J. ELECTRICAL ENGINEERING, 48 (1997), NO. 1-2, 3–11

3–D SCENE RECONSTRUCTION FROM MONOCULAR IMAGE SEQUENCES

Milan Hanajík * — **Paul P. J. van den Bosch** **

In this paper we deal with the problem of 3-D scene description reconstruction from monocular image sequences. The reconstruction is based on the processing of linear features extracted from the acquired images.

Two reconstruction algorithms are introduced. The first one formulates the problem as the stochastic filtering problem and the algorithm computes the scene description incrementally using the Extended Kalman Filter (EKF). Issues of the computational complexity of the algorithm are addressed and an algorithm with computation time in each iteration step linearly proportional to the number of processed line segments is proposed.

The second algorithm formulates the problem as the global optimization problem, and the maximum likelihood estimate (MLE) of the scene is numerically computed from all images in one batch.

Results achieved with a sample image sequence are given and conclusions are derived.

K e y w o r d s: 3-D structure from motion, Extended Kalman Filter, computational complexity

1 INTRODUCTION

Many applications such as acquisition of CAD models, robot motion planning and object recognition involve recovery of a representation for a three dimensional (3-D) geometrical structure from sensor data. Reconstruction of 3-D scene description from a sequence of images known as *structure from motion* has been a topic of active research in the field of computer vision in recent years. In this paper we deal with problem of 3-D scene description reconstruction from monocular image sequences, where the camera path is unknown. The reconstruction is based on the processing of linear features extracted from the acquired images.

For the structure from motion problem of a scene composed of point features a close form solution was proposed by Tomasi and Kanade in [14]. Their reconstruction method is efficient and robust, however, a drawback of the technique is that it assumes parallel projection, and that it cannot deal with the missing data problem (occlusions, feature extraction errors, *etc.*). The method was later extended for the scaled parallel projection, which is a better model of the perspective projection.

Close form solutions of the structure from motion problem for linear features have been developed by Liu and Huang [10], and Spetsakis and Aloimonos [12]. These methods need at least 13 line features in three frames. In practice these techniques tend to be very sensitive to errors in measurements.

The above mentioned closed form solutions to the structure from motion problem fall into a class of *batch algorithms*, as they process all data simultaneously. Next to these, *incremental algorithms* have emerged, which subsequently process the images of the sequence and incrementally update the scene description. Several incremental algorithms are based on the *Extended Kalman Filter* (EKF) [2, 7, 6].

Recently, Taylor and Kriegman [13] published a batch algorithm which processes line features from a monocular sequence to obtain their 3-D description. They formulate the problem as the problem of objective function minimization and propose a numerical solution to this problem.

In this paper we present two algorithms for the structure from motion problem for line features. The proposed algorithms assume (more accurate) perspective projection, and provide an iterative numerical solution of the problem.

The first of two algorithms presented in section 3 is an incremental algorithm. The scene reconstruction is provided by stochastic filtering using EKF for the entire scene at once. Parameters of the 3-D scene description, and the parameters specifying the camera position and orientation constitute the state of the system to be estimated. Each time an image is captured and processed, the state estimate is updated, this is

* Department of Measurements, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, SK-812 19 Bratislava, Slovakia

** Measurement and Control Group, Faculty of Electrical Engineering, Eindhoven University of Technology, P.O.Box 513, NL-5600 MB Eindhoven, The Netherlands

This work was supported by SION, The Dutch Society for Informatics Research.

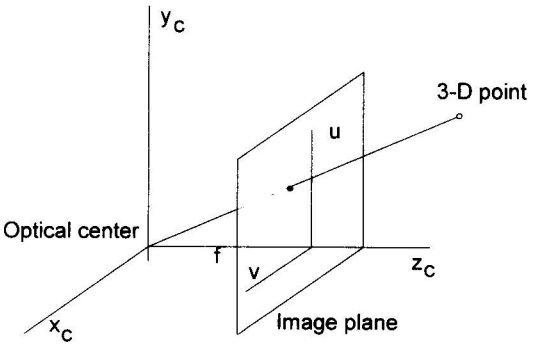

Fig. 1. Camera model.

called a *measurement update*. Between the measurement updates, the state estimate is updated due to the system dynamics (here only the camera motion dynamics), this is called a *time update*.

Issues of the computation complexity are addressed in section 4 where we propose an algorithm with computation time linearly proportional to the number of processed line segments in each iteration step.

The second algorithm introduced in section 5 is a batch algorithm. The scene description and the camera path are computed as a maximum likelihood estimate. Line segments data from all images are processed in a single optimization algorithm.

In the presented algorithms, the knowledge of the camera motion path is not required for scene reconstruction. The camera path is computed as a side result by the techniques. However, a number of reference edges must be present in the viewed 3-D scene to provide some information for the stochastic filtering or batch processing algorithms. These edges must be observed in a few initial images of the sequence.

