Visual Odometry Based on Gabor Filters and Sparse Bundle Adjustment

Bogdan Kwolek

Abstract— **The optimal way for recovering motion and structure from long image sequences is to use Sparse Bundle Adjustment. The objective of this work was to elaborate a method yielding good initial estimates of the pose for SBA based pose refinement. A new approach for determining inter frame correspondences between features is presented. It is based on Singular Value Decomposition of weighted cross correlation matrix of two feature sets. The weighting of potential matches between features is realized on the basis of cross correlation of intensity values and Gobor filter responses. The method is robust to large motions. Corner and SIFT features were used to compare the effectiveness of our method with Kalman filter based tracking of features. Initial estimate of the pose is determined with Singular Value Decomposition and quaternion based representation of camera's pose. The refinement of the pose estimate is achieved using RANSAC and then SBA on several consecutive frames at once. Experimental results demonstrate the capability of the system to estimate visual odometry in real-time.**

I. INTRODUCTION

A considerable amount of work has been done in the area of using vision to localize a robot. The localization of the robot can be considered as coarse place recognition based on color histograms for familiar locations [6], and also as a problem of determining a set of local features allowing recognition of specific location [24].

In this paper we deal with a promising method, referred to as visual odometry. The goal is to provide a reliable estimate of robot motion from visual data only [1]. The key idea of visual odometry is estimating the motion of the robot through tracking the visually selected landmarks by an onboard camera. The method accumulates error over time like dead-reckoning, but it has been proven that it provides more accurate results [5].

The growing interest in visual odometry is due to several reasons. Video sensors permit the mobile robot to self-localize during performing other tasks such as people detection, understanding and prediction their intentions and actions, detecting and avoiding obstacles. At the same time, a huge information about the environment can be captured for exploration and map construction. Video sensors are more flexible and less expensive than laser scanners, widespread utilized in SLAM [4]. The problem of motion estimation is one of the crucial issues in SLAM.

A number of visual odometry algorithms has been proposed recently. The algorithms are based either on single camera [3][1] or stereo vision [5][1]. The approaches mainly differ in feature matching and method applied for estimating the camera's motion. For example, in [3] a Kalman filter is used for sequential estimation of the motion parameters, based on a calibrated camera and a maintenance of salient features in a scene. The estimates of the pose using Sparse Bundle Adjustment usually have a smaller error variance than Kalman filter based methods. The work [5] employs preemptive RANSAC [23] in estimation of the visual motion, which is followed by an iterative refinement. Matching of the features is achieved on the basis of cross correlation between intensities of pixels. However, such a matching can lead to very large number of false matches, especially when the overlap between consecutive images is small. On the other hand, consistency of the feature correspondence has substantial influence on errors of the estimates. When inter frame overlap is small, false matches between features can lead to break down of the whole system.

In most cases, correspondences between features can not be reliably extracted using low-level image cues. Phase of Gabor filter responses has been used in determining disparity in a stereo pair [8]. In our work we utilize the phase as the main cue in determining the correspondence. The principal difference with the existing work is that we built the correspondence using information from all features in the images. In our approach we employ cross correlation between location of two feature sets, cross correlation between pixel intensities and phase. Singular Value Decomposition of combination of cross correlation matrixes allows us to determine correspondences and admits one-to-one matchings only. The resulting algorithm outperforms existing algorithms for inter frame feature matching, especially in case of significant change of overlap between consecutive images. This paper also shows the viability of our approach for estimation the visual odometry.

The remainder of the paper is ordered as follows. First, we briefly describe the extraction of features. Then the tracking of features is explained in Section III. It begins with an outline of correspondence estimated with the usage of SVD and proximity. Then it shows how to integrate in this method the cross correlation between pixel intensities. Finally, it explains how the Gabor filter responses can be utilized in reliable matching of features. Section IV is devoted to initial pose estimation using SVD and quaternion based representation of camera rotation. Then we present Sparse Bundle Adjustment based pose estimation, where our algorithm plays significant role. In section VI we present all ingredients of our system and report results. Finally, some conclusions follow in the last section.

