SIFT BASED ALGORITHM FOR POINT FEATURE TRACKING

Adrian Burlacu, Cosmin Copot and Corneliu Lazar

"Gh. Asachi" Technical University of Iasi ,
Department of Automation and Applied Informatics
Blvd. Mangeron 53A, 70050, Iasi Romania; e-mails:aburlacu@ac.tuiasi.ro

Abstract: In this paper a tracking algorithm for SIFT features in image sequences is developed. For each point feature extracted using SIFT algorithm a descriptor is computed using information from its neighborhood. Using an algorithm based on minimizing the distance between two descriptors tracking point features throughout image sequences is engaged. Experimental results, obtained from image sequences that capture scaling of different geometrical type object, reveal the performances of the tracking algorithm.

Keywords: SIFT feature, magnitude, orientation, descriptor, tracking.

1. INTRODUCTION

Image features poses different properties that allow to be tracked in an image sequence. In a real time analysis of the matching accuracy between frames of an image sequence the disturbances that need to be considered are scaling variance, camera view point change, illumination flux. Recovering trajectories of moving objects represents an application in which the correctness of feature extraction and matching is critical.

Thus considering features that are detected in an image the difficulty is to track them in a different image that was acquired after disturbances affected the environment. Different applications of image features tracking were considered in the last decade: analysis of medical images (Cheung and Hamarneh, 2007), vehicle guidance (Murarka *et al.*, 2006), visual servoing used to control the motion of a robot (Chaumette and Hutchinson, 2006). In the case of visual servoing features may be acquired from a camera that is mounted directly on a robot manipulator or is fixed on the workspace. The first configuration is called eye-in-hand and will be considered in the experimental section of this paper. For tracking features applications, SIFT detector (Lowe, 2004) is a serious candidate due to its properties required in motion analysis: invariant to image scale and rotation, robust to change in illumination or in 3D viewpoint.

In this paper the development of a tracking algorithm for point features extracted from image sequences using SIFT algorithm is presented. A descriptor based on magnitude and orientation is constructed from information precomputed by the SIFT detector. Using this descriptor a low level matching algorithm is employed. For a feature that represents the search model a candidate feature from a different image is considered the matching correspondent if it minimizes a certain criterion like Euclidian distance. The performances of developed tracking algorithm were tested on image sequences obtained from a real visual servoing system, in different imaging conditions: rotation of the camera, view point and scale changes. Considering as start a frame were the positions of the features are known the analysis was conducted in order to establish the number of frames that keep stable the extracted SIFT features.

A work environment composed from a six dof ABB robot with an eye-in-hand configuration and different shaped objects is considered for constructing the

This paper was recommended for publication by Dorel Aiordăchioaie

image sequences that will be analyzed. The considered software for implementing the tracking algorithm was Matlab, especially its image processing toolbox.

The paper is organized as follows: In Section 2 the SIFT detector is presented and in Section 3 the SIFT feature descriptor is detailed. Section 4 is dedicated to experimental results while Section 5 reveals the conclusions of the research.

2. SIFT DETECTOR

SIFT detector allows to extract point features from images invariant to image scale and rotation. The detection of scale-invariant image features algorithm (Lowe, 2004) can be decomposed in four stages: two for detection of scale-space extreme and accurate point features localization and other two for orientation assignment and description of point features. The first two stages which allow point features detection are presented in the sequel.

Detecting invariant locations to scale change of the image is based on reaching stable features across all possible scales. In order to create the scale space an image denoted *I(x,y),* with $N_I \times M_I$ size, must be part of a convolution with the Gaussian kernel:

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}e^{-\frac{\left(x^2+y^2\right)}{2\sigma^2}}. \qquad (1)$$

Application:

$$L: A \subset \Re^3 \rightarrow \Re \qquad (2)$$

defined as:

$$L(x,y,\sigma)=G(x,y,\sigma)\otimes I(x,y) \qquad (3)$$

represents the successive image filtering implementation for creating the scale – space.

The parameter *σ* from (3) is considered to be the application:

$$\sigma:[o_{\min},o_{\max}]\times[0,S-1]\rightarrow\Re \qquad (4)$$

with:

$$\sigma(o,s)=\sigma_0\cdot 2^{\frac{o+s}{S}}. \qquad (5)$$

In (5), *o* represents an octave of the *σ* axis from the scale space, *S* the number of levels for each octave and *s* the index of a level from the *o* octave. Then, computation for the scale – space of *I* can be implemented recursively.

