A Visual Odometry Framework Robust to Motion Blur

Alberto Pretto, Emanuele Menegatti, Maren Bennewitz, Wolfram Burgard, Enrico Pagello

***Abstract*— Motion blur is a severe problem in images grabbed by legged robots and, in particular, by small humanoid robots. Standard feature extraction and tracking approaches typically fail when applied to sequences of images strongly affected by motion blur. In this paper, we propose a new feature detection and tracking scheme that is robust even to non-uniform motion blur. Furthermore, we developed a framework for visual odometry based on features extracted out of and matched in monocular image sequences. To reliably extract and track the features, we estimate the point spread function (PSF) of the motion blur individually for image patches obtained via a clustering technique and only consider highly distinctive features during matching. We present experiments performed on standard datasets corrupted with motion blur and on images taken by a camera mounted on walking small humanoid robots to show the effectiveness of our approach. The experiments demonstrate that our technique is able to reliably extract and match features and that it is furthermore able to generate a correct visual odometry, even in presence of strong motion blur effects and without the aid of any inertial measurement sensor.**

I. Introduction

In mobile robotics, odometry plays an essential role. Odometry information is a precondition in most robot localization and SLAM approaches. Odometry is simple and reliable to obtain with wheeled robots, where it is given by the wheel encoders. However, sometimes odometry is not available: this is the case for flying robots and also for humanoid robots, where the complex kinematics combined with the unpredictable body movements prohibit a reliable reconstruction of the motion from the servo-motor encoders. In such cases, odometry has to be estimated in other ways. In the last few years, several researchers have proposed ”visual odometry” systems, in which the ego-motion of the robot can be estimated using on-board perspective cameras [7], [19], [18], [6]. In almost all visual odometry systems, one can identify the following steps: (i) point features are detected, (ii) these features are tracked along the image sequence, (iii) the odometry is recovered from the apparent motion in the image plane of the tracked features. Item (i) and item (ii) are essential keys for a correct ego-motion estimation.

Most of the proposed visual odometry approaches were developed for wheeled robots, but humanoid robots introduce novel challenges to visual odometry. When a humanoid robot is walking, turning, or squatting, its camera moves in a jerky and sometimes unpredictable way. This causes an undesired motion blur in the images grabbed by the robot's camera that negatively affects the performance of the feature detectors and especially of the feature tracking classic algorithms. A typical image affected by motion blur grabbed by a walking robot is depicted in Fig. 1.

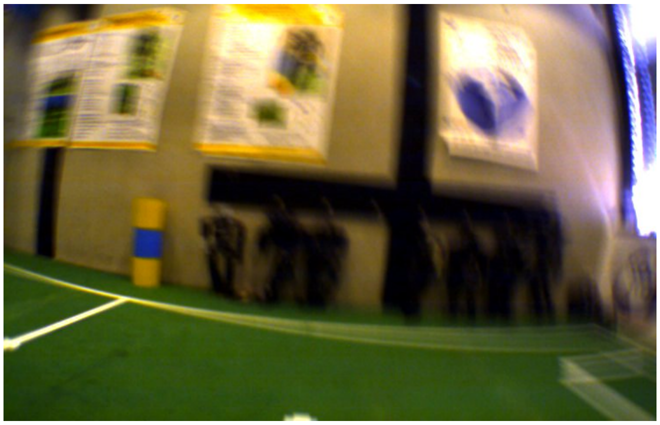


Fig. 1. Typical image grabbed by a walking humanoid robot. As can be seen, the image is highly affected by motion blur.

Indeed, the classical feature detectors and descriptors [13], [1] that proved to work well for wheeled robots, do not perform reliably in presence of motion blur.

This paper proposes a visual odometry framework based on monocular images designed to address the specific problem of motion estimation robust to motion blur. The aim of this work is to provide the robot with a reliable odometry based on the images grabbed by a camera mounted on the robot, suitable to all navigation tasks that need a priori knowledge of the robot motion (e.g., localization and SLAM). The presented approach was tested on small humanoid robots, but it could be applied also to other robots whose quick movements can affect the quality of vision data by inducing a motion blur effect. Several authors studied the problem of estimation, modeling, and elimination of the motion blur in robot images. Some examples are Flusser *et al.* [8] and Klein *et al.* [11], in which only a centrally symmetric motion blur is taken into account, and Mei *et al.* [15], in which a tracking problem is analyzed in presence of spatially variant motion blur generated by a planar template.

Our approach is based on a novel invariant feature scheme robust to motion blur that takes advantage of the previous works of Lowe [13] and Bay *et al.* [1]. In our approach, before detecting interest points, an image preprocessing step estimates the *Point Spread Function* (PSF) of the motion blur in the image. However, we experienced that the motion blur, affecting the images taken by a humanoid robot while

Pretto, Menegatti, Pagello are with the University of Padova, Dep. of Information Engineering (DEI), via Gradenigo 6/B, 35131 Padova, Italy

Bennewitz and Burgard are with with the University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany

walking, is not uniform. Thus, we calculate not a unique PSF in the whole image, but we segment each image on the basis of the local motion blur. The estimated PSFs are then used to build an adapted scale-space representation trying to minimize the undesired effect of the motion blur. The scale-space extrema are extracted based on the determinant of the Hessian, and a SIFT descriptor is calculated for each keypoint. Before matching features between images, features with less distinctive descriptors are discarded based on their entropy. In the end, the robot ego-motion is estimated from the matched features with a method based on the five-point algorithm (similarly to the visual odometry strategy proposed by Nister *et al.* [19]).