B. Kwolek is with Computer and Control Engineering Chair, Faculty of Electrical and Computer Engineering, Rzeszów University of Technology, W. Pola 2, 35-959 Rzeszów, Poland bkwolek@prz.rzeszow.pl

II. Extraction of features

The method for camera pose estimation presented in this work involves three main stages: 1) feature selection, 2) feature tracking, and 3) motion estimation. In the remainder of this section the first stage of our method is discussed.

The algorithm operates on images acquired from a stereo pair. A dense disparity map is generated on the basis of the SRI algorithm [13] in order to determine the 3D coordinates of the extracted features. The algorithm is based on area-based matching, followed by a post-filtering. The post-filtering utilizes a combination of a confidence filter and left-right checking to discard patches with insufficient texture that are the main source of false matches. Features are extracted only in the left image. Features with no depth information are discarded.

The Harris corner detector [14] is one of the most frequently used feature detectors. It is a very stable operator and it is able to extract the interest points of the same object's detail in two or more images, even when the camera was moved between the shots. A description of algorithm and pseudo code of detector can be found in [15]. Figure 1b depicts corners detected in an example image.

The Scale Invariant Feature Transform (SIFT) is invariant to translation, scaling, and rotation [16]. It is also partially invariant to illumination variations as well as affine for 3D projection. It selects a large number of stable features over a large range of locations and scales.

The algorithm consists of four main stages. The first one finds scale-space extrema using a Difference of Gausian to extract interest points. The second stage that is called keypoint localization aims at determining the location and scale of each candidate point on the basis of measures of stability. In next stage the orientation assignment takes place and one or more orientations are assigned to each keypoint using local gradients. In the last stage a keypoint descriptor is generated via local gradients at the scale found in stage two. The resulting feature descriptor contains 128 elements, which captures the orientation information of local image region. Figure 2c depicts location of detected features, whereas Fig. 2d illustrates orientations and scales of features.

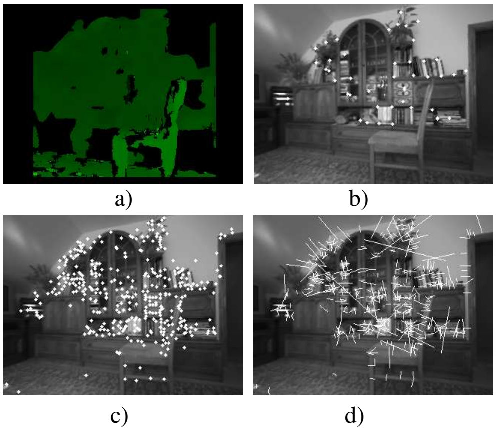
a) b) c) d)

Fig. 1. Depth image (a), corners (b), SIFT features (c), orientation and scale of SIFT features (d).

III. Tracking of features

At the beginning of this section we outline an algorithm proposed by Scott and Longuet-Higgins for determining corresponding features on the basis of Singular Value Decomposition [7]. The section explains also how we incorporated the cross correlation and phase based matching into this framework.

A. *Correspondence based on SVD and proximity*

The correspondence problem consists in finding a pair of pixels or features in two or more views of the scene such that each element in the considered pair corresponds to the same scene point. Due to combinatorial complexity and ill-posedness, finding an acceptably good correspondence in sequence of images is one of the hardest low-level tasks in image analysis. The correspondence between pixels or features is utilized in most optical flow algorithms and stereovision-based computation of depth. Motion can be extracted on the basis of correspondence between temporally consecutive images. Stereovision-based depth can be extracted from images taken possibly simultaneously from spatially arranged cameras. Area-based approaches applied in the correspondence problem usually rely on some kind of statistical correlation between local regions in a stereo-pair.

There are two general approaches to the feature correspondence problem. In the first one the correspondences are sought in the second image using multi-scale techniques. In the second group of methods the features are detected independently and then matched by some kind of relaxation. For example, normalized cross-correlation is used in work [1] to match features between pairs of frames. Each feature in previous image is matched to every feature within a fixed distance from it in the current image. Mutual consistency check is utilized in evaluating potential matches. However, this simple approach produces very large number of false tracks.

