that integrates region tracking and optical flow techniques to cope with the problems mentioned above. We adaptively select image regions based on their suitability for tracking and motion estimation. Since the selected regions produce reliable motion estimation fields, we can track them over time with a region tracking method, to maintain the reliability of motion field estimation and enable 3D motion determination.

The motion field is computed with a multi-stage and iterative computing strategy (Fig 1). The method relies on a metric for determining the error of the estimated field. Our approach is designed to be both reliable and efficient. The computed flow field is fed back to the region tracking module, and this feedback helps maintains the consistency of motion estimates.

2 System Overview

Figure 1 shows the overall system architecture. It integrates three main motion analysis functions: feature selection, tracking, and verification, in a closed-loop cooperative manner. In the feature selection module, two types of image features (points and regions) are selected and evaluated for their suitability for reliable tracking and motion estimation. Feature selection is also based on an evaluation process that measures the confidence of tracking during past estimations. Once selected, features are ranked according to their evaluation values, and then fed into the tracking module.

Our tracking method incorporates a differential-based local optical flow estimation, and we perform a least-squares minimization to find the best fit to a local motion model. Unlike traditional implementations, we developed a multi-stage estimation strategy. For every estimated result, a verification and evaluation are imposed to measure the confidence of estimation. If the estimation error is large, the result is refined to until the error converges, or the feature is discarded. In order to handle local geometric distortions due to large view variations and long sequence tracking, two motion models, a translation model and affine model are used for the point and region features, respectively. These model parameters are utilized for motion verification and evaluation.

Motion verification and feedback are essential components of the architecture for robust tracking. Two verification strategies are used for the two types features (point and region) and motion models (translation and affine). Both verification models depend on the estimated region motion field to synthesize an evaluation frame. The evaluation frame is compared to the true target frame and the difference is a verification metric for that estimate. This verification information is fed back to the tracking module for motion correct and to the feature detection module for feature re-evaluation. The closed-loop approach is inspired by the use of feedback for correcting errors in non-linear control systems. The process acts as "selection-hypothesis-verification-correction" strategy that makes it possible to discriminate between good and poor estimates of feature motions. Selecting the highest quality features and estimates maximizes the accuracy of the final 2D and 3D motion estimates.

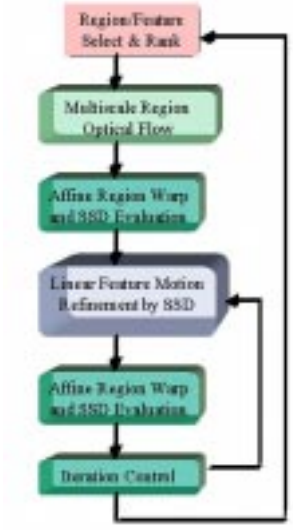


Fig. 1 – Functional blocks in the multi-stage integration of optical flow and region tracking functions

3 Feature Selection and Evaluation

The tracking features selection should be adaptive and fully data-driven. Before deriving our selection metric, we first introduce the motion estimation model we used for optical flow computing.

3.1 Differential-based local optical flow model

The fundamental assumption in differential-based optical flow estimation is intensity conservation, from which the principal relationship between intensity derivatives and optical flow, the *optical flow gradient constraint equation* [2], can be derived

$$\nabla I(\mathrm{x},t)\cdot \mathrm{v} + I_t(\mathrm{x},t) = 0 \qquad (1)$$

Where $I_t(\mathrm{x},t)$ denotes the partial time derivative of $I(\mathrm{x},t)$, $\nabla I(\mathrm{x},t) = \left(I_x(\mathrm{x},t), I_y(\mathrm{x},t)\right)$, and $\nabla I \cdot \mathrm{v}$ denotes the usual dot product. Unconstrained flow fields are typically not well described by any global model. Different local models have been used to facilitate the estimation process, including constant flow with a local window and locally smooth or continuous flow. The former facilitates direct local estimation [8], whereas the latter model requires iterative relaxation. In our work, we use the local constant model, because its results compare favorably with others [2]. In this model, the optical flow field is fitted to a constant model in each small spatial neighborhood. Optical flow estimates are computed by minimizing the weighted least-squared fitting

$$E(\mathrm{x}) = \sum_{\mathrm{x}\in\Omega} W^2(\mathrm{x})\left[\nabla I(\mathrm{x},t)\cdot \mathrm{v} + I_t(\mathrm{x},t)\right]^2 \qquad (2)$$

