A Head-Wearable Short-Baseline Stereo System for the Simultaneous Estimation of Structure and Motion

Hernán Badino and Takeo Kanade
Carnegie Mellon University
Pittsburgh, PA, USA

Abstract

This paper presents a short-baseline real-time stereo vision system that is capable of the simultaneous and robust estimation of the ego-motion and of the 3D structure and the independent motion of thousands of points of the environment. Kalman filters estimate the position and velocity of world points in 3D Euclidean space. The six degrees of freedom of the ego-motion are obtained by minimizing the projection error of the current and previous clouds of static points. Experimental results with real data in indoor and outdoor environments demonstrate the robustness, accuracy and efficiency of our approach. Since the baseline is as short as 13cm, the device is head-mountable, and can be used by a visually impaired person. Our proposed system can be used to augment the perception of the user in complex dynamic environments.

1 Introduction

In this paper we present a wearable mobile system that provides information of 3D structure, independent motion and ego-motion to augment the perception of the user in complex environments. We have built a prototype consisting of a helmet with stereo cameras connected to a portable computer system (Figure 2). As the user navigates in its environment, the system detects and tracks image features and computes their corresponding stereo disparities. The features and disparities of consecutive frames are used to compute the ego-motion of the camera using a robust least squares algorithm. A Kalman filter then fuses the feature tracking, the stereo disparity and the extracted ego-motion to iteratively estimate the 3D position and 3D velocity of each tracked feature in a user oriented coordinate system. Figure 1 shows an example of the output obtained in a crowded scene containing multiple moving objects. Our proposed methods can be used to augment the perception of the visually impaired in complex dynamic environments. The system runs in real time at 17 Hz using a standard laptop, and it was tested online in countless occasions.

2 Related Literature

The simultaneous estimation of structure and motion from stereo images has been heavily covered by the literature. We give here a brief review of the most related methods. Jung and Lacroix [4] uses Kalman filters to refine estimates of ego-motion and 3D landmark position of static world points. Agrawal et al. [1] and Talukder and Matthies [13] estimate independently moving objects by detecting and tracking blobs in the image. The blobs are obtained from image regions that are not in accordance with the computed ego-motion. Similarly, Ess et al. [2] present a framework for the detection of independent motion with a freely moving camera in crowded scenes. Franke et al. [3] use Kalman filters to track independent motion using stereo cameras. The ego-motion of the cameras is obtained from the inertial sensors of a vehicle. Rabe et al. [9] extended this approach to dense motion fields using FPGA and GPU implementations. Klein and Murray [5] also track features using a monocular system for the real time estimation of camera motion and structure for augmented reality applications. Independent motion is not modeled but treated as outlier. Vision has also been used to provide navigation support to the visually impaired. A survey of navigation systems of the visually impaired can be found in [15]. Lu and Manduchi [7] present a stereo system to detect curbs and stairways. Sáez and Escolano [12] detect aerial obstacles in near real time, but only static scenes are considered. Treuillet et al. [14] propose a similar application to localize and guide the walker along a predefined path by using a monocular camera.

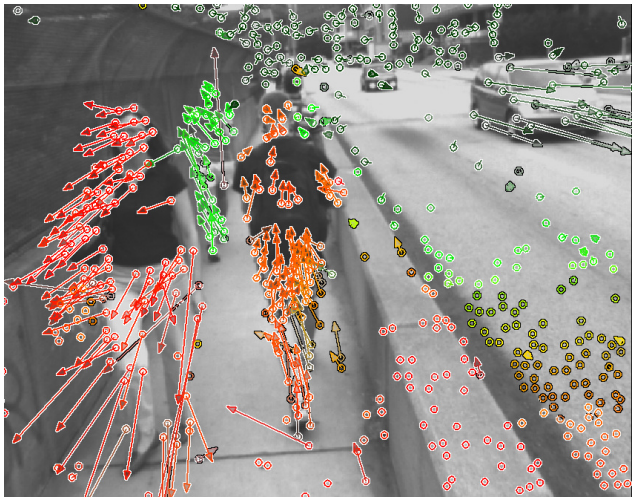


Figure 1. Output of our system while moving in a complex environment. Tracked features are shown as a circle. The vectors show the direction and speed of the features on moving objects.