To provide correct input data, correspondences between line segments in subsequent images have to be established. This is done by tracking line segments in the two-dimensional (2-D) image plane. Predictions of line segments in the next image are computed by Kalman filters, one separate filter for each line segment (section 6). This is another layer of stochastic filtering. Line correspondence is found using stochastic data association.

In section 7, the implementation is described and results achieved with a sample image sequence are given. Finally, the conclusions are derived and further research is proposed in section 8.

2 REPRESENTATION

This section explains parametrization that was used to represent a 3-D scene, the camera, and 2-D line segments contained in the acquired images.

2.1 Scene

The 3-D scene is represented by a set of line segments in the 3-D space. This representation is sufficient, since only straight line segments are extracted from images, and consequently only straight line segments are reconstructed in the 3-D space. Each line segment is specified by a 6-component vector

$$s = (x_b, y_b, z_b, x_e, y_e, z_e)^t, \tag{1}$$

where x_b, y_b, z_b are the coordinates of the segment initial point, and x_e, y_e, z_e are the coordinates of the end point. These coordinates are in the world (absolute) coordinate system. Let N be the number of reconstructed scene segments. The scene is then represented by set $\mathcal{S}$

$$\mathcal{S} = \{s_1, s_2, \ldots, s_N\}, \tag{2}$$

where $s_i, i = 1, \ldots, N$ are vectors specifying the 3-D line segments.

2.2 Camera

Perspective projection onto the image plane is assumed. The optical center of the camera is identical with the origin of the camera centred coordinate system, and the image plane is placed perpendicularly to the z-axis at the distance f from the origin, as it is shown in figure 1. Consequently, the point with coordinates x_c, y_c and z_c in the camera centred system is projected onto the image plane at location (u, v), where u and v are

$$u = f\frac{x_c}{z_c}, \quad v = f\frac{y_c}{z_c}. \tag{3}$$

We use homogeneous coordinates[1] to represent the points in the 3-D space. A point with world system homogeneous coordinates $X_w = (x_w, y_w, z_w, 1)^t$ is transformed into camera centred homogeneous system coordinates $X_c = (x_c, y_c, z_c, 1)^t$ by left multiplication with a 4 × 4 matrix T:

$$X_c = TX_w. \tag{4}$$

The camera position and orientation in 3-D space, *ie* the relation between the world coordinate system and the camera centred system is completely specified by six parameters composing a vector θ:

$$\theta = (\alpha, \beta, \gamma, t_x, t_y, t_z)^t. \tag{5}$$

Consequently, the transformation matrix T is a function of camera parameters θ, $T = T(\theta)$.

[1] Homogeneous coordinates of a point in $\mathbb{R}^3$ is a 4-vector $\boldsymbol{p} = (ax, ay, az, a)^t$, where x, y, z are the coordinates of the point in a common sense, and a is an arbitrary nonzero real number. In our case a is always equal to 1.

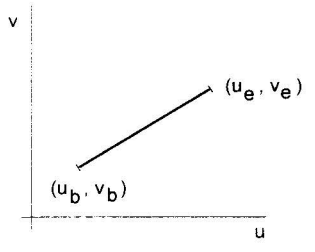

Fig. 2. The line segment parameterized by coordinates of the initial and end points.

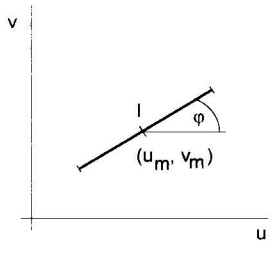

Fig. 3. The line segment parameterized by coordinates of the midpoint, by the length and angle.

Let M be the number of images in the image sequence. Then the sequence (the ordered set) of camera parameter vectors

$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_M\} \tag{6}$$

specifies the path of the camera used to obtain the sequence.

2.3 Line segments in the image sequence

Straight line segments are extracted from the captured grey value images. We have alternatively used two different parameterizations of a line segment:

1. The line segment on the image plane can be represented by image plane coordinates of its initial and end points
$$q = (u_b, v_b, u_e, v_e)^t . \tag{7}$$
2. The line segment on the image plane can be represented by image plane coordinates of its midpoint u_m, v_m (point in the middle of the line segment), its length l and angle φ measured with respect to u axis
$$q = (u_m, v_m, l, \varphi)^t \tag{8}$$

Although each of the two parameterizations have its advantages and drawbacks, the choice of parameterization is not relevant to principles of 3-D reconstruction techniques developed later. In most cases we will only refer to q for the sake of generality.