We present experiments in which odometry could reliably be estimated from images grabbed by walking humanoid robots in the presence of strong motion blur effect. This is obtained without any global bundle adjustment process (a process which is too computationally expensive for the processing units on-board of small humanoids robots) and without the aid of any inertial measurement sensor.

II. Related Work

Several authors investigated the problem of a reliable and robust feature detection and tracking on its own. The best-known and widely used feature detector and descriptor scheme was introduced by Lowe [13] and is called SIFT (Scale-invariant feature transform). SIFT features are invariant to image scale and rotation, and are quite robust in matching across affine transformations and changing of viewpoint. As shown in [17], SIFT features outperformed previous features detectors-descriptor schemes. Ke *et al.* [10] proposed a variation of the SIFT features, called PCA-SIFT: applying PCA in the gradient images, the descriptor is reduced to a 36-dimensional vector, and matching step is faster. PCA-SIFT are robust to focus-blur noise, but are less discriminative compared with SIFT [17]. Mikolajczyk *et al.* proposed a novel approach for detecting interest points invariant to scale and affine transformation [16]. In [17] a novel descriptor called GLOH (Gradient location-orientation histogram) is presented, an extension of the SIFT descriptor designed to increase its robustness and distinctiveness: it also uses PCA to reduce the dimension of the descriptor. GLOH obtains little better performance than SIFT, but at cost of higher computational complexity. Recently, Bay *et al.* [1] presented a novel and computationally efficient scale and invariant feature detector-descriptor called SURF (Speeded Up Robust Features). Interest points are detected as the maxima over location and scale of the determinant of the Hessian. The Hessian is computed over scaled images using an efficient approximation based on the integral images technique. Descriptors are obtained using Haar Wavelet responses: repeatability and distinctiveness performance are similar than SIFT to previous proposed schemes, but SURF features can be computed much faster.

Lately, these approaches have being exploited not only for feature (or object) tracking, but also to estimate the camera motion just from the images (i.e., visual odometry). Davison proposed an invariant point-based SLAM approach using a single perspective camera and EKFs (Extended Kalman Filter) [7]. This approach is well suited for indoor environments and without cumulative error as in conventional odometry. Despite that, Davison visual SLAM tracks a small number of points and assumes to encounter the same points again and again in the future. Nister proposed a robust estimation of the camera motion produced from the point tracks using a geometric hypothesize-and-test architecture [19]: This method is very accurate, but requires to track points for many consecutive frames. Comport *et al.* described an image-based approach for tracking the trajectory of a stereo camera based on a quadrifocal relationship between the image intensities within adjacent views of the stereo pair [6]. Mouragnon *et al.* proposed an accurate and fast incremental motion reconstruction algorithm that uses a local bundle adjustment method to improve motion estimation accuracy [18].

III. Invariant features robust to motion blur

When an image is captured while the camera is moving during the exposure time, the one-to-one relationship between the scene points and the image points is broken and a certain number of scene points are projected at a single pixel contributing to the final pixel value. This effect is called *motion blur*, and it depends on the relative movement between the camera and the objects of the observed scene during the exposure time. When this motion is linear and with uniform velocity, the blur can be determined by two parameters: the blur extent d and the direction θ. The observed image $b(x,y)$ can be expressed as:

$$b(x,y) = h(x,y) * f(x,y) + n(x,y) \tag{1}$$

where $h(x,y)$ is the blurring function, called *PSF* (*Point Spread Function*), $*$ is the convolution operator, $f(x,y)$ is the uncorrupted version of the observed image (i.e., the ideal image grabbed without relative motions), and $n(x,y)$ is an additive noise function. In the linear case:

$$h(x,y) = \begin{cases} \frac{1}{d} & \text{if } \sqrt{x^2+y^2} \leq \frac{d}{2}, \frac{x}{y} = -tan(\theta) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Several deconvolution techniques have been implemented to restore an image affected by motion blur effect. Richardson-Lucy algorithm [14] and Wiener filter [22] are widely used techniques. Unfortunately the quality of the restored image strongly depends on the accuracy of the PSF estimation. Wrong PSF used for the deconvolution can produce unacceptable resulting image. Moreover, this method assumes a linear motion blur and the presence of PSF uniform in the whole image. Even if in simple small humanoid robots, these assumption can hold to a certain extent [21], from our experiments, however, we experienced that the cameras of robots with complex kinematics, like for instance the humanoid platform of the NimbRo Robocup Team [2], perform complex movements resulting in different translation and rotation of the image that can introduce non-linear and non-uniform motion blur effect. In these case, conventional deconvolution techniques can easily fail. Instead of trying to restore the original, unblurred images, we propose an adapted scale-space representation that tries to overcome the negative effect of the motion blur in the invariant features detection and description process. With respect to the work presented in [21], we improved the estimation of the PSF by relaxing the constraint of a uniform PSF over the hole image and which leads to a better estimate than the simple Wiener filter deconvolution presented in that work.