In contrast to the above methods, we track stereo features and use Kalman filters to estimate their position and velocity in Euclidean space while simultaneously estimating the ego-motion of the camera using a robust method. Furthermore, our proposed system runs in real time at 17 Hz using a single laptop.

3 Tracking 3D Points

This section presents a Kalman filter method that iteratively estimate the position and velocity of individually tracked feature points. The Kalman filter we describe in this section is derived from Franke et

12th IAPR Conference on Machine Vision Applications.
June 13-15, 2011. Nara, Japan.

al. [3], where it was used to detect moving object in the automotive domain. We have expanded the system model equations to allow a freely moving camera instead of a motion on the plane, as originally proposed.

System Model. Let $\boldsymbol{p}_{k-1} = (X, Y, Z)^T$ represent the coordinate vector of a world point observed by the system at time $k-1$ and $\boldsymbol{v}_{k-1} = (\dot{X}, \dot{Y}, \dot{Z})^T$ represent its associated velocity vector. The camera platform moves in its environment with a given translational and angular velocity, changing its relative position to the point. After a time Δt_k the new position of the point from the camera point of view is given by

$$\boldsymbol{p}_k = \boldsymbol{R}_k \boldsymbol{p}_{k-1} + \boldsymbol{t}_k + \Delta t_k \boldsymbol{R}_k \boldsymbol{v}_{k-1}$$

where $\boldsymbol{R}_k$ and $\boldsymbol{t}_k$ are the rotation matrix and translation vector of the static scene with respect to the camera. The velocity vector $\boldsymbol{v}_k$ changes its direction according to:

$$\boldsymbol{v}_k = \boldsymbol{R}_k \boldsymbol{v}_{k-1}$$

Combining position and velocity in the state vector

$$\boldsymbol{x_k} = (X, Y, Z, \dot{X}, \dot{Y}, \dot{Z})^T \tag{1}$$

leads to the discrete linear system model equation:

$$\boldsymbol{x}_k = \boldsymbol{A}_k \boldsymbol{x}_{k-1} + \boldsymbol{B}_k + \boldsymbol{\rho}_k$$

with the state transition matrix

$$\boldsymbol{A}_k = \begin{bmatrix} \boldsymbol{R}_k & \Delta t_k \boldsymbol{R}_k \\ \boldsymbol{0_{3\times 3}} & \boldsymbol{R}_k \end{bmatrix} \tag{2}$$

and input vector

$$\boldsymbol{B}_k = (\boldsymbol{t}_k^T, 0, 0, 0)^T \tag{3}$$

The term $\boldsymbol{\rho}_k$ is assumed to be Gaussian white noise.

Measurement Model. A measurement is defined by the vector $\boldsymbol{m} = (u, v, d)^T$, where (u, v) corresponds to the image position of the feature point and d is its disparity. The (u, v) components are obtained from the feature tracking algorithm, while the disparity d is obtained from the stereo algorithm. The non-linear measurement equation $\boldsymbol{h}$ for the state vector of Equation 1 is

$$\boldsymbol{h}(\boldsymbol{x}_k) = \begin{bmatrix} u \\ v \\ d \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \\ B \end{bmatrix} + \boldsymbol{\nu}$$

where f is the focal length of the camera and B is the baseline of the stereo system. The term $\boldsymbol{\nu}$ is assumed to be Gaussian white noise. Since the measurement equation is non-linear, the extended Kalman Filter is used.

4 Visual Odometry Estimation

The computation of ego-motion is one of the most fundamental tasks for most mobile vision problems. The accurate knowledge of the motion of the camera allows to integrate estimates in a global coordinate system. It also provides a solid constraint to detect independent motion. Equations 2 and 3 require $\boldsymbol{R}_k$ and $\boldsymbol{t}_k$ (the rotation and translation of the static scene, i.e. the inverse motion of the camera). This section presents a method for their robust estimation.

