# Combining Harris Interest Points and the SIFT Descriptor for Fast Scale-Invariant Object Recognition

Pedram Azad, Tamim Asfour, Rüdiger Dillmann

*Institute for Anthropomatics, University of Karlsruhe, Germany*

*azad@ira.uka.de*, *asfour@ira.uka.de*, *dillmann@ira.uka.de*

***Abstract*— In the recent past, the recognition and localization of objects based on local point features has become a widely accepted and utilized method. Among the most popular features are currently the SIFT features, the more recent SURF features, and region-based features such as the MSER. For time-critical application of object recognition and localization systems operating on such features, the SIFT features are too slow (500–600 ms for images of size 640×480 on a 3 GHz CPU). The faster SURF achieve a computation time of 150–240 ms, which is still too slow for active tracking of objects or visual servoing applications. In this paper, we present a combination of the Harris corner detector and the SIFT descriptor, which computes features with a high repeatability and very good matching properties within approx. 20 ms. While just computing the SIFT descriptors for computed Harris interest points would lead to an approach that is not scale-invariant, we will show how scale-invariance can be achieved without a time-consuming scale space analysis. Furthermore, we will present results of successful application of the proposed features within our system for recognition and localization of textured objects. An extensive experimental evaluation proves the practical applicability of our approach.**

## I. INTRODUCTION

In the recent past, the recognition and localization of objects based on local point features has become a widely accepted and utilized method. Among the most popular features are currently the SIFT features (Scale Invariant Feature Transform) [1], [2], the more recent SURF features (Speeded Up Robust Features) [3], and region-based features such as the MSER (Maximally Stable Extremal Regions)[4]. The most popular interest point operators are the Harris corner detector [5] and the *Good Features to Track* [6], also referred to as Shi-Tomasi features.

The main task of matching features that are defined by interest points is to achieve invariance to the mentioned changes. In this context, the term feature *descriptor* is often used, denoting the data structure that is compared in order to calculate the similarity between two feature points. Various methods have been proposed for this purpose. In [7], an approach is presented using a rotationally symmetric Gaussian window function to calculate a moment descriptor. In [8], *local jets* according to [9] are used to compute multiscaled differential grayvalue invariants. In [10], two types of affinely invariant regions are proposed: one based on the combination of interest points and edges, and the other based on image intensities. In [3], a speeded up approach named SURF is presented, using a fast Hessian detector and gradient-based descriptor.

In [11], the performance of five types of local descriptors is evaluated: SIFT, steerable filters [12], differential invariants [9], complex filters [13], and moment invariants [14]. In all tests, except for light changes, the SIFT descriptor outperforms the other descriptors.

In [15], an object recognition system with a database of 50 objects is presented, which uses the Gabor wavelet transformation around Shi-Tomasi interest points in order to calculate a feature descriptor. k-means clustering is used to reduce the number of features stored in the database. Murphy-Chutorian and Triesch show empirically that for their test database, 4,000 shared features are the optimal tradeoff between computation time (27 s) and detection rate (79%). Without feature sharing, the storage and comparison of 160,000 independent features would be required.

A completely different approach for point matching is presented in [16]. Instead of calculating a descriptor analytically to achieve invariance, robustness to scaling, rotation, and skew is achieved in a brute-force manner. Each image patch around a point feature is represented by a set of synthetically generated different views of the same patch, intended to cover all possible views. In order to speedup matching, PCA is applied to all view sets. Point matching is performed by calculating the nearest neighbor in the eigenspace for a given image patch. The complete process takes about 200 ms for a single frame on a 2 GHz CPU.

This type of feature representation was used in our previous work [17]. It was shown that through combination with the idea of applying k-means clustering from [15] an object can be recognized within 350 ms, using a database consisting of 20 objects. However, the learning procedure is very time consuming (approx. 20 hours for 20 objects) due to the computation of the covariance matrix for PCA computation and the subsequent k-means clustering. More importantly, such an approach does not allow incremental updates of the database, since the PCA must be computed for all features, as well es k-means clustering.

In this paper, we will present our novel types of features, which combine the Harris corner detector with the SIFT descriptor[1]. In order to achieve scale-invariance in spite of omitting the scale space analysis step of the SIFT features,

[1]Note: The unpublished term *Harris-SIFT* that can be found on the internet has nothing to do with the proposed features and describes a completely different approach.

the features are computed at several predefined spatial scales explicitly. A thorough analysis of the scale coverage of the SIFT descriptor and the proposed extension justifies the choice of the involved parameters. Furthermore, we will present our 2D object recognition system that uses the proposed features. Experimental results show that the proposed features are computed within approx. 20 ms on images of resolution 640×480 and allow robust real-time recognition and localization of a single object at frame rates of 30 Hz using conventional hardware.