The scale-space theory of Lindeberg [12] aims to represent the input image at different scales and it is at the base of the scale-invariant feature detectors and descriptors such as SIFT and SURF. Scale-space representation is obtained convolving the original images $f(x,y)$ with a set of Gaussian filters with zero-mean $g(x,y,\sigma)$ and with increasing standard deviations σ (normally referred to as the *scale* of the smoothed image):

$$l(x,y,\sigma) = g(x,y,\sigma) * f(x,y) \tag{3}$$

If one uses the conventional scale-space representation for images affected by motion blur, it will blur with Gaussian noise the image $b(x,y)$ that is already blurred with the motion blur $h(x,y)$. Thus, the resulting filter is not the desired Gaussian filter, but the composition of a Gaussian filter plus the motion blur filter (applied to the uncorrupted image by the motion of the camera). The motion blur filter can be approximated to be Gaussian, but cannot be approximated to be with equal marginal standard deviations. Thus, the resulting filter is no longer circular symmetric. Therefore, we propose to compute the scale-space representation of an image corrupted by motion blur by finding an appropriate non-circular symmetric $g'(x,y)$, determined from the PSF of the actual motion blur in the image, that convolved with $h(x,y)$ approximates a Gaussian filter with equal marginal standard deviations. In other words, *we smooth less the image along the motion blur direction*. We obtain from Eq. (1) and Eq. (3) (omitting for simplicity the additive noise):

$$l'(x,y,\sigma) = g'(x,y) * h(x,y) * f(x,y) \tag{4}$$

where $g'(x,y)$ is a zero-mean Gaussian smoothing filter with different marginal standard deviations. The proposed strategy is to find a $g'(x,y)$ filter that minimize the sum of squared difference between l and l' over the whole image (here, w is image width and h is image height in pixels):

$$\sum_{x=1}^{w}\sum_{y=1}^{h}(l-l')^2 \tag{5}$$

For the distributivity and associativity properties of the convolution operator one can write:

$$l - l' = (g(x,y,\sigma) - g'(x,y) * h(x,y)) * f(x,y) \tag{6}$$

So, for an image $f(x,y)$, we have to find a filter $g'(x,y)$ that minimizes the difference:

$$g(x,y,\sigma) - g'(x,y) * h(x,y) \tag{7}$$

Let us define as σ (i.e. the scale), the marginal standard deviation of $g'(x,y)$ in the direction perpendicular to the PSF direction, and σ' the marginal standard deviation in the PSF direction. One might think that $g'(x,y)$ could be easily obtained in the frequency domain by a standard deconvolution techniques as Wiener filter, but, for the reason explained above, without an accurate estimation of the real PSF $h(x,y)$, results are very poor. Thus, we compute the value of σ' by minimizing the function (7) in the discrete domain (i.e., using discrete kernel's filter): we use the Levenberg-Marquardt algorithm (LMA) for the solution of least squares problems in non-linear case. For example, given the PSF $h_0(x,y)$ with extent d and direction $\theta = 0$ and given the $scale = \sigma$, we compute using LMA the σ' that minimize:

$$g(x,y,\sigma) - g'(x,y,\Sigma) * h_0(x,y), \quad \Sigma = \begin{bmatrix} \sigma' & 0 \\ 0 & \sigma \end{bmatrix} \tag{8}$$

where Σ is the covariance matrix of the adapted Gaussian filter $g'(x,y,\Sigma)$. For a general PSF $h(x,y)$ with $\theta \neq 0$, we rotate the Gaussian kernel obtained for $h_0(x,y)$ according to θ (see Fig. 2).

A. PSF estimation

For PSF estimation, we use an approximated version of the *whitening method* [24]. Motion during exposure affects the image by decreasing its resolution mostly in the motion direction. We search for the direction in the image with the lowest resolution: this can be done high-pass filtering the image in all directions. The direction with the lowest responses corresponds the blur direction. The high-pass filter we use is the absolute value of the derivatives in the candidate directions: we take the absolute value of the difference between two adjacent pixels along the direction. For a better approximation, pixels are interpolated. In order to preserve the efficiency, we compute responses in 5 pixels, regularly spaced sample points along 18 directions (every 10 degrees): responses are accumulated in a 18-bins histogram. The bin with the lowest value represents the blur direction. In Figure 3 (a), the responses for a 45° blurred image. In order to estimate the blur extent, the PSF correlation properties along its direction are emphasized.

An auto-correlation operation (*ACF*) in the image derivative lines along motion direction is performed:

$$R_d(j) = \frac{1}{M}\sum_{i=-M}^{M} l(i+j)d(i), \quad j \in [M,-M] \\ d(i) = 0 \quad \text{for} \quad i \notin [0,M] \tag{9}$$

where $d(i)$ is the image derivatives line of index i in the motion direction. For theoretical details, see [24]. The operation is obtained by rotating the image with an angle of $-\alpha$, where α is equals to the blur direction angle. The derivatives along the x-axis of the resulting image are then calculated. Again, the auto-correlations responses are accumulated in a histogram. The global minimum of the histogram correspond to the blur extent estimation. In Figure 3 (b), the ACF for an image with motion blur extent of 30 pixels. In our