To improve the performance of correlation based inter-frame matching we propose an approach which consists in multiplying each element of the cross correlation matrix by a Gaussian weighted distance between features. Using such a correspondence matrix we perform Singular Value Decomposition (SVD) within framework that has been proposed in [7]. In a more sophisticated version of the algorithm we additionally take advantages of multi-scale analysis and utilize the correspondence probability distributions. The correspondence probability distributions are computed on the basis of Gabor filters.

The method proposed in [7] first generates a pairwise proximity matrix G between all features. Each element G_{ij} is a distance between two features i and j, weighted by a Gaussian. Such a Gaussian weighted distance is computed according to formula: $G_{ij} = e^{-d_{ij}^2/2\sigma^2}$, where d_{ij} is Euclidean distance between features i and j. Small value of σ permits local interactions between features, whereas a larger value can be used to achieve more global interactions. In the next step of the algorithm the SVD of this matrix is calculated: $G = USV'$, where U and V are orthogonal matrices and S with nonnegative diagonal elements is the same dimension as G. The diagonal values in S are set to 1 in the next step. Once multiplied the matrices back together, we are given a scoring matrix P between pairwise correspondences. This matrix has the interesting property of selecting good pairings. The squares of elements in each row of this matrix add to 1 [7]. This implies that a feature i cannot be associated with more than one feature j. Similar to cross-correlation based approach that was presented in [1], the algorithm permits feature exclusion as only one-to-one matchings are possible. If P_{ij} is the greatest element in its row and simultaneously the greatest element in its column, then the features i and j are in correspondence [7]. The original algorithm relies only on distances between features and does not consider similarities between features.

B. Matching by SVD using cross-correlation and proximity

In order to embed into the algorithm the similarity between the features we determine the elements of matrix G as follows:

$$G_{ij} = C_{ij} e^{-d_{ij}^2/2\sigma^2} \tag{1}$$

where C_{ij} is a factor expressing similarity between features. It is determined on the basis of the normalized cross correlation and takes values between 0 for uncorrelated patches and 1 for indistinguishable patches. The normalized cross-correlation between two $W \times W$ blocks of pixel intensities I and T is given by [15]:

$$C_{ij} = \frac{\sum_{\{x,y\} \in W} (I_{x+i,y+j} - \overline{I}_{i,j})(T_{x,y} - \overline{T})}{2\, W^2 \sigma_I \sigma_T} + \frac{1}{2} \tag{2}$$

where $\overline{I}$ and $\overline{T}$ are averages, σ_I and σ_T are the standard deviations. Such correspondence measure is more discriminative than the measure that is only based on proximity. The SVD algorithm for feature correspondence using proximity and cross correlation still admits one-to-one matchings only.

Figure 2a shows correspondences that were determined between images with large overlap. The cross correlation was calculated in windows of size 11×11. It can be observed that in case of small change of the camera pose all matchings are correct. If overlap between images is relatively small there are false matches, see Fig. 2b.

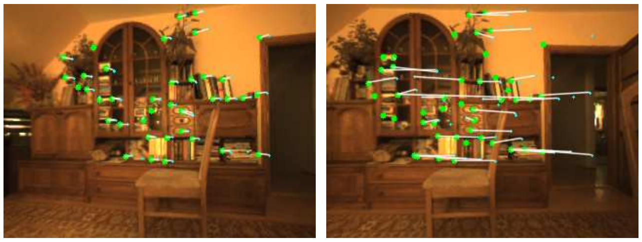
a) b)

Fig. 2. Inter-frame feature matching using SVD, cross-correlation and proximity

C. SVD based matching using proximity, cross-correlation and phase

In work [8] Gabor filters are employed to determine correspondence in a stereo pair. The method assumes that corresponding points have nearly the same local phase. The choice of Gabor filter responses is biologically motivated since they model the response of human visual cortical cells [12]. The main advantage of Gabor wavelets is that they allow analysis of signals at different scales, or resolution, and further they accommodate frequency and position simultaneously. Gabor filters remove most of variation in lighting and contrast. They are also robust against small shifts and small object deformations. The Gabor wavelet is essentially a sinewave modulated by a Gaussian envelope. The 2-D kernel of Gabor filter is defined in the following manner [15]:

$$f(x,y,\theta_k,\lambda) = \exp\left[-\frac{1}{2}\left\{\frac{R_1^2}{\sigma_x^2} + \frac{R_2^2}{\sigma_y^2}\right\}\right] \exp\left\{i\frac{2\pi R_1}{\lambda}\right\} \tag{3}$$