Starting with *I(x,y),* $L(x,y,\sigma(o,s))$ is computed first using (3) and is denoted the dimension of the filtered image with $N_O \times M_O$. Next, $\sigma(o,s+1)$ is computed using (4) and it is tested the condition *s+1=S.* If the condition is fulfilled, then the image in the next octave will have the size defined by:

$$N_{O+1}=\left[\frac{N_I}{2^O}\right], M_{O+1}=\left[\frac{M_I}{2^O}\right] \qquad (6)$$

and the algorithm is reiterated using the image $L(x,y,\sigma(o,S))$. After the scale – space was computed, it will be created the difference-of-Gaussian (DoG) space, using the difference of two nearby scales separated by a constant multiplicative factor *k*:

$$D(x,y,\sigma)=L(x,y,k\sigma)-L(x,y,\sigma). \qquad (7)$$

In the DoG space, (that can be interpreted as a derivation of the scale – space), we look for positions $\tau=(x,y,\sigma)$ where the value $D(\tau)$ is larger than all of its 26 neighbours from the 3x3 cube neighborhood. The ensemble of these positions is denoted with *Λ*.

In the second stage of scale – invariant image feature detection, different criteria for establishing the stability of each element of *Λ* are used. First step is to establish the difference of Gaussian extreme in the neighborhood of each element *τ* from *Λ*. Lowe's approach used Taylor expansion up to the quadratic terms of the scale- space function:

$$D(\tau)=D(0)+\frac{\partial D^T}{\partial\tau}\tau+\frac{\partial^2 D^T}{\partial\tau^2}\tau^2 \qquad (8)$$

shifted so that the origin is the sample point.

D and its derivatives are evaluated at the sample point and *τ* is the offset from this point. The resulting extremum from $\frac{\partial D(\tau)}{\partial\tau}=0$ is an interest point in the scale – space having the location:

$$\hat{\tau}=-\frac{\partial D^{-1}}{\partial\tau^2}\frac{\partial D}{\partial\tau}. \qquad (9)$$

In order to reject unstable extrema with low contrast the function value at the extremum:

$$D\left(\hat{\tau}\right)=D\left(0\right)+\frac{\partial D^T}{\partial \tau}\hat{\tau} \qquad (10)$$

is computed and all extrema, with a value of $|D\left(\hat{\tau}\right)|$ less than a threshold *α,* are eliminated.

Eliminating edge responses is done using the eigenvalues of Hessian matrix:

$$\boldsymbol{H}=\begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \qquad (11)$$

which are proportional to the principal curvatures of *D.* The derivatives are estimated taking into account differences of neighbor sample points. Denoting with λ_L the eigenvalue with the larger magnitude, with λ_S the smaller one and with r the ratio λ_L/λ_S, it results:

$$\frac{Tr\left(\boldsymbol{H}\right)^2}{Det\left(\boldsymbol{H}\right)}=\frac{\left(\lambda_L+\lambda_S\right)^2}{\lambda_L\cdot\lambda_S}=\frac{\left(r+1\right)^2}{r}, \qquad (12)$$

where:

$$\begin{aligned} Tr\left(\boldsymbol{H}\right)&=D_{xx}+D_{yy}=\lambda_L+\lambda_S \\ Det\left(\boldsymbol{H}\right)&=D_{xx}D_{yy}-D_{xy}^{\ 2} \end{aligned}. \qquad (13)$$

Thus, using the inequality:

$$\frac{Tr\left(\boldsymbol{H}\right)^2}{Det\left(\boldsymbol{H}\right)}<\frac{\left(r+1\right)^2}{r}, \qquad (14)$$

point features with a ratio between the principal curvatures greater than threshold *r* are eliminated.

For each candidate, point feature interpolation of nearby data is used to accurately determine its position. Point features with low contrast and responses along edges are removed.

3. TRACKING SIFT FEATURES

Tracking features is one of the most difficult task in computer vision. The need of high performance algorithms for tracking features generated different approaches (Gavrila and Munder, 2007). Next, a tracking algorithm based on constructing a magnitude and orientation descriptor is detailed. Information precomputed using stages described in Section 2 like the scale space are used.

3.1 SIFT feature descriptor

After keypoint extraction using the first two stage of the SIFT algorithm presented above, an invariant descriptor construction is revealed as follows. Using properties of local neighborhood of each keypoint a description based on magnitude and orientation is constructed. Considering the scale of the keypoint it is selected the *L* Gaussian smoothed image with the closest scale in order to perform the computations in an invariant scale manner.