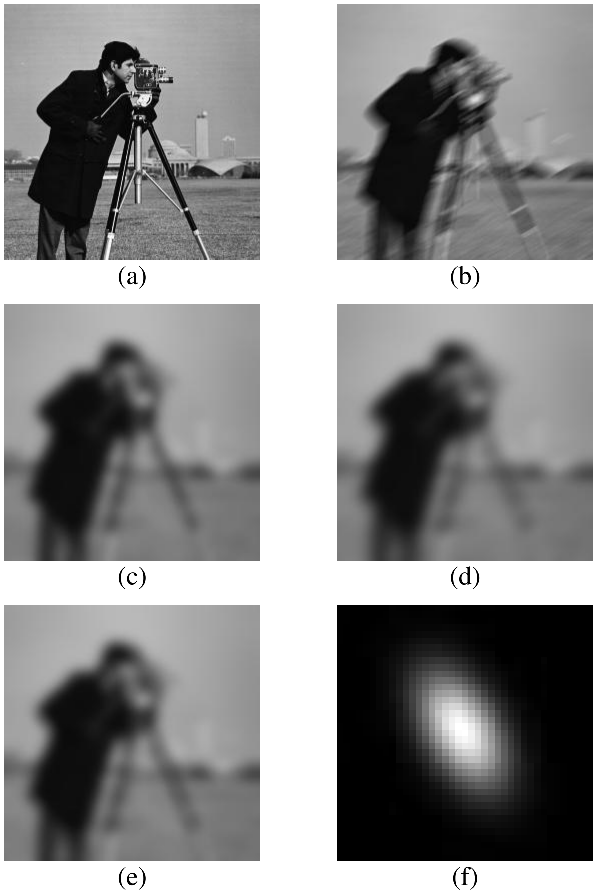


Fig. 2. (a) The Cameraman original image. (b) Original image synthetically blurred with PSF extent $d = 18$ and direction $\theta = \frac{5}{6}\pi$. (c) Original image smoothed with circular symmetric bivariate Gaussian kernel with $\sigma = scale = 6.4$. (d) Motion blurred image (b) smoothed with circular symmetric bivariate Gaussian kernel with $\sigma = scale = 6.4$. (e) Motion blurred image (b) smoothed with non circular bivariate Gaussian kernel with marginal standard deviation $\sigma = scale = 6.4$ in the direction perpendicular to the PSF and in this case $\sigma' = 3.69$ (computed with the LMA algorithm) in the PSF direction. (f) The adapted Gaussian filter used to obtain (e). We can see that (e) tends to approximate the original smoothed image (c) better than (d).

implementation, after a histogram-smoothing step, we search for negative peaks over a certain threshold (e.g., 2-3 pixels): The presence of noise in the images can induce negative peaks in the auto-correlation function at very low value of extent.

B. PSF clustering

The proposed adapted scale-space representation assumes that the PSF is linear: this is, in general, an approximation of the real PSF, that also isn't usually uniform in the whole image. In order to take into account of the non-uniform nature of the PSF, we introduce a clustering step that aims to divide the image in sub-regions characterized by different PSF.

The segmentation of the image is performed using a modified version of the K-means clustering algorithm [4]. Formally, we divide the image points in K clusters where the 3-dimensional vector $\mu_k = (x_{\mu_k}, y_{\mu_k}, \alpha_{\mu_k})$ (here, x_{μ_k}, y_{μ_k} are

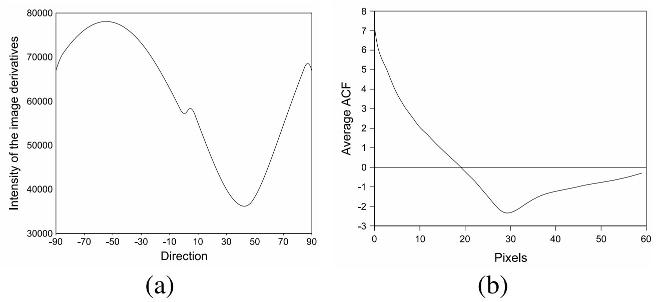


Fig. 3. (a) The motion-blur direction identification: the global minimum falls in the blur direction estimation (in degrees). (b) The average ACF used for estimation of the extent. The global minimum fall in the blur extent estimation (in pixels).

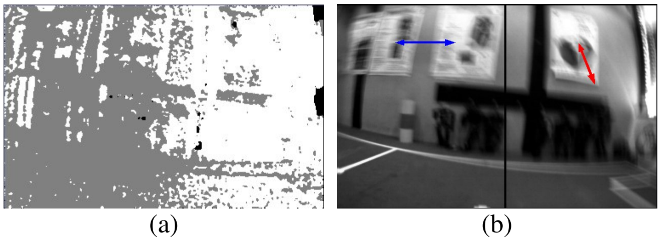


Fig. 4. (a) The result of the clustering process for the image (Fig. 1): Using the K-means algorithm (here with $K = 2$), each point is assigned to a cluster that is characterized by an (approximated) uniform PSF in the region close to the cluster centroid. (b) Based on segmentation performed in (a), the image is divided in rectangular subregions with assigned uniform PSFs. Red and blue arrows represent the computed directions of the PSF in each single subregion: about 0° on the left side and around 80° on the right side.