where $R_1 = xcos\theta_k + ysin\theta_k$ and $R_2 = -xsin\theta_k + ycos\theta_k$, σ_x and σ_y are the standard deviations of the Gaussian envelope along the x and y dimensions, λ and θ_k are the wavelength and orientation of the sinusoidal plane wave, respectively. The spread of the Gaussian envelope is defined in terms of the wavelength λ. θ_k is defined by $\theta_k = \frac{\pi(k-1)}{n}$, $k = 1, 2, ..., n$, where n denotes the number of orientations that are taken into account. For example, when $n = 4$, four values of orientation θ_k are used: 0^o, 45^o, 90^o, and 135^o.

A Gabor filter response is achieved by convolving the filter kernel given by (3) with an image. The response of the filter for sampling point (x, y) is defined as follows:

$$g(x,y,\theta_k,\lambda) = \sum_{u=-(N-x)}^{N-x-1} \sum_{v=-(N-y)}^{N-y-1} I(x+u, y+v) f(u,v,\theta_k,\lambda) \tag{4}$$

where $I(x, y)$ denotes a $N \times N$ grayscale image.

In this work four different orientations and four different wavelengths have been utilized. The Gabor filter responses were used to locally measure the phase. In contrast to work [8] we are not interested in determining the highest score for the features being in the correspondence, but in determining a weighting factor to express degree of similarity between potential matches. Such a weighting factor should provide more discriminative power for (1) and reflect the appropriate probabilities as the cross-correlation can do. The simplest way to achieve this goal is to use a Gabor filter with orientation θ and scale λ to extract the phase $\phi_{\theta,\lambda}$ of features i and j and then to compare the considered features according to: $\exp(- \mid \phi_{\theta,\lambda}(i) - \phi_{\theta,\lambda}(j) \mid)$. Using the phase of all filters we obtain the following correspondence measure:

$$G_{ij} = c \prod_{\theta,\lambda} \exp(- \mid \phi_{\theta,\lambda}(i) - \phi_{\theta,\lambda}(j) \mid) \tag{5}$$

where c is a normalization constant ensuring that G_{ij} varies between 0 and 1.

Figure 3 demonstrates some phase-based matching results. Figure 3a shows a feature from Fig. 1 that has been matched falsely using SVD based matching built on proximity and correlation. Figure 3c depicts correspondence probability between the marked feature and pixels from Fig. 3b. It illustrates the Gabor wavelet's capability to match features if camera rotates substantially. It should be noted that in most our experiments the probability distribution was something less discriminative.

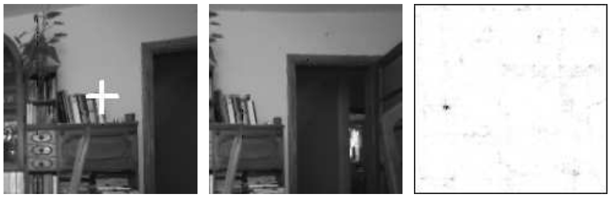
a) b) c)

Fig. 3. Matching using Gabor filter responses. Feature that undergoes matching (a), image for matching (b), probability image of the correspondence between the marked feature and the image in the middle (c)

Figure 4 illustrates sample results that were obtained in SVD-based matching using proximity, cross correlation and responses of Gabor filter. An improved matching performance can be observed considering algorithm that was discussed in previous section, see also Fig. 2.

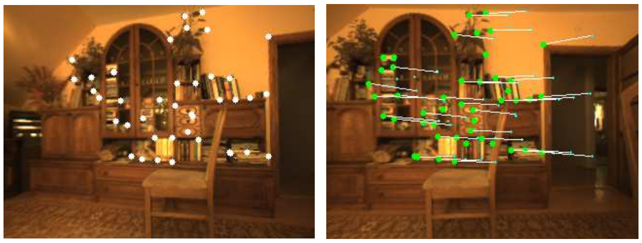
a) b)

Fig. 4. SVD-based inter-frame feature matching using proximity, cross correlation and Gabor filter responses. Features in previous image (a), features in current image with inter-frame correspondences (b)