The *M* magnitude of gradient and *θ* orientation are computed using (Lowe, 2004):

$$\begin{aligned} M\left(x,y\right)&=\sqrt{\left(L_{x+1,y}-L_{x-1,y}\right)^2+\left(L_{x,y+1}-L_{x,y-1}\right)^2} \\ \theta\left(x,y\right)&=\tan^{-1}\left(\left(L_{x,y+1}-L_{x,y-1}\right)/\left(L_{x+1,y}-L_{x-1,y}\right)\right) \end{aligned} \quad (15)$$

An orientation histogram is constructed using information from the 16x16 neighborhood of each keypoint. Having a 16x16 neighborhood, a decision of positioning the “center” of the neighborhood is needed. It was considered that the keypoint’s position will be (8,8).

After computing the 256 magnitudes and orientations the 16x16 neighborhood is divided in 16 blocks with 4x4 dimensions. For each block the 16 orientations and magnitudes are merged using 8 bins that equally dived the 360° degrees of the trigonometric circle as it is shown in Fig.1.

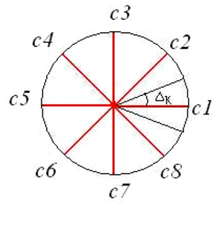

Fig.1 The orientation circle.

The weight (as magnitude) of each orientation is computed using:

$$h_r\left(l,m\right)=\sum_{x,y\in r\left(l,m\right)} M\left(x,y\right)\left(1-|\theta\left(x,y\right)-c_k|/\Delta_k\right), \quad (16)$$

where c_k is one of the 8 bins and Δ_k is a constant of 22.5° degrees that is used to establish the relation of each orientation θ to c_k. For a 4x4 block the resulting histogram appears like in Fig.2.

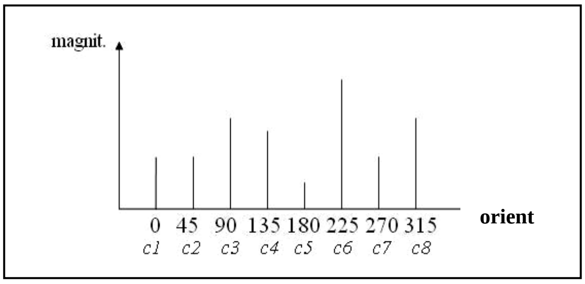

Fig.2 Orientation histogram

All the information that describes a keypoint will be grouped in a 128 length vector. The final step is to normalize the vector values for an invariance response to different transformations of the analyzed images.

3.2 Feature tracking algorithm

A low level tracking algorithm is developed using a distance to reveal the matching correspondence. Considering an image sequence composed from *n* frames the tracking algorithm can be described iteratively. For every frame *i* (*i*=1,*n*) the SIFT detector is used to find positions of point features. Each point feature is characterized by position and scale level. Using this information, the SIFT feature descriptor presented in section 3.1 is engaged in constructing the features descriptor. Capturing extended properties of the SIFT feature by considering a neighborhood of 16x16 from an scaled image the magnitude and orientation of each 256 pixels are computed. Next the 16x16 magnitude and orientation table is divided in 4x4 blocks. Each block represents an orientation histogram obtained after computing weights using equation (16). These weights are grouped in vectors of length equal to 128:

$$X_i = \left(x_1, x_2, \ldots, x_{128}\right),\ i = \overline{1,n} \tag{26}$$

Tracking a feature in an image sequence can be considered using the followings: if the analysis is conducted in order to reveal the stability of a feature extracted in the first frame or for each two consecutive frames it is necessary to match features.

It is considered a SIFT feature to be model and all the other SIFT features from a different frame to be matching candidates. Each candidate $C_j,\ j = \overline{1,n},\ j \neq i$ from the next frame is compared using Euclidian distance:

$$d\left(X_i, C_j\right) = \sqrt{\sum_{k=1,128} (X_i^k - C_j^k)^2}\ . \tag{27}$$

If

$$d\left(X_i, C_j\right) < \alpha\ , \tag{28}$$

where α is a threshold value that will influence the quality of the tracking, then C_j is declared the matching correspondent of X_i. In order not to depend on the choice of α value and considering the evaluation of the performances of SIFT algorithm, it was chosen a shortest distance search algorithm that find the minimum of the distances between the point feature descriptor and all the candidates.