Let M be the number of images in the image sequence, let N_i be the number of line segments extracted from the i-th image, and let $q_{i,j}$ be the parameter vector of the j-th line segment extracted from the i-th image. The i-th image is represented by the set $\mathcal{Q}_i$

$$\mathcal{Q}_i = \{q_{i,1}, q_{i,2}, \cdots, q_{i,N_i}\} , \tag{9}$$

where $q_{i,j}, j = 1, \ldots, N_i$ specify the line segments. The entire sequence is defined by the set

$$\mathcal{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_M\} . \tag{10}$$

3 SCENE RECONSTRUCTION BY STOCHASTIC FILTERING

The scene and the camera moving along the path can be viewed as a dynamic system. The 3-D reconstruction can be provided by estimation of the state of this system by stochastic filtering. We shall introduce the *state vector* X_t

$$X_t = (S_{1,t}, S_{2,t} \ldots S_{N,t}, \Theta_t, \dot{\Theta}_t)^t , \tag{11}$$

where $S_{i,t}$, $i = 1 \ldots N$, specify 3-D line segments, and $\Theta_t, \dot{\Theta}_t$ are the camera parameters and their time derivatives *ie* components of the camera speed and camera angular speed) at discrete time instants $t = \{t_1, t_2, \ldots, t_M\}$. The state vector (11) is a random vector, thus reflecting the fact that the true values of vectors $s_1, s_2, \ldots, s_n$, θ and $\dot{\theta}$ are not known to us. The probability distribution of the random vector X_t expresses our knowledge of the scene and the camera position at time t. Although we have assumed that the scene is static, and only the camera parameters change over time, random vectors $S_{i,t}$, $i = 1, \ldots, N$ evolve over time as more images of the sequence are processed.

The captured images can be viewed as observations (measurements) of the state vector. We shall define *the measurement vector* as the vector of parameters of particular image line segments:

$$y_t = \left(q_{t,1}, \ldots, q_{t,N_t}\right)^t . \tag{12}$$

The probability distribution of X_t is updated at time instants $t_1, t_2, \ldots, t_M$ when each new image is captured and processed, this is called the *measurement update*. Also, between the subsequent time instants the distribution is updated because of the camera movement, this is called the *time update*. The process of subsequent measurement and time updates constitutes *stochastic filtering*.

We will assume that X_t is normally distributed, $X_t \sim N(\hat{x}_t, P_t)$, and therefore it is fully specified by its mean vector $\hat{x}_t$ and covariance matrix P_t. This assumption considerably simplifies the problem as the probability distribution updates are then identical to the updates of the mean vector and the covariance matrix. Technical details of this process are explained in the following subsections. We will denote by $t-$ the time instant immediately before observing and processing an image at time t, and by $t+$ the time instant immediately after observing and processing the image at time t.

3.1 Measurement update

Line segments found in processed images suffer from random perturbation due to imaging errors, presence of noise, *etc.* Therefore, the measurement vector y_t obtained at time t can be considered to be the outcome of a random variable Y_t. We may assume that the perturbations are additive and

$$Y_t = h(X_{t-}) + \xi_t, \qquad \xi_t \sim N(0, R_t) \tag{13}$$

where $h(X_{t-})$ is an ideal projection of line segments onto the image plane using equations (4) and (3), and the random perturbation ξ_t has a normal distribution with a zero mean and covariance matrix R_t.

The probability distribution of X_{t+} after $Y_t = y_t$ has been observed can be expressed in terms of probability density functions (pdf-s) using Bayes rule

$$f_{X_{t+}}(x) = f_{X_{t-}|Y_t}(x \mid y_t) = \\ = \frac{f_{X_{t-},Y_t}(x, y_t)}{f_{Y_t}(y_t)} = \frac{f_{Y_t|X_{t-}}(y_t \mid x) f_{X_{t-}}(x)}{f_{Y_t}(y_t)}. \tag{14}$$

The density $f_{X_{t-}}(x)$ is the density of X before making the measurement, and it follows from equation (13) that the likelihood function $f_{Y_t|X_{t-}}(y_t \mid x)$ is

$$f_{Y_t|X_{t-}}(y_t|x) = f_{\xi_t}(y_t - h(x)). \tag{15}$$

Due to the fact that the likelihood function (15) is a nonlinear function of state and measurement vectors, the updated distribution of X_{t+} will not be normal. However, it may be approximated by a normal distribution obtained in the following way:

1. A maximum a posteriori (MAP) estimate of state x, that is $x = \hat{x}_{\text{map}}$ maximizing $f_{X_{t+}}(x \mid y_t)$ is computed:

$$\hat{x}_{\text{map}} = \operatorname{argmax}\{f_{X_{t+}}(x \mid y_t)\} = \\ = \operatorname{argmax}\{f_{Y_t|X_{t-}}(y_t \mid x) f_{X_{t-}}(x)\}, \tag{16}$$

2. The nonlinear function $h(x)$ is linearized at point $x = \hat{x}_{\text{map}}$, so that

$$h(x) \approx h(\hat{x}_{\text{map}}) + \nabla_{h(\hat{x}_{\text{map}})}(x - \hat{x}_{\text{map}}), \tag{17}$$

   where the matrix $\nabla_{h(\hat{x}_{\text{map}})}$ is the Jacobian of function $h(x)$ at $x = \hat{x}_{\text{map}}$.