IV. Estimation of pose of stereo head using inter-frame correspondence and quaternions

Estimating the rigid motion transformation between two 3D point clusters is a fundamental problem in visual odometry as well as in 3D scene reconstruction. In our approach the position of features in 3D is determined using a calibrated stereo camera. The triangulated 3D locations are affected by errors along the line of sight. The noise increases with the distance between the considered locations of features and the stereo camera. The SVD algorithm built on quaternion based representation of the rotation, which we utilize in this work in determining the rough pose of the camera, can produce pose estimates with a significant bias. This can take place even when a large amount of features is in disposal. The method can yield optimal estimates when the 3D measurements are only affected by i.i.d noise. Therefore, we utilize this method to determine rough estimates of the camera pose for Sparse Bundle Adjustment based method. The quaternions were introduced by Horn in [17]. A rigid motion estimator based on Singular Value Decomposition of the cross-correlation matrix of the two data sets was proposed in work [2]. The method first eliminates the translation component by centering the data about the mean values and next estimates the rotation matrix $\hat{R}$. When the rotation matrix is determined the translation $\hat{t}$ is calculated.

Assume that we have in disposal n noise-free and matched 3D measurements $U = \{u_1, u_2, ..., u_n\}$ and $V = \{v_1, v_2, ..., v_n\}$. These ideal values satisfy the rigid motion constraint $v_i = Ru + t$, where R is the 3×3 rotation matrix and t is the translation vector. The rotation can be parameterized by quaternions $q = [q_0,\ q_1,\ q_2,\ q_3]^T$, which are four dimensional unit vectors. The rotation can then be estimated on the basis of the eigenvector corresponding to smallest eigenvalue of the following cross-correlation matrix:

$$M = \sum_{i=1}^{n} Z_i^T Z_i. \tag{6}$$

where the matrix Z_i is computed according to the following formula:

$$Z_i = \begin{bmatrix} \tilde{v}_{1i} - \tilde{u}_{1i} & 0 & -\tilde{v}_{3i} - \tilde{u}_{3i} & \tilde{v}_{2i} + \tilde{u}_{2i} \\ \tilde{v}_{2i} - \tilde{u}_{2i} & \tilde{v}_{3i} + \tilde{u}_{3i} & 0 & -\tilde{v}_{1i} - \tilde{u}_{1i} \\ \tilde{v}_{3i} - \tilde{u}_{3i} & -\tilde{v}_{2i} - \tilde{u}_{2i} & -\tilde{v}_{1i} + \tilde{u}_{1i} & 0 \end{bmatrix}$$

and $\tilde{u}_i = u_i - \tilde{u}$, $\tilde{v}_i = v_i - \tilde{v}$, $\tilde{u} = \frac{1}{n}\sum_{i=1}^{n} \tilde{u}_i$, $\tilde{v} = \frac{1}{n}\sum_{i=1}^{n} \tilde{v}_i$. The estimate of rotation $\hat{R}$ is calculated on the basis of the following equation:

$$\hat{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{7}$$

where $r_{11} = q_0^2 + q_1^2 - q_2^2 - q_3^2$, $r_{12} = 2(q_1 q_2 - q_0 q_3)$ $r_{13} = 2(q_1 q_3 + q_0 q_2)$, $r_{21} = 2(q_2 q_1 + q_0 q_3)$, $r_{22} = q_0^2 - q_1^2 + q_2^2 - q_3^2$, $r_{23} = 2(q_2 q_3 - q_0 q_1)$, $r_{31} = 2(q_3 q_1 - q_0 q_2)$, $r_{32} = 2(q_3 q_2 + q_0 q_1)$, and $r_{33} = q_0^2 - q_1^2 - q_2^2 + q_3^2$. Finally, the estimate of translation $\hat{t}$ is computed in the following manner: $\hat{t} = \tilde{v} - \hat{R}\tilde{u}$.

The correspondence algorithm we utilize in pose estimation can sporadically generate mismatches. Least squares methods use as many data points as possible to increase the influence of outliers. The RANSAC (RANdom SAmple Consensus) algorithm [9] in contrast starts with as little amount of data as possible to fit a model and increases the subset of points during operation. All points that are consistent with the model are called inliers, whereas the non-consistent points are discarded. In our approach we start the RANSAC with stable features that had been successfully tracked in the last frames and in several iterations we discard wrong tracks. The number of POSIT iterations to find the true model parameters with desired probability is determined on the basis of work [21]. In order to further reduce the pose error we employ Sparse Bundle Adjustment on resulting inlier points of several consecutive frames at once.