4 EXPERIMENTAL RESULTS

For testing the point feature tracking algorithm presented in Section 3, grasping applications of objects with different structure were considered. The experimental set (Fig.3) is composed from an ABB six dof robot, a camera mounted on the end effector and different objects. Image sequences that present the motion of the end effector toward an object were acquired and frames were extracted using a sample period of 1 sec.

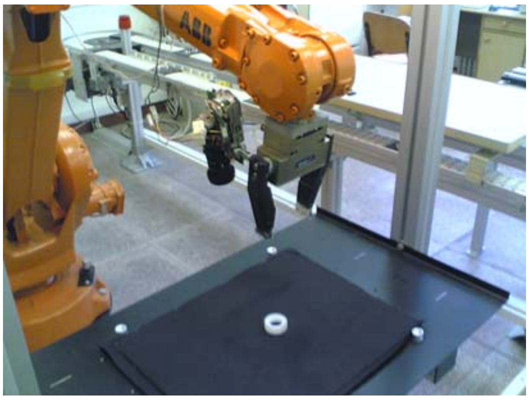

Fig.3 Experimental set

The considered objects have different shapes: rectangular and circular. For each object, an image sequence, that captures motions of the eye-in hand camera towards the desired position, was acquired. Using Matlab software (image processing toolbox mainly) each image of the sequence was first transform from rgb format to monochrome (gray scale).

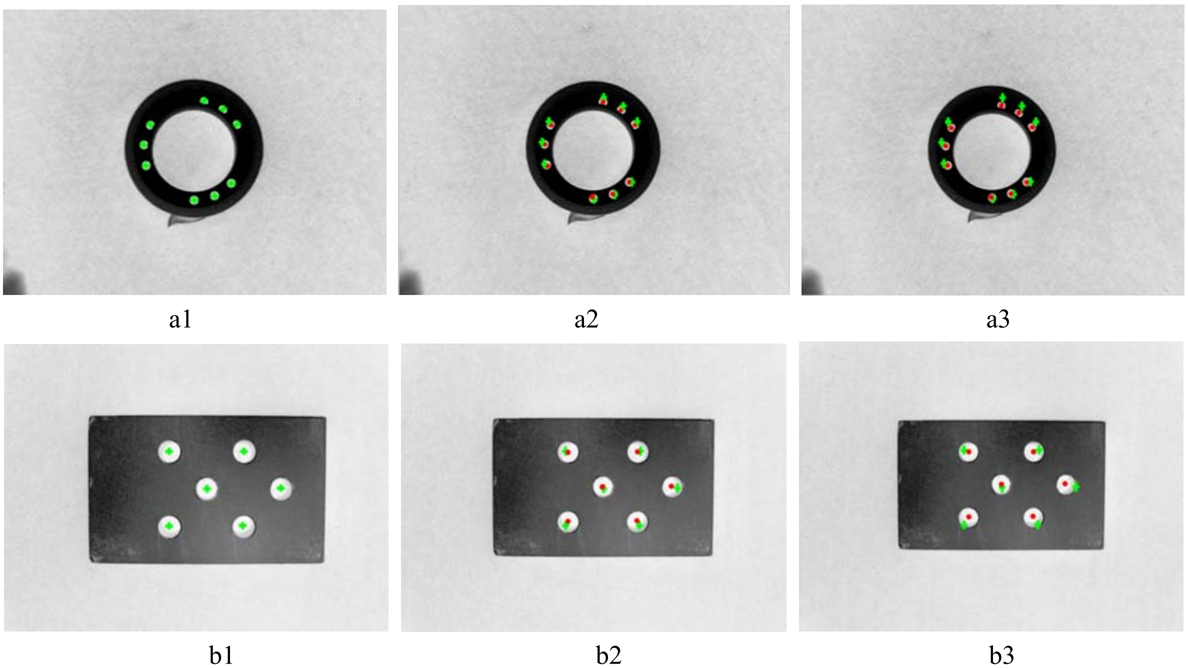

Fig.4 SIFT feature tracking results for a circular object (a1,a2,a3) and a rectangular object (b1,b2,b3)

Frames were extracted from image sequences and using the SIFT detector, point features were found for each frame. In Fig.4, the frame *a1* represents the initial position of the camera when the analyzed scene contains the circular object and the frame *b1* the initial position of the camera in the case of rectangular object. With dots are represented the point features detected using SIFT algorithm. Inputs in the tracking algorithm are considered the point features detected from the first frame of the image sequence. For the other frames, descriptors of the point features extracted using SIFT algorithm will represent the matching candidates in the proposed experiment. Using the algorithm presented in Section 3.1, each point feature was characterized using a 128 length descriptor. First the magnitude and orientation of the 256 pixels in the 16x16 neighborhood were computed and one of the results in presented in Fig.5.