3. Approximation (17) is substituted into (15) and the Bayes rule (14) is then evaluated. This leads to a normal a posteriori distribution $X_{t+} \sim N(\hat{x}_{\text{map}}, P_{t+})$, where the mean vector $\hat{x}_{\text{map}}$ is the MAP estimate obtained from (16), and the covariance matrix is given by

$$P_{t+} = \left[P_{t-}^{-1} + \nabla^t_{h(\hat{x}_{\text{map}})} R_t^{-1} \nabla_{h(\hat{x}_{\text{map}})}\right]^{-1}. \tag{18}$$

It follows from formula (18) that after the measurement update the uncertainty of the state has decreased.

3.2 Time update

Between two measurements, the evolution of the state over time can be modelled in the following way:

$$S_{i,(t+\Delta t)-} = S_{i,t+} \qquad \text{for } i = 1, \ldots, n \tag{19}$$

$$\Theta_{(t+\Delta t)-} = \Theta_{t+} + \Delta t\, \dot{\Theta}_{t+} + \mu_t \tag{20}$$

$$\dot{\Theta}_{(t+\Delta t)-} = \dot{\Theta}_{t+} + \upsilon_t. \tag{21}$$

Equations (19) reflect the fact that the scene is static. Random vectors μ_t and υ_t are the perturbations of the camera position and camera speed, respectively. Equation (20) states that the camera would stay without disturbances in movement with a constant speed and angular speed of its rotation. μ_t is the deviation of the camera position from the expected one, and υ_t is the deviation of the camera speed (and angular speed) from the previous value.

Equivalently, it can be written

$$X_{(t+\Delta t)-} = \Phi_{t+\Delta t,t} X_{t+} + \omega_t, \quad \omega \sim N(0, Q_t), \tag{22}$$

where $\Phi_{t+\Delta t,t} =$

$$= \begin{pmatrix} 1 & \ldots & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \vdots & & \ddots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & \ldots & \ldots & 1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & \ldots & \ldots & 0 & 1 & \ldots & 0 & \Delta t & \ldots & 0 \\ \vdots & & & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & 0 & 0 & \ldots & 1 & 0 & \ldots & \Delta t \\ 0 & \ldots & \ldots & 0 & 0 & \ldots & 0 & 1 & \ldots & 0 \\ \vdots & & & \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 1 \end{pmatrix}. \tag{23}$$

The time update is provided by the following equations

$$\hat{x}_{(t+\Delta t)-} = \Phi_{t+\Delta t,t} \hat{x}_{t+} \tag{24}$$

$$P_{(t+\Delta t)-} = \Phi_{t+\Delta t,t} P_{t+} \Phi^t_{t+\Delta t,t} + Q_t. \tag{25}$$

$\hat{x}_{(t+\Delta t)-}$ and $P_{(t+\Delta t)-}$ are the mean and the covariance of the updated probability distribution of the state at time $(t + \Delta t)-$. It follows from (25) that the uncertainty of the state (namely of the camera parameters in the state vector) has increased.

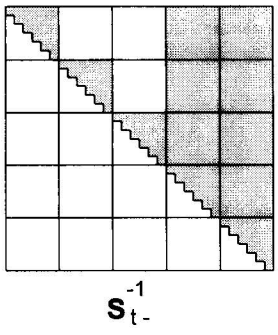

Fig. 4. The structure of the square root inverse of the covariance matrix of the state in the case of three line segments. The matrix size is 30-by-30 elements. White areas represent zero elements.

3.3 State initialization

The *a priori* covariances of the unknown parameters are set to (some) large values. The uncertainty of the few initial camera positions is diminished by the measurement updates, when a known part of the scene is captured. When new lines enter the camera view, their processing is deferred until the next image, when the corresponding line pairs are formed. Initial line parameter estimates are set to values obtained by triangulation, and the initial covariances are also set to large values. By processing subsequent images, the uncertainty of parameters is diminishing.

4 REDUCING COMPUTATIONAL COMPLEXITY

The most time consuming operations involved in stochastic filtering are the maximization problem (16), and the covariance update (18). The dimensions of the state vector, measurement vector and covariance matrices might be very large, for instance for 100 lines the covariance P_t has dimensions 612 x 612 elements.

To avoid computation of matrix inversions and square roots[2], we have used a procedure that updates directly the square root of the inverse of the covariance matrix [11]. During updates, the square roots of the inverses of the covariance matrices are preserved to be square upper triangular matrices, and in our case they will be sparse as well. Equation (18) can be rewritten as

$$S_{t+}^{-t} S_{t+}^{-1} = S_{t-}^{-t} S_{t-}^{-1} + \nabla_{h(\hat{x}_{\mathrm{map}})}^{t} S_R^{-t} S_R^{-1} \nabla_{h(\hat{x}_{\mathrm{map}})} \quad (26)$$

where S_{t+}, S_{t-} and S_R are square roots of covariance matrices as follows: $P_{t+}^{-1} = S_{t+}^{-t} S_{t+}^{-1}$, $P_{t-}^{-1} = S_{t-}^{-t} S_{t-}^{-1}$, and $R_t^{-1} = S_R^{-t} S_R^{-1}$. Notation S^{-t} denotes the transpose of the inverse of matrix S.