the image coordinates, α_{μ_k} is the PSF direction (discretized) and $k = 1, \ldots, K$) is a prototype associated with the k^{th} cluster. One can think of the μ_k as representing the centroids of the clusters with uniform PSF with direction α_{μ_k}. Given an image point X_i with coordinates x_i, y_i and H_i being the histogram (normalized to unit vector) that holds the absolute values of the derivatives in each discretized directions for this point (see section III-A), we define the distance from a cluster centroid μ_k as the weighted Euclidean distance:

$$d(X_i, \mu_k) = H_i(\alpha_{\mu_k}) * \sqrt{(x_{\mu_k} - x_i)^2 + (y_{\mu_k} - y_i)^2} \quad (10)$$

where $H_i(\alpha)$ is the response of the absolute derivatives along the direction α for the sample X_i: This response tends to be a minimum in the motion blur direction. Here's the algorithm:

1) For each sample point i, compute the histogram H_i that holds the absolute values of the derivatives in each discretized direction, 18 in our case (see section III-A). Each histogram is normalized to the unit vector.
2) Compute, as accumulation of the histograms of point 1), the global histogram that, for each discretized direction, holds the sum of the responses in that direction for all the sample points. Extract the K directions with minimum responses and assign them as initial choices for the direction α_{μ_k} of the the K cluster centroids μ_k. For all cluster centroids, assign at x_{μ_k}, y_{μ_k} the center of the image.

3) Using the distance (10), assign each sample point to the closest cluster, i.e., the cluster with the closest centroid.
4) Re-assign each sample point to the mode of its 8-neighbors sample points, i.e., to the cluster that occurs most frequently in its 8 neighbors (Fig. 4 (a)).
5) For each cluster with centroid μ_k, compute as accumulation the histogram H_{μ_k} that holds the sum of the the responses of the single histograms of the sample points that fall in the cluster.
6) Re-compute the centroids μ_k with coordinates x_{μ_k}, y_{μ_k} equal the mean of the coordinates of the sample points assigned to the corresponding cluster and with α_{μ_k} corresponds to the directions with minimum response in histograms H_{μ_k}.
7) Repeat from point 3 until convergence.
8) For each cluster, compute the PSF extent as explained in section III-A
9) Divide the image in rectangular subregions with uniform PSFs based on their cluster centroids (Fig. 4 (b)).

We experimentally found out that $k = 2$ yields good results.

C. Finding distinctive features

After the PSF clustering step, we have a set of rectangular subregions characterized by a local PSF (Fig. 4 (b)): For each subregion we can now easily compute the adapted scale-space representation, as explained at the beginning of section III, using the local PSF. As in [13], the scale-space is divided in *octaves* (i.e., the last smoothed image of the octave has twice the scale of the first). Each octave is divided into an integer number s of intervals, with scales $\sigma_i = \sigma_{i-1} * 2^{\frac{1}{s}}$, where σ_0 is the initial scale chosen to be 1.6. We choose $s = 3$, so we compute Eq. (3) at scales $1.6, 2.0159, 2.5398, 3.2, 4.0317$. The latest scale is computed to detect local scale space maxima at the higher scale of the octave, i.e., 3.2. For each scale, we compute the adapted scale-space representation. Once an octave is completed, the image is re-sampled to half its original size: This image has obviously twice the scale of the original image. A new octave is then processed on the re sampled (smaller) image, using the same σ_i values. In order to detect interest points, the scaled images $L(x, y, \sigma)$ are convolved with filters that response mainly to invariant local features of the image. The scaled images L are computed according to the local PSF with the adapted scale-space representation explained before. As in [1], we use the determinant of the Hessian of the scaled image for selecting both location and characteristic scale of the interest points:

$$det(\sigma^2 H(x, y, \sigma)) = \qquad (11)$$
$$det \begin{bmatrix} \sigma^2 L_{xx}(x, y, \sigma) & \sigma^2 L_{xy}(x, y, \sigma) \\ \sigma^2 L_{xy}(x, y, \sigma) & \sigma^2 L_{yy}(x, y, \sigma) \end{bmatrix}$$

where in Eq. (III-C) L_{xx}, L_{yy}, L_{xy} are the second derivatives of the scaled images $L(x, y, \sigma)$. The second derivatives are multiplied with the square of the scale σ: this is due to the fact that the amplitude of spatial derivatives decreases with scale, so normalization is required for true scale invariance [12]). According to our experience, the determinant of the Hessian seems to be more stable with motion blurred images than Difference-of-Gaussian (DoG) used in [13]. Once computed the determinant of Hessian for each location of the multi-scaled image, interest point are detected searching for local maxima over scale and location space in a $3 \times 3 \times 3$ neighborhood of each point: only local maxima with determinant of Hessian greater than a threshold are selected as interest points. Finally, the location and the scale (called *characteristic scale*) of the extracted points are interpolated [5] by fitting a 3D quadratic to the scale-space determinant of Hessian and taking the maxima of this quadratic. This step is useful to obtain a more accurate characteristic scale of the point (negatively affected by the discrete nature of the scale space) and to reduce the localization errors. In our experiments, we noticed that the SIFT descriptor are slightly more stable than other approach. Detection of the interest points are then performed using the description method from SIFT [13]: we implement SIFT descriptor, tuning the parameters of the algorithm to improve reliability.