where $W(\mathrm{x})$ denotes a window function that gives more influence to constraints at the center of the neighborhood than those at the periphery. Minimizing this fitting error with respect to *v* leads to the equation $\nabla E(\mathrm{v}) = 0$, from which the optical flow field can be solved as

$$\mathrm{v} = \mathrm{A}^{-1}\mathrm{B} \qquad (3)$$

where

$$\mathrm{A} = \sum_{\mathrm{x}\in\Omega} W^2(\mathrm{x}) \begin{bmatrix} I_x(\mathrm{x},t)^2 & I_x(\mathrm{x},t)I_y(\mathrm{x},t) \\ I_y(\mathrm{x},t)I_x(\mathrm{x},t) & I_y(\mathrm{x},t)^2 \end{bmatrix}$$

$$\mathrm{B} = \sum_{\mathrm{x}\in\Omega} W^2(\mathrm{x}) \begin{bmatrix} I_x(\mathrm{x},t)I_t(\mathrm{x},t) \\ I_y(\mathrm{x},t)I_t(\mathrm{x},t) \end{bmatrix}$$

The local model method repeatedly solves this linear system and then integrates the computed values of **v** to provide an estimate of the optical flow field over the aggregate time interval.

3.2 Point Feature Detection

The motion estimation equation (3) can be solved in a closed form only when the 2×2 matrix $\mathbf{A}$ is nonsingular. We can reliably compute the value of optical flow at a point if the system represents suitable measurements and can be solved reliably at that point. The rank of the matrix $\mathbf{A}$ will be full unless the directions of gradient vectors everywhere within the window coincide. It must be well conditioned, which means the eigenvalues of $\mathbf{A}$ can not be too small and their lengths can not differ by several orders of magnitude. We also note that the matrix $\mathbf{A}$ is actually a covariance matrix of a region of the image. It measures the correlation of image intensity, which can be a good indicator to measure the distributions of image structures over a small patch [9]. Summarizing this relationship between the matrix $\mathbf{A}$ and image structure, small eigenvalues of $\mathbf{A}$ correspond to a relatively constant intensity within a region. A pair of large and small eignvalues corresponds to a uni-directional texture pattern, and two large eigenvalues represent corners, salt-and-pepper textures or other patterns.

The eigen-distribution of covariance matrix $\mathbf{A}$ is one metric for selecting tracking features in our system since it predicts the confidence for the optical flow computation at a point. If both eigenvalues of $\mathbf{A}$ are greater than a pre-defined threshold, then the point is accepted initially as a candidate tracking feature.

$$\min(\lambda_1, \lambda_2) > TH \tag{4}$$

After the feature candidates are selected by the constraint (Eq. 4), an evaluation value is assigned for each candidate to indicate the its expected reliability of detection and tracking. If λ is the minimum eigenvalue of above matrix A, and δ is the tracking confidence fed back from tracking module, the evaluation value assigned to each candidate is defined as

$$C = k_1\lambda + k_2\delta \tag{5}$$

where k_1, k_2 are weighting coefficients for each component. The detected point-feature candidates are ranked according to their evaluation values, and the highest-ranking points are selected as tracking features $\{PF_i\}$. (The number of points selected depends on the application and computing budget.)

$$\{PF_i = \mathrm{x}_i(C) \mid i \in \textit{candidate set}, C > \textit{threshold}\} \tag{6}$$

The selected feature list can be updated dynamically, but update is generally not required every frame. While the system tracks a sufficient number of points and regions, the update is deferred. List update is only required to replace points and regions that pass out of view and whose evaluation value falls below an acceptable threshold. Since the evaluation metric utilizes both the point's predicted optical flow stability (λ) and its model-fitted tracking confidence (δ), the selected points are arguably the "best" features for this particular tracking method.

3.3 Region Feature Detection

A similar metric is also used to detect reliable region features. Since the role of region tracking is to assist the optical flow estimate by providing a global guidance for fine motion estimation, a region is deemed reliable for tracking if it includes sufficient good point features. In our implementation, the image is first divided into non-overlapping regions R_i. The number of good points in each candidate regions is tabulated and we select the N "best" regions $\{RF_i\}$ as

$$\{RF_i = R_i(S) \mid i \in \textit{candidate region}, S > \textit{threshold}\} \tag{7}$$

where the metric of quality is *S,* given by

$$S = \frac{N_g}{N_T} \tag{8}$$

where N_g is the number of good points within the region, and N_T is the number of pixels in the region. The number of selected regions is arbitrary, depending on the complexity of the scene structure and the application, but each region adds to the computing time.