There are infinitely many solutions of (26) with respect to S_{t+}^{-1}. One solution which can be readily computed is

$$\tilde{S}_{t+}^{-1} = \begin{pmatrix} S_{t-}^{-1} \\ S_R^{-1} \nabla_{h(\hat{x}_{\mathrm{map}})} \end{pmatrix}. \quad (27)$$

This solution does not have the same (square) dimensions as the matrix S_{t-}^{-1}, however, it can be converted by QR-decomposition into a form

$$\tilde{S}_{t+}^{-1} = Q \begin{pmatrix} S_{t+}^{-1} \\ 0 \end{pmatrix}, \quad (28)$$

where Q is an orthogonal matrix, and S_{t+}^{-1} is a square upper triangular matrix. Hence, S_{t+}^{-1} is the requested solution of (26).

The matrix on the right hand side of equation (27) is a $(10N + 12) \times (6N + 12)$ matrix, where N is the number of processed line segments. The number of floating point operations (FLOPS) needed to QR-factorize this matrix by *eg* Householder transformations [5] is roughly $400\,N^3$. QR-factorization (28) can be performed in time linearly proportional to N, if the sparsity of the matrices is exploited. Consider the structure of matrices on the right hand side of equation (27), S_{t-}^{-1}, S_R^{-1} and $\nabla_{h(\hat{x}_{\mathrm{map}})}$. In the presented algorithm the structure of S_{t-}^{-1} is kept as it is shown in figure 4. The depicted matrix is for the case of three line segments with the size 30-by-30 elements (three times 6 for each line segment plus 12 for camera parameters). The sparse structure of matrix S_R^{-1} (see figure 5) is due to the independence of the observation noise for the different line segments. Similarly, the sparse structure of the Jacobian $\nabla_{h(\hat{x}_{\mathrm{map}})}$ is due to the fact that the central projections onto the image plane are independent of each other for two different line segments. The factorization is performed in N processing steps (see figure 6 for the case of three lines). In each step a submatrix having a fixed size 22-by-18 elements is selected from the entire matrix (as indicated in figure 6 by bold lines), and the submatrix is QR-factorized as it is shown in figure 7. In this way zeros are selectively introduced in the bottom part of the entire matrix, and after N processing steps factorization of the matrix is accomplished. The structure of the top square part of the matrix is identical with the structure of S_{t-}^{-1} (figure 4). The number of FLOPS needed for the computation is roughly $4000\,N$, thus it is linearly proportional to the number of line segments.

Equation (26) is solved (and the above mentioned algorithm is applied) repetitively in each iteration,

[2]The square root of matrix P is matrix S such that $P = SS^t$. If some matrix S is a square root of P, then matrix SQ, where Q is an orthogonal matrix, is also a square root of P.

Fig. 5. The structure of the square root inverse of the covariance matrix of the observation vector, the structure of the Jacobian of the function $h(x)$, and the structure of their product. White areas represent zero elements. The number of observed line segments is three.

Fig. 6. The factorization is performed in steps. In each step a submatrix indicated by bold lines is QR-factorized, and the portions of the matrix are selectively zeroed.

when maximizing the objective function in (16), and also afterwards when computing the covariance matrix update (18). It is not difficult to show that the structure of the square root of the inverse of the state covariance matrix is preserved also when computing the time update, because of the sparse structure of the matrix (23). The overall improvement of the computation time by avoiding matrix inverses, square roots and by exploiting the structure of matrices is more than $(N^2\backslash 10)$-fold, where N is the number of processed line segments.

5 SCENE RECONSTRUCTION BY BATCH PROCESSING OF IMAGES

The estimate $\hat{x}_{t_i}$, computed by stochastic filtering is based on images observed at time instants $t_1, t_2, \ldots, t_i$, ie the images observed till time t_i. We will denote this estimate by $\hat{x}_{t_i|t_1,\ldots,t_i}$. When processing the image sequence off line, better results can be obtained by computing the estimate based on all measurements and denoted by $\hat{x}_{t_i|t_1,\ldots,t_M}$. This is known as stochastic interpolation.

Furthermore, the entire idea of the dynamic system can be abandoned and the 3-D reconstruction can be performed as a parameter estimation, where the parameters to be estimated constitute the 3-D scene and the complete camera path. In this way all images are processed in one parameter estimation process, in one batch. In the sequel a technique based on a maximum likelihood estimation is developed.