V. Pose estimation using Sparse Bundle Adjustment

The optimal way for recovering motion and structure from long sequences is to use sparse Bundle Adjustment [10]. Bundle Adjustment is a non-linear optimization problem that is solved through iterative non-linear least square methods. It can be utilized at a refining stage of estimation and the algorithm requires a good initial estimate of the pose. This algorithm has been chosen for our experiments with camera's pose estimation because the estimates of the pose usually have a smaller error variance than Kalman filter based methods. When successive images are dominated by short tracks between features the global optimization process, can degenerate and trap in local minima. The work [10] demonstrates also that the usage of triplets of images rather than pairs improves robustness. Recently an efficient solution to the BA problem has been proposed in [11].

Consider a set of n points X_j in 3D world coordinates is observed from of m cameras with projection matrices P_i. Let $x_{ij} = P_i X_j$ be a projection of the i-th point on the image j. Bundle adjustment refines a set of initial projection matrices P_i and coordinates X_j that most accurately predict the locations of the observed n points in the m images. The optimization problem involves simultaneous refinement of the 3D structure and viewing parameters (i.e. camera's pose and possibly intrinsic camera's parameters) and aims to obtain a reconstruction that is optimal under certain assumptions regarding the noise belonging to the image features. The method finds the set of parameters so that the following squared reprojection error takes a minimal value:

$$\min_{P_i, X_j} \sum_{i=1}^{n} \sum_{j=1}^{m} d\left(P_i X_j, x_{ij}\right)^2 \tag{8}$$

where $d(x, y)$ represents the Euclidean distance between image points x and y. Such cost function guarantees that the estimated camera motions are consistent with each other in the sequence utilized in optimization. The optimization problem is over a large dimensional space. But the unknown 3D point structures are independent from each other. This results in a sparse structure of the Jacobian of the objective function. Very great savings of computational time can be achieved by taking the advantages of sparseness in optimization process. However, such algorithm is quite complex. The implementation [18] that has been elaborated quite recently takes the advantages of sparseness and enables solving huge optimization problems within seconds on a typical PC. It exploits sparseness by utilizing a tailored sparse variant of the Levenberg-Marquardt algorithm.

In our experiments the camera's pose is parameterized by quaternions. The number of parameters expressing a single camera pose is equal to 7. That means that for a sequence consisting of three images, where each contains 16 features, the total number of parameters to be estimated is equal to 69. A typical computation time on 2.4 GHz P IV for such a sequence of images is 0.07 sec., which makes this algorithm very useful for time-critical applications for pose estimation.

VI. Experiments

To test the proposed method of feature tracking we performed various experiments on real images. We compared our method with Lucas-Kanade algorithm for computation of optical flow in pyramids [19], see Fig. 5. The algorithm finds the flow with sub-pixel accuracy. However, as we can observe, even in case of relatively large overlap between images the algorithm can calculate false matches.

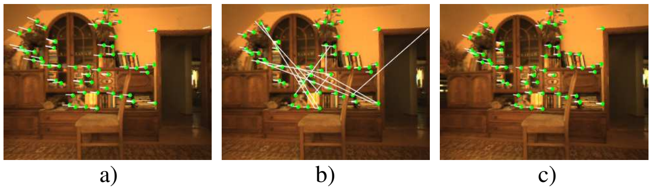

Fig. 5. Corner tracking using Lucas-Kanade optical flow, frame #2 (a), frame #3 (b), frame #4 (c).

In second test we compared our method with Kalman-based feature tracking, see Fig. 6. Through this sequence we want to highlight the behavior of the Kalman-based tracking algorithm in case of relatively large overlap between images. In depicted sequence the tracking starts from frame #1 and the subsequent frames are processed next. However, our experiment findings show that this method also calculates false tracks. This was observed when it started from frame #1 and then processed every third (or even every other) frame of sequence from Fig. 2. This method is not able to perform tracking of features in case of small overlaps between images, compared to Fig. 2 and Fig. 4. It should be noted that computation burden of this method is quite a large because for each tracked corner we add every candidate corner to the list of possible matches.