The work presented in this paper is part from [18]. In parallel, Wagner et al. have developed a similar approach based on the same idea, using a combination of the SIFT descriptor and Ferns descriptor [19] together with the FAST detector [20], as presented in [21]. In this paper, the original SIFT descriptor is combined with the Harris corner detector, and all parameters are derived from a thorough analysis of the scale coverage of the SIFT descriptor.

## II. Feature Calculation

In this section, the developed feature calculation method is presented. As already stated, our experiments proved that the SIFT descriptor is a very robust and reliable representation for the local neighborhood of an image point. However, the scale-space analysis required for the calculation of the SIFT feature point positions is too slow for visual servoing applications. As stated in [3], the computation of the SIFT features for an image of size 800×640 takes approx. 1 s (using a Pentium IV, 3 GHz). This scales to about 0.6 s for the resolution of 640×480. The SURF features require approx. 0.15–0.24 s (depending on the SURF variant) on the same image size. The goal was to find a method that allows feature calculation in approx. 20 ms for an image of size 640×480.

One of the main strengths of the SIFT features are their scale-invariance. This is achieved by analyzing and processing the images at different scales. For this, a combination of Gaussian smoothing and a resize operation is used. Between two so-called octaves, the image size is halved, i.e. resized to half width and half height. The different scales within an octave are produced by applying a Gaussian smoothing operator, and the variance of the Gaussian kernel is chosen in a way that the last scale of one octave and the first scale of the next octave correspond to each other.

Since the scale space analysis performed by the SIFT features for calculating the feature point positions is the by far most time-consuming part, the idea is to replace this step by a faster method, namely an appropriate corner detector. As shown in [22], the Harris corner detector is a suitable starting point for the computation of positions of scale and affine invariant features. In [22], the Harris-Laplace detector, which is based on the Harris corner detector, is extended to the so-called Harris-Affine detector, which achieves affine invariance.

However, the computational effort for the calculation of the Harris-Laplace or even more the Harris-Affine features is again too high for visual servoing applications. Therefore, the goal was to investigate if it is possible to combine the conventional Harris corner detector with the SIFT descriptor, while keeping the property of scale-invariance.


Fig. 1. Image used for evaluation of the scale coverage of the SIFT descriptor. For this image, 284 feature points were calculated by the Harris corner detector, using a quality threshold of 0.01. The computed feature points are marked by the green dots.

As a first step, the scale coverage of the SIFT descriptor computed with a fixed window size of 16×16 was evaluated. For this, the Harris corner points were calculated for the image from Fig. 1 and stored as a set $\{\boldsymbol{x}_i\}$ with $i \in \{1, \ldots, n\}$ and $\boldsymbol{x}_i \in \mathbb{R}^2$. The image was then resized with bilinear interpolation to different scales $s \in [0.5, 2]$. At each scale $s$, the stored corner point locations were scaled, i.e. $\boldsymbol{x}_i^{(s)} = s\,\boldsymbol{x}_i$, so that ground truth for the correspondences is given by $\boldsymbol{x}_i^{(s)} \sim \boldsymbol{x}_i$. For each feature in the scaled image, the best matching feature in the set $\{\boldsymbol{x}_i\}$ was determined. In Fig. 2, the resulting percentages of correct matches at the different scales are plotted. In order to see the symmetry of the scale coverage, a $\frac{1}{s}$ scale was used for the part of the $s$-axis left of 1.0.

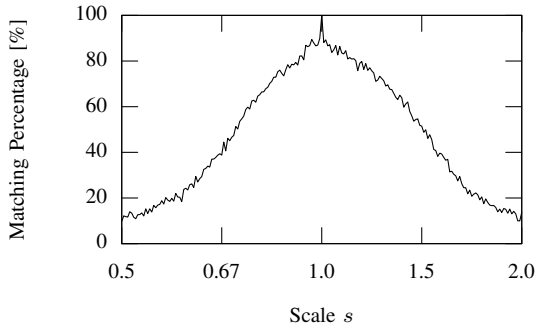


Fig. 2. Plot of the scale coverage of the SIFT descriptor. The evaluation was performed on image scales computed by resizing with bilinear interpolation.