We shall introduce the parameter vector Θ

$$\Theta = (s_1, \ldots, s_N, \theta_1, \ldots, \theta_M)^t, \tag{29}$$

where $s_i, i = 1 \ldots N$, specify the 3-D scene, and $\theta_i, i = 1 \ldots M$ specify the camera path.

The vector of measurements is defined as

$$y = (y_1^t, \ldots, y_M^t)^t, \tag{30}$$

where $y_i, i = 1, \ldots, M$ are the measurements extracted from each particular image given by (12). The likelihood function $f_{Y|\Theta}(y|\Theta)$ is a product of several independent terms:

$$f_{Y|\Theta}(y|\Theta) = \prod_{i=1}^{M} f_{Y_i|s,\theta}(y_i|s,\theta_i)\ f_\theta(\theta). \tag{31}$$

Terms $f_{Y_i|s,\theta}(y_i \mid s,\theta_i)$ are the likelihoods of image processing for each image of the sequence and have

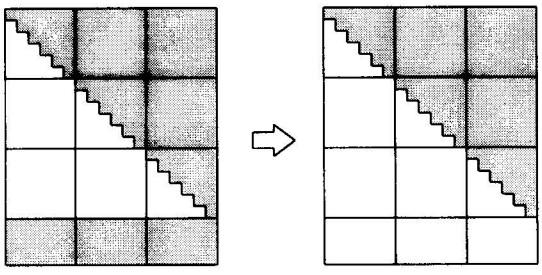

Fig. 7. QR-factorization of the fixed size (22-by-18 elements) submatrix.

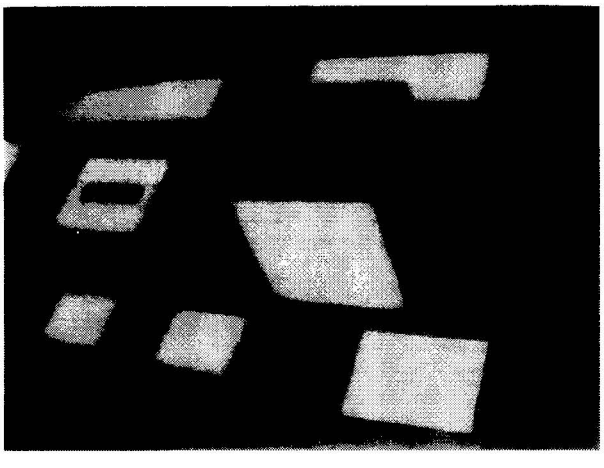

Fig. 8. Image of the scene.

been discussed already in section 3.1. The term $f_\theta(\theta)$ is the probability density of the camera path. The estimate of the parameter vector is obtained by the maximization of the likelihood

$$\hat{x} = \arg\max_{x \in \Theta} f_{Y|X}(y \mid x), \tag{32}$$

which leads to a minimization problem that can be solved numerically using the Gauss-Newton method.

Batch processing presents an increase in the number of parameters and measurements processed in one step when compared with the number of state variables and measurements in each update of stochastic filtering, and computational complexity problems have to be addressed in a similar way as in section 4.

6 TRACKING LINE SEGMENTS

The correspondence between line segments in subsequent images is provided by tracking 2-D line segments in the 2-D image plane. If the time difference between images from a smoothly moving camera is sufficiently small, the movement of line segments on the 2-D image plane is smooth too. This can be used to predict the next position of the 2-D line segment by a Kalman filter (see also [3] and [4]). A state vector of a 2-D line segment contains information about the line segment's position, velocity and acceleration. The predictions of the line segment's position in the next image are made using an assumption that 2-D lines are moving with a constant acceleration. An actually observed line segment is assigned to the predicted one, if the *Mahanalobis distance* between the predicted and the observed line is smaller than some threshold.

7 IMPLEMENTATION AND RESULTS

7.1 Implementation

The 3-D scene reconstruction has been implemented in Matlab, a numeric computation and visualization software package. Both approaches, stochastic filtering and batch processing, have been implemented and tested. The 3-D reconstruction is performed off line and requires as an input the measurement vectors extracted from the images and the correspondence between the line segments in the images. For the 3-D reconstruction to converge correctly, there must be some reference 3-D line segments imaged in a part of the sequence, *eg* in the first two or three images. The parameters (the 3-D coordinates) of the reference line segments are also supplied as an input.

7.2 Acquisition and preprocessing

The 3-D scene reconstruction has been tested with an image sequence of a scene acquired in our laboratory. The sequence has been acquired by a black-and-white CCD camera which was attached to a carriage moving on a rail along a straight path. The rail was placed slantwise above the scene, so that objects in the scene were about 4 m far from the camera. The images have been taken each 0.5 m of the camera trajectory. The number of acquired images was 8. The camera was also rotating around its axis with the speed 2 degrees per image. As a consequence, the scene was followed by the camera, and scene objects stayed longer in the camera field of view when compared with a sequence without camera rotation. The resolution of images was 512 by 512 pixels, the quantization was 256 grey value levels.