Least Squares Formulation. Given a set of tracked feature points $\boldsymbol{m}_i = (u_i, v_i, d_i)^T$ for $i = 1, 2, \cdots, n$ in the current frame, and the set of corresponding feature points $\boldsymbol{m}'_i = (u'_i, v'_i, d'_i)^T$ in the previous frame, we seek to estimate the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$, such that for all points in the sets, $\boldsymbol{g}(\boldsymbol{m}'_i) = \boldsymbol{R}\, \boldsymbol{g}(\boldsymbol{m}_i) + \boldsymbol{t}$, with $\boldsymbol{g}() = \boldsymbol{h}^{-1}()$, i.e., the triangulation equation. One way of obtaining the translation and rotation is to calculate the absolute orientation between both set of points. Many solutions to the absolute orientation problem exist when the error in the 3D points is isotropic [6]. However, stereo triangulation error can be highly anisotropic and correlated [11]. Instead of minimizing the residuals in Euclidean space, we minimize them in the image space, where the noise level is similar for all components of the measurement vector:

$$E = \arg\min_{\{\boldsymbol{R},\boldsymbol{t}\}} \frac{\sum_{i=1}^{n} w_i^2 \left(\boldsymbol{m}'_i - \boldsymbol{h}(\boldsymbol{R}\, \boldsymbol{g}(\boldsymbol{m}_i) + \boldsymbol{t})\right)^2}{\sum_{i=1}^{n} w_i^2} \tag{4}$$

where w_i is a weighting factor determining the contribution of the measurement to the least square solution. In order to minimize Equation 4, the rotation matrix $\boldsymbol{R}$ is parameterized by the pseudo-vector $\boldsymbol{r} = (w_x, w_y, w_z)^T$. The matrix $\boldsymbol{R}$ is obtained by rotating the identity matrix $|\boldsymbol{r}|$ radians around the axis $\boldsymbol{r}/|\boldsymbol{r}|$. Assuming $\boldsymbol{t} = (t_x, t_y, t_z)^T$, the parameter for minimization is then the six-dimensional vector $\mathbf{x} = (w_x, w_y, w_z, t_x, t_y, t_z)^T$.

Newton Minimization. Because of the non-linearity imposed by the rotation and the projection equation $\boldsymbol{h}()$, we use an iterative Newton optimization method to solve Equation 4, for which we require the computation of first and second order derivatives of the loss function, i.e.:

$$\boldsymbol{\mathcal{J}}_{[i]} = \frac{\partial \boldsymbol{E}}{\partial \mathbf{x}_{[i]}} \quad \text{and} \quad \boldsymbol{\mathcal{H}}_{[i,j]} = \frac{\partial^2 \boldsymbol{E}}{\partial \mathbf{x}_{[i]} \mathbf{x}_{[j]}}$$

Given an initial estimate $\mathbf{x}_0$, the Newton method iteratively converge to a local minimum by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \boldsymbol{\mathcal{H}}_k^{-1} \boldsymbol{\mathcal{J}}_k. \tag{5}$$

Equation 5 is iteratively applied until the residual E_k of Equation 4 is small enough, or no significant change in the estimate is observed. The closer the initial value is to the real solution, the less iterations are required to find a minimum. For small camera motion, it is usually enough to set $\mathbf{x}_0$ to the zero vector. For a large motion between frames, an inertial sensor unit can provide the initial estimate of the motion of the camera. For the experiments shown in Section 5, the method converges in less than six iterations.

Iterative Robust Estimation. The sets of feature correspondences often contain outliers due to false correspondences or moving objects. In order to cope with the outliers, an optimization approach is applied that iteratively rejects them. Assuming that the outliers are bounded (constraint imposed by the stereo and tracking algorithm), the motion estimated by the Newton method will be approximately correct, and therefore,

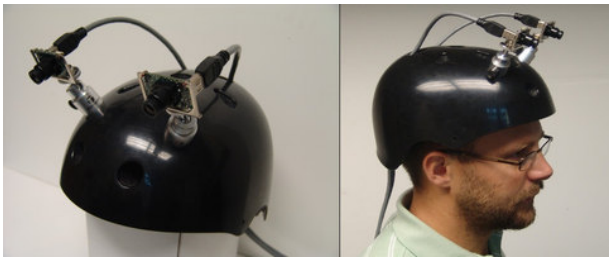


Figure 2. Wearable mobile stereo head

the outliers will have a larger residual than the inliers. This provides a simple method for outliers rejection: if the residual for a given feature is larger than a threshold, the feature is eliminated from the set, and the whole estimation process is repeated until convergence (Algorithm 1). A nice property of the above procedure is that the Newton method converges faster after each robust estimation cycle, since the resulting estimate is closer to the solution. After the initial estimate, the Newton method usually requires only two cycles to converge.