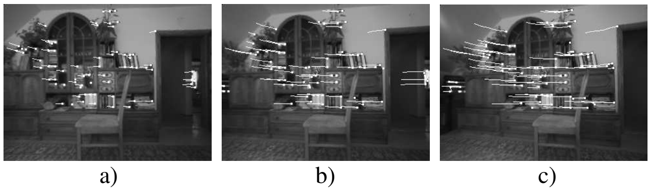

Fig. 6. Kalman filter based corner tracking on the basis of successive images, frame #2 (a), frame #4 (b), frame #7 (c) from the sequence.

The confidence measure that the candidate corner corresponds to the tracked corner is built on: (i) scaled differences between gradient vectors at the tracked and candidate locations, scaled by the standard deviation of the gradient, (ii) difference between the predicted corner location (on the basis of a linear motion model) and the candidate corner location, scaled by the standard deviation in position, (iii) cross correlation between pixel values. The standard deviation is utilized to express the reliability of the measure and is extracted from the covariance matrix of the Kalman filter.

In the next stage we verified the features that were extracted through corner and SIFT detectors. Experiments demonstrated that the larger the number of features, the better is the pose estimate, in general. But they demonstrated also that the number of consecutive frames, in which the same feature has been extracted, is very important factor. Experiments on several image sequences demonstrated that each corner was visible in 4-5 frames on average, whereas each SIFT feature was visible in only 3-4 frames. Using images from previous sections the SIFT features are computed in about 0.94 sec, whereas corners are extracted in 0.04 sec.

Several tests were performed in order to verify the effectiveness of the pose estimation. The system determines initial orientation of a robot using algorithm termed Pose from Orthography and Scaling with ITerations (POSIT) [20]. By approximating perspective projection with weak-perspective projection POSIT determines a camera pose estimate from a given image. The system assumes at this stage that the environment is piecewise planar. For hallway dominated scenes or home rooms this is a reasonable assumption. The mentioned algorithm works on the basis of data provided by a plane extractor, which utilizes simple image processing techniques and depth information. Next using feature pairs from two consecutive images we determine the initial pose change. The algorithm that was described in Section IV is utilized at this stage. Assuming the system should work with the frame rate of 3.5 Hz, SBA processes features from 3-5 images depending on image complexity and the number of extracted features. In the next time stamp we calculate the initial pose for SBA using a linear combination of data obtained by method described in Section IV and the previous SBA's result. The accuracy of the pose estimation was examined on L-shaped path of 2×8 m in a typical hallway. The end pose deviation from ground-truth was 0.29 m on average. The pose in work [22] was estimated with similar accuracy but our method runs in real time.

The algorithms have been implemented with VC++, and all experiments were conducted on a laptop equipped with 2.4 GHz Pentium IV processor and 1 MB RAM memory. The resulting system runs in near real-time on the laptop installed on the robot Pioneer 2DX with a Videre Design stereo head. The system processes images of size 320×240. The Gabor filter response is computed in about 0.12 sec. The system runs at rates about 3.5 Hz using a configuration of the software with corner extraction and SBA processing the features from 3-5 images.

VII. Conclusions

We presented a new method for determining inter frame correspondence between features. The method is based on Singular Value Decomposition of the cross correlation matrix of two feature sets, which is weighted on the basis of cross correlation and Gabor filter responses. Compared to correlation based feature matching that was used in recent algorithms for determining odometry using only visual input, our method is far more suitable in such tasks. Experiments indicate also that our method for correspondence determining leads to more precise pose estimation with less computation time. This is achieved thanks to reduced number of wrong matches between features. This fact significantly contributes to the efficiency of the whole algorithm for camera's pose estimation. At the refinement stage of pose estimation the Sparse Bundle Adjustment algorithm is used. We demonstrated that in case of good initial estimates the SBA can be executed in real-time. In order to reduce the error through processing features from up to 10 frames at once by SBA, the Gabor filter will be re-implemented to run on GPU.