4 Multi-Stage Tracking Strategy

Any imaging distortion, especially in the natural environment, may significantly effect a feature's appearance and result in unreliable tracking. To achieve robust tracking, we developed a multi-stage computing strategy that utilizes the current motion estimate to synthesize a confidence frame. The confidence frame is fed back to the tracking approach to verify and

evaluate the current motion estimate. The confidence frame models the motion deformations, so it can effectively compensate for those effects.

The multi-stage strategy has three steps: image warping, residual motion estimation, and motion model refinement.

Let $R_{t_0}(\mathbf{x}, t_0)$ be a selected region for tracking in the frame t_0. $R_t(\mathbf{x}, t)$ is its correspondence that has been tracked successfully at time t, and the estimated motion vector is $\mathbf{v} = [v_1, v_2, \cdots, v_6]$. The estimated motion vector models the translation motion of the region (the center of the region) and the deformation (affine) of the region. A new region $R_c(\mathbf{x}, t)$ can been reconstructed, from $\mathbf{v}$ the estimated motion parameters, by warping the region $R_{t_0}(\mathbf{x}, t_0)$ towards $R_t(\mathbf{x}, t)$.

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} v_2 x_{t_0} + v_3 y_{t_0} + v_1 \\ v_4 x_{t_0} + v_5 y_{t_0} + v_6 \end{bmatrix} \tag{9}$$

Since the new region is generated directly from its motion estimate, it can be used to measure of how well the region is tracked. This newly constructed region is called a confidence frame.

The motion residual is computed based as the least-square distance between the confidence frame $R_c(\mathbf{x}, t)$ and its target $R_t(\mathbf{x}, t)$,

$$\varepsilon = \frac{\left\| R_t(\mathbf{x}, t) - R_c(\mathbf{x}, t) \right\|^2}{\max\{\left\| R_t(\mathbf{x}, t) \right\|^2, \left\| R_c(\mathbf{x}, t) \right\|^2\}} \tag{10}$$

and this is the estimate of the motion tracking error in the current frame.

If the estimate error is large, the current region underwent significant motion or distortions that were not captured in the current motion estimate. The estimate is refined by re-computing the motion between $R_c(\mathbf{x},t)$ and $R_t(\mathbf{x},t)$. The process is repeated until the estimation error converges to within a pre-defined threshold. If the motion residual remains larger than the threshold after a fixed number of iterations, the region is eliminated from the feature list, and a re-selection process is started.

The multi-stage procedure is applied to both the point tracking and region tracking, but different processes are used. The region confidence frame is built by warping based on an affine motion model. The point confidence frame is just a translation based on the estimated motion field. Since the role of region tracking is to provide global guidance for fine (point) motion estimation, it finds and compensates for global distortions during long sequence tracking.

5 Tracking Feedback

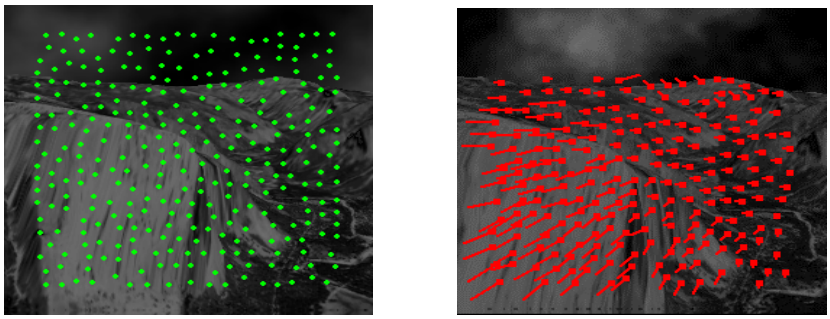
The tracking confidence is fed back to the feature detection module to guide selection and detection. This is vital for keeping the whole system operating in the "optimum" state of tracking the most reliable features. In section 3.2, the tracking confidence δ is utilized for computing the evaluation value of feature detection (Eq. 5). The evaluation value is employed to select and rank features according their reliabilities. The role of δ is to provide dynamic information about a feature's tracking reliability, and we defined it as

$$\delta = \frac{1}{1+\varepsilon} \tag{11}$$

where ε is the motion residual defined in (Eq. 10).