Input: Sets $\{\boldsymbol{m}_i\}$ and $\{\boldsymbol{m}'_i\}$ and vector $\mathbf{x}_0$
Output: Rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$
Initialize $k = 0$, $w_i = 1$ for $i = 1, 2, \cdots, n$;
# Start robust estimation cycle
while *not converged* **do**
  # Start newton minimization cycle
  while *not converged* **do**
    Calculate $\boldsymbol{\mathcal{J}}_k$, $\boldsymbol{\mathcal{H}}_k$ and E_k at $\mathbf{x}_k$;
    Update: $\mathbf{x}_{k+1} = \mathbf{x}_k - \boldsymbol{\mathcal{H}}_k^{-1}\boldsymbol{\mathcal{J}}_k$;
    Increase k;
  end
  Update $r^2_{max} = 3^2 E_k$;
  # Outlier rejection cycle
  foreach *pair* $\boldsymbol{m}_i, \boldsymbol{m}'_i$ **do**
    Calculate $r^2 = (\boldsymbol{m}'_i - \boldsymbol{h}(\boldsymbol{R}\,\boldsymbol{g}(\boldsymbol{m}_i) + \boldsymbol{t}))^2$;
    if $r^2 > r^2_{max}$ **then**
      Set $w_i = 0$; # outlier
    else
      Set $w_i = 1$; # inlier
    end
  end
end

Algorithm 1: Robust ego-motion estimation algorithm.

The final value E_k after each Newton cycle in Algorithm 1 is a measure of the average variability of the residuals. We define inliers as those measurements for which their squared residual is smaller than $3^2 E_k$. Assuming that the residuals are normally distributed, that threshold ensures that 99.7% of the features belong to the same distribution.

5 Experimental Results

We have implemented the proposed method in C++ using OpenMP technology to benefit from multi-core processing. Our algorithm runs in real time on a standard laptop PC, and we have extensively tested the algorithms on-line in innumerable scenes and situations. We use the KLT algorithm [10] for tracking features. In our configuration, KLT provides up to 1024 tracks with a relatively low computational cost. The stereo disparity of a feature is computed by correlating a window of size 15×15 px centered on the feature position. We use a pyramidal implementation for both, tracking and stereo computation. The baseline of the stereo system is 12.8 cm with a focal length of 654 px and image size of 640×480 px.

"Bridge" Data Set. More than 2000 images were acquired as the user was walking through the sidewalk of a bridge with a length of approximately 60 meters. Figure 3 shows some excerpts of the sequence. The sequence is challenging because it contains not only repetitive structures, lack of texture, and semi-transparencies produced by the railing at the left, but also multiple pedestrians and vehicles moving in both directions. In particular, the middle of the sequence presents a difficult situation for the estimation of the camera's ego-motion. The images are occupied with up to a 30% of moving objects (see Figure 1 and the second and third row of Figure 3). The robust least squares algorithm presented in Section 4 was still able to provide a correct ego-motion estimate in those situations.

Observe that, since we build a local map of the environment, the visual odometry error will grow superlinearly over time [8]. Nevertheless, our robust algorithm is accurate enough to allow the generation of accurate 3D reconstructions of large environments with small drift. In order to demonstrate this, we have performed a reconstruction of the scene by accumulating all observed static 3D points − excluding moving points such as those on pedestrians − into the same reference frame. Figure 4 shows the reconstruction result. It can be seen that the ego-motion algorithm was not only robust throughout the sequence, but also precise enough to produce a coherent spatial perception of the real overall structure.

"Footbridge" Data Set. A sequence containing 750 images was acquired inside a building. Figure 5a shows a picture of the tested environment. The user started approximately at the camera position of Figure 5a and then turned left to walk on the footbridge. Figures 5b and 5c show the structure obtained by the accumulation of all observed static points of the sequence. As it can be seen from the bird's eye view of Figure 5b, the estimation was accurate enough to provide an almost perfect planar reconstruction of the lateral footbridge wall. A careful inspection of the ego-path shown in Figure 5c reveals the typical sinusoidal undulation performed when walking.