From the acquired images, straight line segments were extracted in the following way:

1. First, edges were detected with a Lee edge detector [8, 9]. Since the Lee edge detector is a special case of Canny edge detector and is capable to detect the image intensity changes along one direction, the

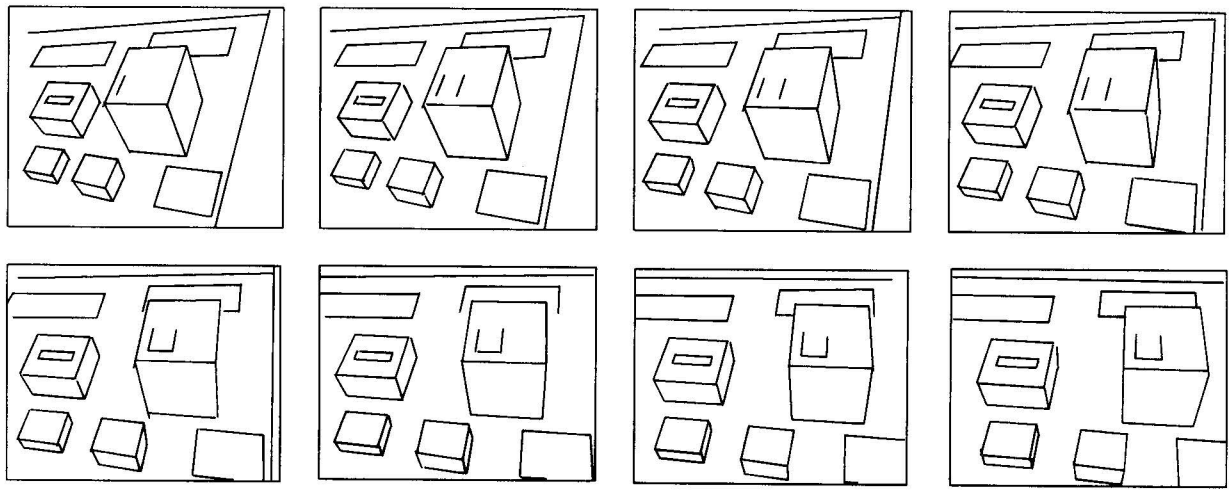

Fig. 9. Sequence of eight images. Line segments were extracted from intensity images.

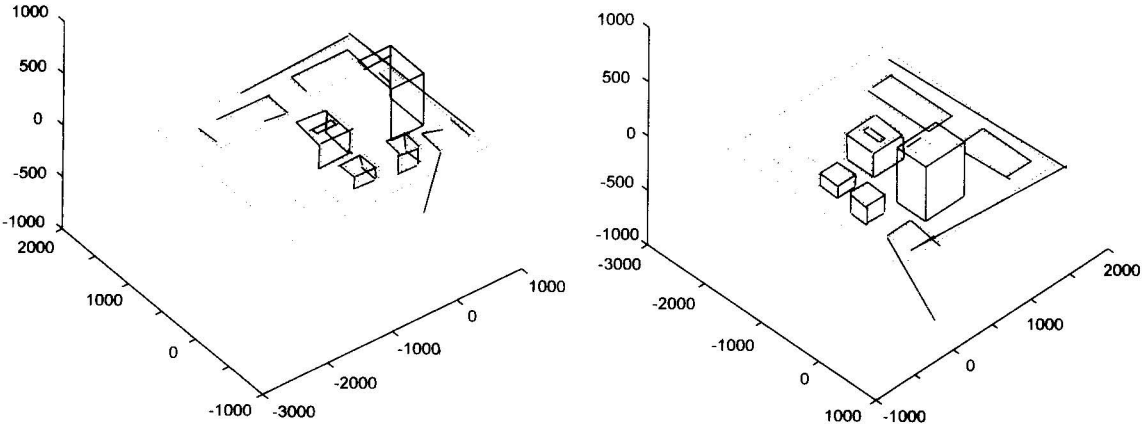

Fig. 10. The result of the 3-D scene reconstruction depicted from two different viewpoints. Solid lines are the reconstructed lines, and dotted lines are the line segments of the actual scene.

detector has been applied twice, in horizontal and in vertical directions.

2. Straight line segments have been extracted by search for chains of edge points in the pair of edge detected images. The applied straight line segment extraction algorithm has been described in [1] and resembles the algorithm of Deriche [4]. The parameters of the extracted line segments are inputs of the 3-D reconstruction algorithms.

Then, correspondences between line segments in different images have been established to eliminate the effect of errors of the line tracking algorithm.

In figure 8 a grey value image of the scene is shown. In figure 9 the extracted sequence of line images is shown.