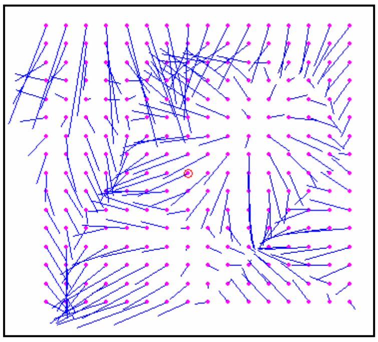

Fig.5 A 16x16 neighborhood of a SIFT feature in which the gradient magnitude and orientation was computed.

The next stage represents the splitting of the 16x16 neighborhood in 4x4 blocks (fig.6a) and for each 4x4 block an orientation based histogram is constructed (fig.6b).

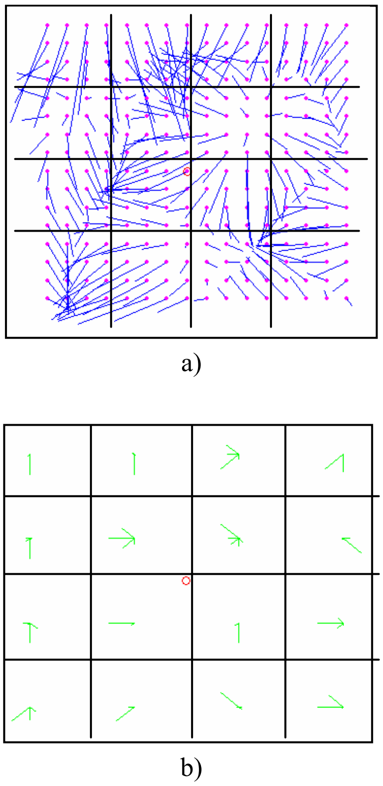

Fig.6 Splitting the 16x16 neighborhood in 4x4 blocks (a) and representing the 4x4 orientation histogram (b)

Due to the performances of the SIFT detector (Lazar and Burlacu, 2006) through the entire image sequence the stability, appearance or disappearance of features, of the point features extracted in the first frame is 100%. Knowing this fact the proper choice of the threshold value for the Euclidian distance (27) is the minimum of all distances between a point features descriptor and all its possible candidates. After a matching is occurred a high value is saved for the matched candidate and the algorithm reiterates with the next point feature.

The experimental results showed a high matching percentage for image sequences composed from 10 frames for the circular object and 12 frames for the rectangular object. In Fig.4 only 3 frames for each image sequence are presented. In both cases, circular object Fig4 (a2, a3) and rectangular object Fig.4 (b2,b3), the tracking algorithm works very well. The two set of dots represent the positions of the point features extracted in the first frame and the positions of the matching point features extracted using SIFT algorithm in a different frame.

5. CONCLUSIONS

In the present paper a scale invariant feature tracking algorithm was developed. Using SIFT feature a descriptor based on magnitude and orientation was computed and stored in a 128-length vector. A low level matching algorithm based on minimizing the Euclidian distance between two feature descriptors was considered in order to establish the matching correspondence of SIFT features from different frames. Experimental results showed that the developed tracking algorithm works properly in industrial environments with low illumination fluctuation.

REFERENCES

Chaumette F., S. Hutchinson (2006), Visual Servo Control Part I: Basic Approach, *IEEE Robotics and Automation Magazine*, **13**(4), pp.82-90

Cheung, W. Hamarneh, G. (2007), N-SIFT: N-Dimensional Scale Invariant Feature Transform for Matching Medical Images, *ISBI 2007. 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro,* pp. 720-723,

Gavrila D.M. and Munder S. (2007), Multi-cue pedestrian detection and tracking from a moving vehicle, *International Journal of Computer Vision*, **73**(1), pp.41-59.

Lazar C. and Burlacu A.(2006), Detection and tracking of feature points in image sequences, in *Proc. 12th IEEE Int. on Methods and Models in Automation and Robotics*, Miedzyzdroje

Lowe D.G. (2004), Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, **60**(2), pp. 91-110.

Murarka A., Modayil J. and Kuipers B. (2006), "Building Local Safety Maps for a Wheelchair Robot using Vision and Lasers**"**, Proceedings of the The 3rd Canadian Conference on Computer and Robot Vision (CRV'06) - Volume 00, pp.25-32.