Computation Times. The following table shows the measured average computation times for the "Bridge" data set in milliseconds.

Rectification	Stereo	Ego-Motion	KLT	KF	Total
3.66	4.19	29.83	19.81	0.68	58.17

A laptop with an Intel Core 2 Duo 2.66GHz CPU was used. The images were down sampled by a factor of 2. The current implementation of the algorithm can process online video at 17 Hz.

Figure 3. Excerpts of the outdoor sequence. The circles show the tracked features. The color encodes the depth of the corresponding 3D point, where red is close (less than 1 m), yellow is at a middle distance (between 1 and 3 m) and green is far (more than 3 m). The vectors show the Kalman filtered predicted position of the 3D point in 0.5 seconds, back projected to the image.

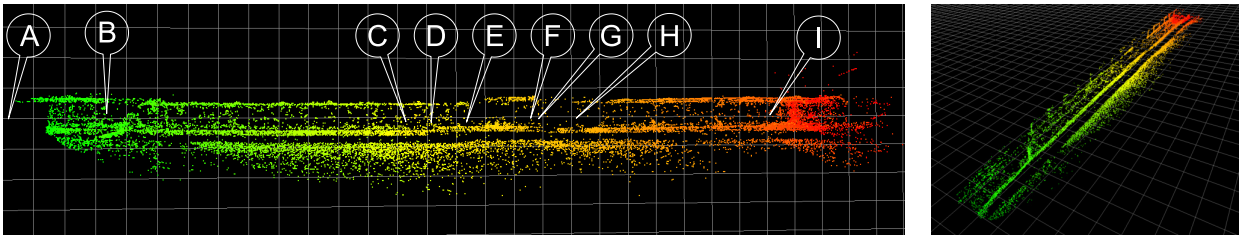


Figure 4. Bird's eye views of the sidewalk's 3D reconstruction. The labels show the approximate positions of the images shown in Figure 3. The color encodes the time at which the 3D points were first observed.

6 Future Work

The information obtained from our system can be used to augment the perception of the visually impaired in complex dynamic situations. In particular, the estimation of static objects will allow the user to plan detours around obstacles and hazards, while providing reference points for orientation. The estimation of the direction and intensity of the motion of pedestrians and vehicles is useful to avoid collisions and to follow a person in a determined path. Our future works includes the design of a suitable interface taking into account the psychological and ergonomic factors of the end user.

References

[1] M. Agrawal, K. Konolige, and L. Iocchi. Real-time detection of independent motion using stereo. In *IEEE Workshop on Motion and Video Computing*, 2005.

[2] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Moving obstacle detection in highly dynamic scenes. In *ICRA*, 2009.

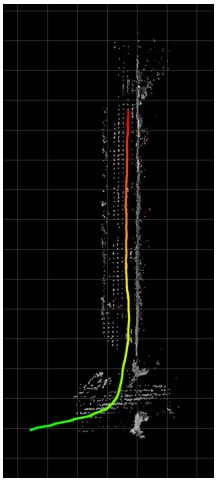
[3] U. Franke, C. Rabe, H. Badino, and S. Gehrig. 6d-vision: Fusion of stereo and motion for robust environment perception. In *DAGM '05*, 2005.

[4] I.-K. Jung and S. Lacroix. High resolution terrain mapping using low altitude aerial stereo imagery. In *ICCV*, 2003.