Motion blur effects tend to suppress the high frequency components, the resulting image so loses a lot of small details: It happens that some interest point extracted during detection step represents in reality very simple and not much distinctive features, that they can compromise the stability of the feature matching step producing more outliers. In order to avoid this issue, we introduce a discarding process based on the Shannon entropy of the normalized descriptor. If we take a normalized feature descriptor $s(i)$, we can see it as a probability mass function, with possible values $1, \ldots, n$ the indexes of the bins of the descriptor, in the case of SIFT descriptor $n = 128$. We can compute the entropy as:

$$H(s) = -\sum_{x=1}^{n} s(i) log(s(i)) \qquad (12)$$

We notice that simple and not much distinctive features tends to obtain descriptors with low entropy. For each descriptor, we first compute the entropy, then we compute the mean μ_H and the standard deviation σ_H of all entropy values. We finally discard all the descriptors with entropy values less than $\mu_H - \sigma_H$. This step improves noticeably the stability of the following features matching precess.

IV. Visual Odometry

The interest points are matched between pairs of frames using an efficient Best Bin First (BBF) algorithm [3] that finds an approximate solution to the nearest neighbor search problem. The algorithm is similar to the kd-tree search algorithm, where the tree is explored searching for the node that is closer to the input descriptor. The BBF algorithm only search m candidates, and returns the nearest-neighbor for a subset of queries. Given five corresponding points, it's possible to recover the relative positions of the points and cameras, up to a scale. This is the minimum number of points needed for estimating the relative camera motion from two calibrated views and it is called *five-point algorithm.* The Five-point algorithm offers many benefits compared with other relative pose problem solutions, as the well-known eight-point algorithm. The Five-point algorithm needs fewer correspondences to find a solution. Moreover, it is essentially unaffected by the planar degeneracy and it still works for planar scenes where other methods fail. The estimation accuracy of the five-point algorithm is also higher than other solutions to the relative pose problem. In our visual odometry approach, we use the efficient solution to the five-point relative pose problem[1] proposed by Nister [20]. We assume that the camera used in the visual odometry is fully calibrated, i.e., intrinsic matrix K is given. For a static scene point projected in two views, we can write:

$$m'^T F m = 0 \tag{13}$$

where F is a fundamental matrix and m and m' are the image points expressed in homogeneous coordinates for the first and second view, respectively. If the camera is calibrated, the fundamental matrix is reduced to an essential matrix, denoted by E, and the relationship becomes;

$$q'^T E q = 0 \tag{14}$$

with $q = K^{-1}m$ and $q' = K^{-1}m'$. Using the five-point algorithm with five correspondences q_i, q'_i, $i = 1, \ldots, 5$, one can obtain at most ten possible essential matrices (including complex ones) as solutions of the problem. For each essential matrix four combinations of possible relative rotation R and translation T of the camera can be easily extracted [20]. In order to determine which combination corresponds to the true relative movement, the constraint that the scene points should be in front of the camera for both the two views is imposed. The image points are triangulated into 3D points [9] using all the combination of R and T. The final solution is identified as configuration more compliant with the given constraints. We use the five-point algorithm in conjunction with MLESAC estimator [23]: MLESAC uses the same sampling strategy as RANSAC where minimal sets of correspondences (5 in our case) are used to derive hypothesized solutions. The remaining correspondences are used to evaluate the quality of each hypothesis. Unlike RANSAC, that count the number of inliers, MLESAC evaluates the likelihood of the hypothesis by representing the error distribution as a mixture model. Our mono-camera visual odometry scheme operates as follows:

1) Extract the features from the images using the proposed features detection and descriptor scheme.
2) Track interest points over two frames using the BBF matching strategy.
3) Randomly chose a number of samples each composed of 5 matches between the first and the second frame. Using the five-point algorithm generate a number of hypotheses for the essential matrix.
4) Search for the best hypotheses using MLESAC estimator and store the correspondent inliers. The error function is the distance between the epipolar line E_q associated with q and p',
5) Extract from the resulting essential matrix E the four combinations of possible relative rotation R and translation T. Triangulate all the inlier correspondences for each combination. Take the configuration with more 3D points in front of both camera views.
6) If this is not the first time inside the loop, select the features tracked in the present reconstruction, that were tracked also in the previous one, and compute using triangulation the depth for both reconstruction. Use these information in conjunction with RANSAC to estimate the scale factor between the present reconstruction and the previous. Put the present reconstruction in the coordinate system of the first reconstruction.
7) Repeat from Point 1.

V. Results

We implemented our detection-descriptor scheme in C++ using the efficient OpenCV image processing library[2]: the whole process take on average 1 second for a 640X480 image on a 2Ghz core 2 PC.
As experimental platforms, we used the custom built NimbRo Robocup Team humanoid robot and the commercial Kondo KHR-1 HV humanoid robot.