7.3 3-D reconstruction results

The ground through data (dimensions) of the scene were available. The edges of the largest cube in the scene were given accurately and served as reference lines. The reconstructed 3-D lines are shown in figure 10, together with the lines of the actual scene. The elements of the covariance matrix of the state were decreasing rapidly, the standard deviations of line parameters were below 50 mm after processing the images. However, a few lines were not estimated correctly (notice *eg* the bottom line of the white rectangle in the bottom right corner of the image). The large perturbation is caused by occlusions at the edge of the image which occur in the last 4 images of the sequence, in combination with the fact that the direction of the camera movement is parallel with the line segment direction. For such lines the technique is less accurate.

8 CONCLUSIONS AND FUTURE WORK

In this paper the problem of 3-D scene reconstruction from a monocular image sequence is treated. The

problem is viewed as a problem of unknown parameters estimation. Two techniques are presented: the first one estimates the unknown parameters incrementally by stochastic filtering using EKF, and the second one estimates the parameters in a single batch as an MLE estimate.

We have proposed an algorithm which reduces the computation time linearly with the number of line segments in the images, in each iteration step. As a consequence, it should be possible to implement the first of the techniques in real time (this technique is incremental and processes the images one at a time).

In the derivations we have made the assumption that the distribution of random perturbations can be approximated by a jointly normal distribution. This assumption is valid when the perturbations are caused by inaccuracy of image processing and feature extraction, however it is violated when the perturbations are due to occlusions in the scene or at the edge of the image plane. Such occlusions lead to rude estimation errors, and their detection would improve the results.

The application of the techniques is not limited to the processing of line segments, and potentially any features which can be parameterized and extracted from images can also be 3-D reconstructed by the techniques.

REFERENCES

[1] BROERTJES, R. A. J.: Extracting straight lines and elliptic arcs from edge images, Master's thesis, Eindhoven University of Technology, 1993.

[2] CROWLEY, J.—STELMASZYK, P.: Measurement and integration of 3D structures by tracking edge lines, In: Proc. First European Conf. Computer Vision, 1990, pp. 269–280.

[3] CROWLEY, J.–STELMASZYK, P.–DISCOURS, C.: Measuring image flow by tracking edge-lines, In: ICCV 88: 2nd International Conf. on Computer Vision, 1988, pp. 658–664.

[4] DERICHE, R.—FAUGERAS, O.: Tracking line segments, Image and Vision Computing 8 No. 4 (1990), 261–270.

[5] GOLUB, G. H.—VANLOAN, C. F.: Matrix Computation, The Johns Hopkins Univ. Press, second edition, 1989.

[6] HONG, L.—BRZAKOVIC, D.: 3D scene reconstruction from noisy image sequences using data fusion, Control Engineering Practice 2 No. 5 (1994), 825–831.

[7] JEZOUIN, J.-L.—AYACHE, N.: Three-dimensional structure from a monocular sequence of images, Technical report, INRIA, Domaine de Voluceau, Rocquencourt, B.P. 150, 78153 Le Chesnay, France, 1990.

[8] LEE, D.: Coping with discontinuities in computer vision: Their detection, classification and measurement, In: IEEE 2nd International Conference on Computer Vision, Tampa, Florida, USA, 1988, pp. 546–557..

[9] LEE, D.: Edge detection, classification, and measurement, In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1989, pp. 2–10..

[10] LIU, Y.—HUANG, T. S.: A linear algorithm for motion estimation using straight line correspondences, Computer Vision, Graphics & Image Processing 44 No. 1 (1988), 35–57.

[11] MAYBECK, P. S.: Stochastic Models, Estimation and Control, vol. 1, Academic Press, 1979.

[12] SPETSAKIS, M. E.—ALOIMONOS, J.: Structure from motion using line correspondences, International Journal of Computer Vision, 4 No. 3 (1990), 171–184.

[13] TAYLOR, C. J.—KRIEGMAN, D. J.: Structure and motion from line segments in multiple images, Technical report, Center for systems science, Dept. of Electrical Engineering, Yale University, New Haven, CT, 1994.

[14] TOMASI, C.—KANADE, T.: Shape and motion from image streams under orthography: A factorization method, International Journal of Computer Vision 9 No. 2 (1992), 137–154.

Received 5 October, 1996

Milan Hanajík (Dr, Ing), was born in Bratislava, Slovakia, in 1962. He received Ing (MSc) degree in Electrical Engineering from the Slovak University of Technology, Bratislava in 1986, and Dr (PhD) degree from the Eindhoven University of Technology, Eindhoven, the Netherlands in 1995. His research interests include 3-D computer vision and image processing.

Paulus Petrus Johanes van den Bosch (Prof, Dr, Ir), was born in Rotterdam, the Netherlands in 1948. He received his Ir (MSc) and Dr (PhD) degrees in Electrical Engineering from the Delft University of Technology, Delft, the Netherlands in 1972, and 1983, respectively. From 1988 until 1993 he was appointed as a full professor at the Control Engineering Group at the Delft University of Technology, and from 1993 as the full professor and the head of the Measurement and Control Group at the Eindhoven University of Technology, Eindhoven, the Netherlands.