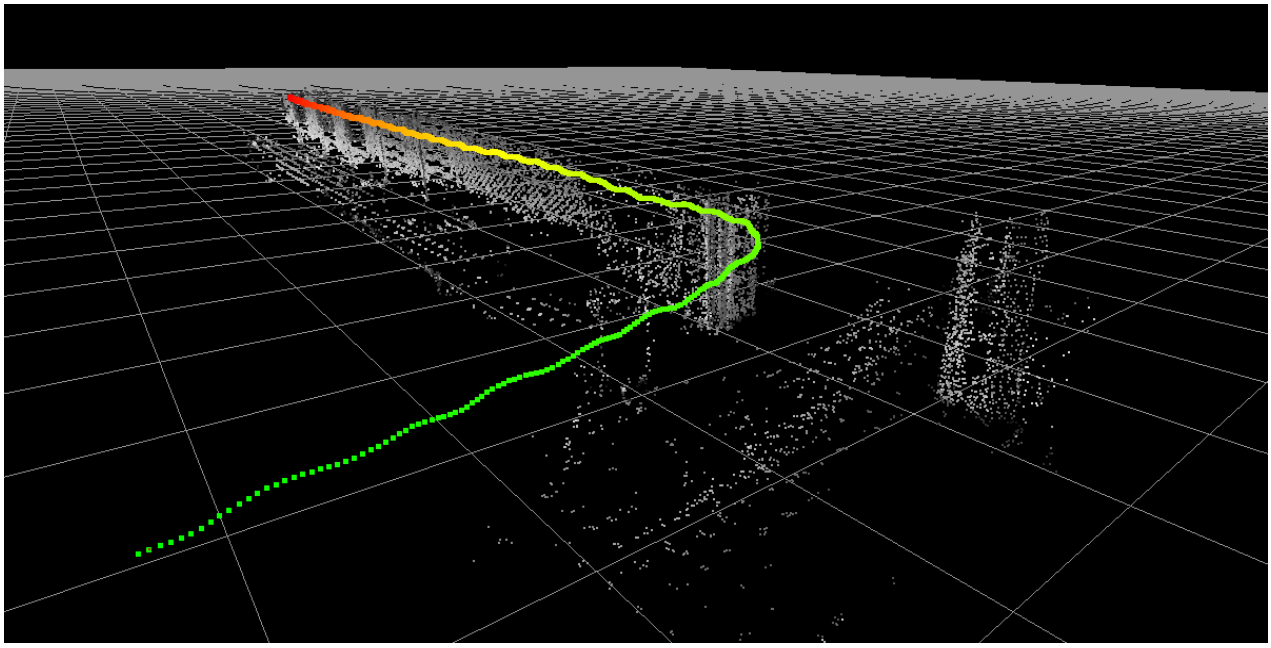
[5] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *International Symposium on Mixed and Augmented Reality*, 2007.

[6] A. Lorusso, D. Eggert, and R. B. Fisher. A comparison

(a) Indoor environment.



(b) Bird's Eye View



(c) Reconstruction observed from a similar viewpoint as in Fig. (a). The 3D points of the structure are shown with their original luminance obtained from the images. The colored points shows the estimated position of the camera at each frame. The color encodes the time of acquisition (from green to red).

Figure 5. Results on the indoor data set.

of four algorithms for estimating 3-d rigid transformations. In *BMVC*, 1995.

[7] X. Lu and R. Manduchi. Detection and localization of curbs and stairways using stereo vision. In *ICRA*, 2005.

[8] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone. Rover navigation using stereo ego-motion. In *Robotics and Autonomous Systems*, volume 43(4), pages 215–229, 2003.

[9] C. Rabe, T. Müller, A. Wedel, and U. Franke. Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In *ECCV*, 2010.

[10] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

[11] G. Sibley, L. Matthies, and G. Sukhatme. Bias reduction filter convergence for long range stereo. In *International Symposium of Robotics Research*, 2005.

[12] J. Sez and F. Escolano. Stereo-based aerial obstacle detection for the visually impaired. In *ECCV Workshop on Computer Vision Applications for the Visually Impaired*, 2008.

[13] A. Talukder and L. Matthies. Real-time detection of moving objects from moving vehicles using dense stereo and optical flow. In *IROS*, 2004.

[14] S. Treuillet, E. Royer, T. Chateau, M. Dhome, and J.-M. Lavest. Body mounted vision system for visually impaired outdoor and indoor wayfinding assistance. In *Conference & Workshop on Assistive Technologies for People with Vision & Hearing Impairments*, 2007.

[15] J. Zhang, S. Ong, and A. Nee. Navigation systems for individuals with visual impairment: A survey. In *International Convention for Rehabilitation Engineering and Assistive Technology*, 2008.