A. Performance evaluation of the proposed features detection-descriptor scheme

We compared our detection and descriptor scheme to the SIFT features [13] and to the SURF-128 features [1] (i.e., the improved version of the SURF features). Comparisons are performed using well-known implementation of these methods[3],[4] without changing the standard parameters of the algorithms. The input data are from two sources: (i) a standard dataset[5] with added synthetic motion blur, (ii) sequences of images grabbed by the CMOS camera of a walking humanoid robot affected by real motion blur effect. Testing image pairs are composed by two images of the same scene taken from different viewpoint. One or both the images are affected by motion blur. The standard dataset we used is provided with *homographies* (plane projective transformations) between images: the map between the two images is known, the exact correspondence of every point in one frame to the corresponding points in the other frame is known. We can determine in this case ground truth matches and also the accuracy (i.e., the localization error of the matches). For the real images set, we manually identified the correct matches between frames. For all tested approaches, we use the Nearest Neighbor Distance Ratio matching strategy (see [17]), with distance ratio equal to 0.5. Fig. 5 presents matching results for image pairs of the standard dataset. One or both the image images of the pairs are blurred with synthetic motion-blur functions with different directions and extent variable between 10 an 40 pixels. The matching accuracy is the distance in pixels between the ground truth

[1] We use the five-point algorithm implementation provided with the VW34 library by Oxford's Active Vision Lab.

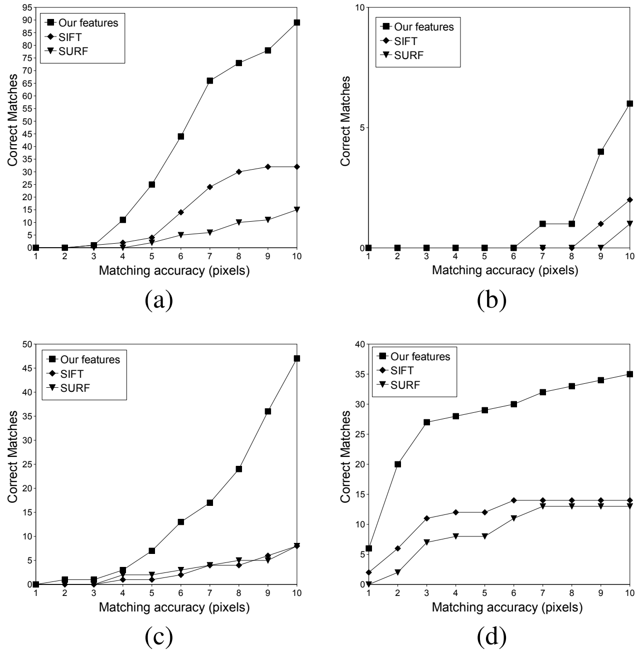
[2] http://sourceforge.net/projects/opencvlibrary/

[3] http://www.cs.ubc.ca/˜lowe/keypoints/

[4] http://www.vision.ee.ethz.ch/˜surf/

[5] http://www.robots.ox.ac.uk/˜vgg/research/affine/

match and the obtained match. As can be seen, our approach outperforms SIFT and SURF-128 in both the number of correct matches and the localization accuracy. This is very important especially in visual odometry tasks, where the accuracy in matching affect significantly results in motion estimation. Results for some real images are presented in Fig. 6: Here, the x-axis represents the image pairs used in matching process. As the results demonstrate, also with real images our approach outperforms the others with respect to the higher number of correct matches.



(a) (b)

(c) (d)

Fig. 5. Correct matches for the standard dataset images. (a) Image pair 1 and 3 of the *graf* series. (b) Image pair 1 and 6 of the *boat* series. (c) Image pair 1 and 6 of the *trees* series. (d) Image pair 1 and 6 of the *leuven* series. Accuracy is the distance in pixels between the ground truth match and the obtained match.

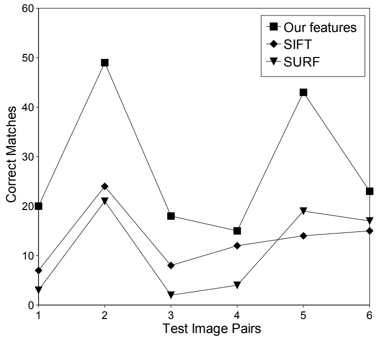


Fig. 6. Correct matches for the real image sets. In the x-axis, the corresponding image pairs are denoted. The y-axis indicates the number of correct matches. Our approach finds much more feature correspondences compared to SIFT and SURF.

B. *Testing visual odometry*

We tested our visual odometry framework with trajectories walked by humanoid robots. The accuracy is measured by checking the error between the start and the endpoints of the recovered trajectory (Fig. 7). The path of Fig. 7 (a) is a closed loop in which the starting point and the end point are the same point in the environment. The robot walked a loop of about 4-5 m in diameter in the cluttered environment of our laboratory. In the path of Fig. 7 (b) the robot walked down a corridor for 5 m, it turned around, and it walked back to almost the same position. Unfortunately, it was not possible to record the ground-truth of the robot, but the robot path was closely surveilled and in particular the start and the end points. Despite the fact that the paths of Fig. 7 are calculated up to a scale, one can see that the proposed visual odometry can reliably estimate the motion of the robot, even it is not so accurate when the robot is turning. This is the reason why the start and end points do not overlap in Fig. 7 (a) and the mutual distance is a bit too large in Fig. 7 (b). In fact, the reconstructed paths are open-up because the robot rotation was underestimated. Unfortunately, we cannot report a comparison with SIFT and SURF approaches on this visual odometry experiment, because both approaches did not pick enough features in the image sequence to reliable reconstruct the path. Indeed, the unmodified environments in which we performed the experiments where quite dim and most of the surfaces did not have bright patterns. Moreover, as reported in Fig. 6, the motion blur affecting the images in the walking sequence lowered even more the number of feature detected by SURF and SIFT approaches.