As can be seen in Fig. 2, the matching robustness of the SIFT descriptor is very high (>80%) within a range of approx. 10–15 %. Therefore, it must be possible to close the gap between two scales by exploiting the scale coverage of the SIFT descriptor only, if the scales are close enough to each other. In other words: The idea is that a time-consuming scale space analysis based on a scale space representation using Gaussian filtering can be omitted, and instead a suitable scale factor is used for computing predefined scales using a resize operation with bilinear interpolation. For the

conventional SIFT features, the scale factor between two consecutive octaves is 0.5. The question is now, what is a suitable scale factor $\Delta s$ with $0.5 < \Delta s < 1$ when omitting the scale-space analysis and closing the gap between adjacent spatial scales by exploiting the scale coverage of the SIFT descriptor only?

In Fig. 3, the matching percentages for the same experiment as before are plotted, this time computing SIFT descriptors at multiple predefined scales. Three scales were used for producing the SIFT descriptors, i.e. $(\Delta s)^0$, $(\Delta s)^1$, and $(\Delta s)^2$. As before, the Harris corner points were only calculated once for the original image, and the image locations were scaled for calculating the SIFT descriptor at the lower scales. Note that this is only done for comparison purposes; for normal application, the interest points are re-calculated at the lower scales to avoid the computation of dispensable features. The peaks at 100 % occur when $(\Delta s)^i = s$, i.e. the features to be matched are computed on the exact same image.

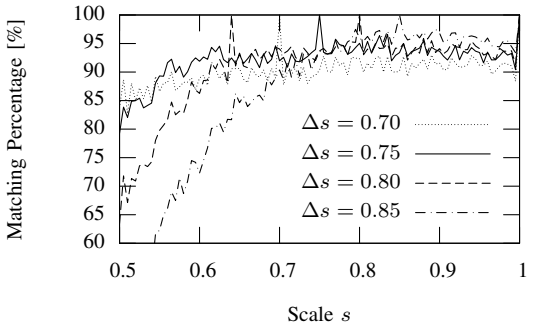


Fig. 3. Plot of the scale coverage when using SIFT descriptors at multiple scales. The evaluation was performed on image scales computed by resizing with bilinear interpolation. Three levels were used; the parameter $\Delta s$ denotes the scale factor between two consecutive levels.

As can be seen, the scale factors $\Delta s = 0.75$ and $\Delta s = 0.8$ essentially achieve the same performance within the interval $[0.6, 1]$. For the scales smaller than 0.6, $\Delta s = 0.75$ is superior, as expected. Within the interval $[0.8, 1]$, $\Delta s = 0.85$ achieves the best results. However, the performance decreases rapidly for scales smaller than 0.7, since only three levels are used. Within the interval $[0.7, 1]$, $\Delta s = 0.7$ achieves the worst results. The strengths become visible at the smaller scales. However, this can be also achieved by using a larger $\Delta s$ and an additional fourth level if necessary, while the inferior performance of $\Delta s = 0.7$ for the crucial higher scales cannot be improved. Judging from theses results, $\Delta s = 0.75$ is a good tradeoff between a high matching performance and a high coverage.

Finally, the extended Harris-SIFT features must prove to perform as well when applied in practice, i.e. the training view and the current view are acquired using different setups. The two images used for the following experiment are shown in Fig. 4. The training view on the very right is the same as shown in Fig. 1; it is included again only for illustrating the scale differences. The features were tested on the image


Fig. 4. Images used for testing the performance of the extended Harris-SIFT features. The computed feature points are marked by the white dots. Left: view corresponding to a scale of 0.32 relative to the training view, with 438 computed feature points. Middle: view corresponding to a scale of 0.64 relative to the training view, with 500 computed feature points. Right: training view, with 284 computed feature points.

shown in the middle of Fig. 4, which contains the object at a scale of approx. 0.64. For the tests, this image was resized to scales from $[0.5, 1]$, i.e. the smallest effective scale of the object was $0.5 \cdot 0.64 = 0.32$ (see left image from Fig. 4), compared to the training view.

In Fig. 5, the total number of successfully matched interest points at each scale for this experiment is plotted. Note that according to [1], for each point, several SIFT descriptors are computed, if the calculated orientation tends to be ambiguous. In order to not falsify the results by counting several matches for a single interest point, for each interest point at most one match was counted. By doing this, the resulting plot shows what counts for recognition and pose estimation: the number of successfully matched image locations. The plot shows the results for $\Delta s = 0.75$, using 3, 4, and 5 levels, respectively. The maximum number of interest points was restricted to 500. For the computation of the SIFT descriptor, a fixed window size of 16×16 was used.