6 Results and Conclusions

The tracking system described above has been implemented and evaluated for a number of synthetic image sequences and live video sequences. Figure 2 illustrates an experiment result (Yosemite-Fly-Through) for evaluating the optical flow approach. The sequence is graphically generated by moving a camera along its line of sight towards the mountain and valley, generating diverging motion flow around the upper right of the mountain. A one pixel/frame translating motion is obtained in the cloud area, and about four pixels/frame are in the lower left of the image. In this test, only one image region is selected as a tracking region whose size is the same as the original image size (256x256). In the tracking region, 50% of the pixels whose evaluation values are top ranked are selected as point features. We chose these numbers for performance comparisons with other optical flow approaches that compute motion estimates for full images. Figure 2 (a) shows the selected tracking points, and Figure 2 (b) illustrates the final tracking results at the 15th frame. In this test, about 3% of the selected features are declared as unreliable due to their low tracking confidence. The resulting average angle error [2] is 2.84, and the RMS measure is 4.51. To keep the performance analysis equitable, we compared our approach with some other optical flow approaches, including Horn and Schunck's global regularization algorithm [3], Lucas and Kanade's local differential method [8], Anandan's matching correlation algorithm [6], and Fleet and Jepson's frequency-based method [7]. The results show superior



(a) (b)

Fig. 2 - Yosemite-Fly-Through image sequence with detected tracking features (a), and estimated motion field (b)

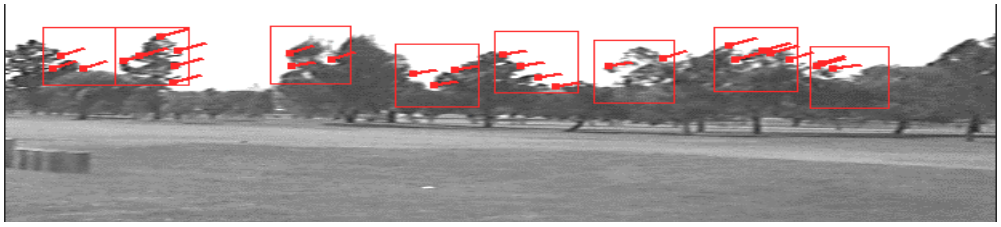


Fig. 3 – Natural features and regions that are dynamically selected and tracked in outdoor scene

accuracy from our multi-stage processing technique. Figure 3 shows tracking of a natural outdoor scene.

This paper presented our system for robust image motion estimation with integrating region tracking and optical flow techniques. The multistage tracking process produces accurate motion field estimates, and the entire system operates in a closed-loop that stabilizes its performance and accuracy. This strategy appears to provide robust tracking for our intended application of augmented reality tracking in outdoor environments.

References

1. U. Neumann and J. Park. Extendible Object-Centric Tracking for Augmented Reality. *Proc. of IEEE Virtual Reality Annual International Symposium*, 148-155, 1998.

2. S. S. Beauchemin and J. L. Barron. The Computation of Optical flow. *ACM computing surveys*, 27(3):433-466, 1995.

3. B. K. P. Horn and B. G. Schunk. Determining Optical Flow. *Artificial Intelligence*, 17:185-203.

4. H. H. Nagel. On a Constraint Equation for the Estimation of Displacement Rates in Image Sequences. *IEEE Trans. PAME*, 11:13-30, 1989.

5. E. C. Hildreth. Computation Underlying the Measurement of Visual Motion. *Artificial Intelligence*, 23:309-354, 1984.

6. P. Anandan. A Computational Framework and an Algorithm for the Measurement of Visual Motion. *Int. J. Comp. Vision*, 2:283-310, 1989.

7. D. J. Fleet and A. D. Jeson. Computation of Component Image Velocity from Local Phase Information. *Int. J. of Computer Vision*, 5:77-104, 1990.

8. B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. *Proc. DARPA IU Workshop*, 121-130, 1981.

9. C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method. Technical Report, Carnegie Mellon University, Pittsburgh, PA, September 1990.

10. G. D. Hager and P.N. Belhumeur. Real-Time Tracking of Image Regions with Changes in Geometry and Illumination. *IEEE Proc. of CVPR*, 1996.