VI. CONCLUSIONS

In this paper, we presented a novel framework for visual odometry with a single camera robust even to non-uniform motion blur. We developed an improved feature detector based on SIFT that can find good correspondences even in heavily blurred images, such as the ones grabbed by robots performing brisk movements. The proposed method outperforms the SIFT and the SURF approach in detecting and matching corresponding features between two images in which one image or both of them are corrupted with motion blur. We evaluated our method on images taken from standard datasets and on images grabbed by walking humanoid robots. As a final validation, we reported experiments for successful visual odometry estimation for small humanoid robots walking a 10 m, respectively a 20 m path. Due to space constraints, we could not report experiments on images without motion blur in which our approach can find the same (and more) correct matches than SIFT and SURF.

Future work includes the integration of the presented visual odometry strategy as a priori motion prediction estimation into a visual SLAM approach. This is well suited humanoid robots, where as correction step more global and topological visual information can be taken into account. Moreover, the feature matching and tracking phase can certainly be improved by mounting an inertial measurement

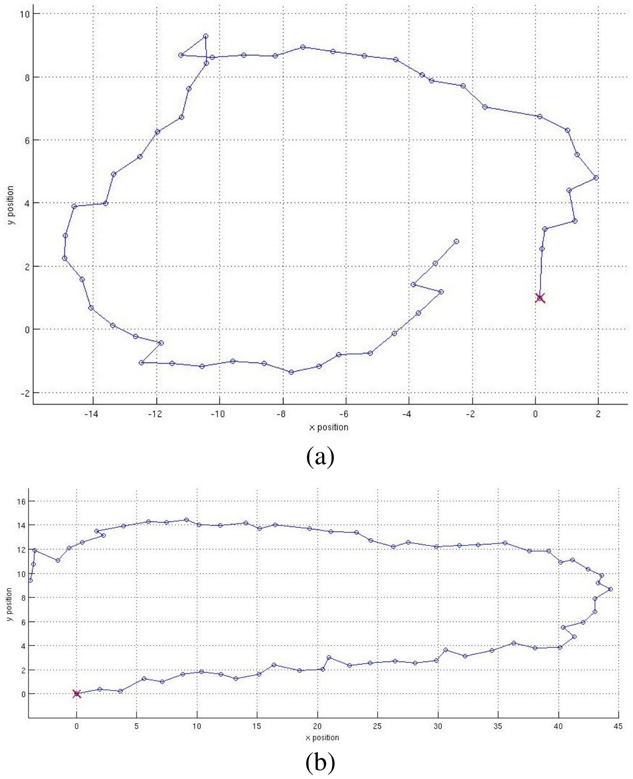


Fig. 7. Estimation of the robot motion using the proposed visual odometry framework for two closed trajectories. The red crosses are the start points of the trajectories.

unit on the humanoid robot that can provide a first guess on the motion of the camera.

VII. Acknowledgments

We wish to thank Jörg Stückler for his help in collecting data with a humanoid robot of Team NimbRo.

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proc. of the ninth European Conference on Computer Vision (ECCV)*, 2006.

[2] S. Behnke, M. Schreiber, J. Stückler, R. Renner, and H. Strasdat. See, walk, and kick: Humanoid robots start to play soccer. *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 497–503, Dec. 2006.

[3] J. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, 1997.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[5] M. Brown and D. Lowe. Invariant features from interest point groups. In *BMVC02*, 2002.

[6] A.I. Comport, E. Malis, and P. Rives. Accurate quadrifocal tracking for robust 3d visual odometry. In *IEEE International Conference on Robotics and Automation*, pages 40–45, 2007.

[7] Andrew J. Davison, Ian D. Reid, , Nicholas D. Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1–16, 2007.

[8] J. Flusser, J. Boldys, and B. Zitova. Moment forms invariant to rotation and blur in arbitrary number of dimensions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(2):234–246, Feb 2003.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[10] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proc. of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[11] Georg Klein and Tom Drummond. A single-frame visual gyroscope. In *Proc. British Machine Vision Conference (BMVC'05)*, volume 2, pages 529–538, Oxford, September 2005. BMVA.

[12] Tony Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270, 1994.

[13] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[14] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79.

[15] C. Mei and I. Reid. Modeling and generating complex motion blur for real-time tracking. In *Computer Vision and Pattern Recognition*, June 2008.

[16] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[17] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[18] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Realtime localization and 3d reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition*, page 10271031, 2006.

[19] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proc. of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[20] David Nister. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.

[21] A. Pretto, E. Menegatti, and E. Pagello. Reliable features matching for humanoid robots. *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, Dec. 2007.

[22] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume 1. Academic, 1982.

[23] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[24] Y. Yitzhaky, I. Mor, A. Lantzman, and N. S. Kopeika. Direct method for restoration of motion-blurred images. *Journal of the Optical Society of America A*, 15:1512–1519, June 1998.