References

[1] D. Nister, O. Naroditsky, and J. Bergen, ”Visual odometry,” Proc. IEEE Comp. Society Conf. on Comp. Vision and Pattern Recognition, pp. 652-659, 2004.

[2] S. Umeyama, ”Least-squares estimation of transformation parameters between two point patterns,” IEEE Trans. on Pattern Anal. Machine Intell., vol. 13, no. 4, pp. 376-380, 1991.

[3] A. J. Davison, ”Real-time simultaneous localization and mapping with a single camera,” IEEE Int. Conf. on Comp. Vis., pp. 1403-1410, 2003.

[4] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, ”FastSLAM: A factored solution to the simultaneous localization and mapping problem,” In Proc. of AAAI National Conf. on Artificial Intelligence, Edmonton, Canada, pp. 593-598, 2002.

[5] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, ”Rover navigation using stereo ego-motion,” Robotics and Aut. Systems, vol. 43, pp. 215-229, 2003.

[6] I. Urlich, and I. Nourbakhsh, ”Appearance-based place recognition for topological localization,” IEEE Int. Conf. On Robotics and Automation, pp. 1023-1029, 2000.

[7] G. Scott, and H. Longuet-Higgins, ”An algorithm for associating the features of two patterns,” In Proc. Royal Society London, vol. B244, pp. 21-26, 1991.

[8] D. Fleet, ”Disparity from local weighted phase-correlation,” In Proc. IEEE Int. Conf. on System Man and Cybernetics (SMC), pp. 46-48, 1994.

[9] M. A. Fischler, and R. C. Bolles”, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”, Comm. of the ACM, vol. 24., no. 6., pp. 381-385, 1981.

[10] Z. Zhang, and Y. Shan, ”Incremental motion estimation through local bundle adjustment” Tech. Report, Microsoft Research, 2001.

[11] R. I. Hartley, and A. Zisserman, ”Multiple view geometry in computer vision” Cambridge University Press, sec. ed., 2004.

[12] J. Jones, and L. Palemer, ”An evaluation of the two dimensional Gabor filter model of simple receptive fields in cat striate cortex,” Journal of Neurophysiology, vol. 58 pp. 1233-1258, 1987.

[13] K. Konolige, ”Small Vision System: Hardware and Implementation,” Proc. of Int. Symp. on Rob. Res., Hayama, pp. 111-116, 1997.

[14] C. Harris, and M. Stephens, ”A combined corner and edge detector,” Proc. of Fourth Alvey Vision Conference, pp. 147-151, 1988.

[15] M. Nixon, and A. Aguado, ”Feature Extraction and image processing,” Newnes, Oxford, Boston, 2002.

[16] D. Lowe, ”Distinctive image features from scale invariant keypoints,” Int. J. Comput. Vision, vol. 60., no.2, pp. 91-110, 2004.

[17] B. K. P. Horn, ”Robot Vision,” The MIT Press, 1986.

[18] M. I. A Lourakis, and A. A. Argyros, ”The design and implementation of a generic sparse bundle adjustment software package based on Levenberg-Marquardt algorithm,” Tech. Report 340, Institute of Computer Science-FORTH, Crete, Grece, 2004.

[19] J.-Y. Bouguet, ”Pyramidal implementation of the Lucas Kanade feature tracker, Intel Corporation, 2000.

[20] D. F. DeMenthon, and L. S. Davis, ”Model-Based Object Pose in 25 Lines of Code”, In Proc. of ECCV, pp. 335-343, 1992.

[21] C. V. Stewart, ”Robust parameter estimation in computer vision”, SIAM Review, vol. 41, no. 3, pp. 513-537, 1999.

[22] N. Sünderhauf, K. Konolige, S. Lacroix, and P. Protzel, ”Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle”, Tagungsband Autonome Mobile Systeme, Springer Verlag, pp. 157-163, 2005.

[23] D. Nistér, ”Preemptive RANSAC for Live Structure and Motion Estimation,” In Proc. IEEE Int. Conf. on Computer Vision, Mice, France, pp. 199-206, 2003.

[24] S. Se, D. Lowe, and J. Little, ”Global localization using distinctive visual features,” In Proc. Int. Conf. on Intell. Robots and Systems, Lausanne, Switzerland, pp. 226-231, 2002.