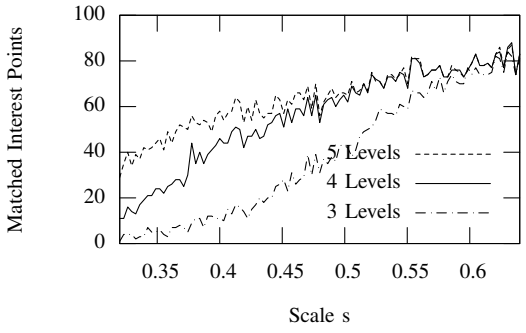


Fig. 5. Illustration of the performance of the extended Harris-SIFT features for the views shown in Fig. 4. As scale factor, $\Delta s = 0.75$ was used. The plot shows the total number of successfully matched interest points at each scale $s$, where $s$ is understood in relation to the object's size in the training image.

As can be seen, using four or five levels leads to the same results within the interval $[0.47, 1]$. When using three levels, the performance starts to decrease noticeably at approx. 0.57. For the performed experiments, three levels were used for the training views, which proved to be fully sufficient when using images of size 640×480. Note that, in practice, often the limiting factor is the effective resolution of the object in the image, and not the theoretical scale invariance of the features.

## III. Recognition and 2D Localization

In this section, our recognition and 2D localization system, in which the proposed features are applied, is summarized briefly. The approach is a variant of Lowe's framework [1]; the main differences are the voting formula for the Hough transform and the final optimization step using a full homography. Details are given in [18].

The feature information used in the following is the position $(u, v)$, the rotation angle $\varphi$ and the feature vector $\{\boldsymbol{f}_j\}$ consisting of 128 floating point values in the case of the SIFT descriptor. These feature vectors are matched with those of the features stored in the database using a nearest neighbor approach. For recognizing objects on the basis of point feature correspondences, an approach consisting of three steps is used:

A) Hough transform
B) RANSAC
C) Least squares homography estimation

### A. Hough Transform

In the first step, a two-dimensional Hough space with the parameters $u, v$ is used; the rotational information $\varphi$ and the scale $s$ are used within the voting formula. In contrast to [1], the scale $s$ is not taken from the features but votes are cast at several scales [23], since the scale is not computed by the Harris-SIFT features.

Given a feature in the current scene with $u, v, \varphi$ and a matched feature from the database with $u', v', \varphi'$, the following bins of the Hough space are incremented:

$$\begin{pmatrix} u_k \\ v_k \end{pmatrix} = r \left[ \begin{pmatrix} u \\ v \end{pmatrix} - s_k \begin{pmatrix} \cos\Delta\varphi & -\sin\Delta\varphi \\ \sin\Delta\varphi & \cos\Delta\varphi \end{pmatrix} \begin{pmatrix} u' \\ v' \end{pmatrix} \right] \tag{1}$$

where $\Delta\varphi := \varphi - \varphi'$ and $s_k$ denotes a fixed number of discrete scales. According to the results of the extended Harris-SIFT features for $\Delta s = 0.75$ and using three levels (see Fig. 5), $s_k := 0.5 + k \cdot 0.1$ with $k \in \{0, \ldots, 5\}$ was used for the performed experiments. The parameter $r$ is a constant factor denoting the resolution of the Hough space.

After the voting procedure, potential instances of an object in the scene are represented by maxima in the Hough space. The set of correspondences is then filtered by only considering those correspondences that have voted for a maximum or cluster of interest.

### B. RANSAC

In the second step, a RANSAC approach is applied using the filtered set of correspondences from the previous step. The RANSAC algorithm allows to filter outliers, which could potentially lead to a wrong local minimum throughout the least squares approach for accurate homography estimation in the third step. For the error tolerance, 5 pixels are used and a fixed number of 200 iterations.

### C. Least Squares Homography Estimation

For the filtered set of feature correspondences resulting from the RANSAC algorithm, now a full homography is estimated with a least squares approach. First, in an iterative procedure, an affine transformation is computed, filtering outliers in each iteration. In the final step a full homography is estimated to allow for maximum accuracy.

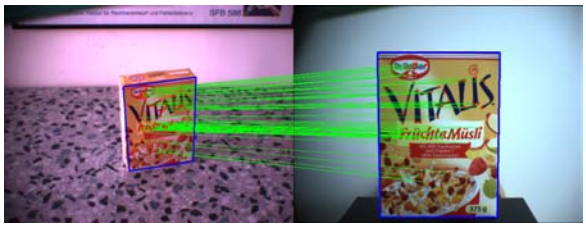


Fig. 6. Filtered feature correspondences after iterative computation of the affine transformation.

If after the complete process of homography estimation, a certain number of feature correspondences is remaining and the mean error is smaller than a predefined threshold, an instance of the object is declared as recognized. The final, filtered set of feature correspondences for an example scene is illustrated in Fig. 6. The 2D localization is given by the transformation of the contour in the training view to the current view.

## IV. Run-time Considerations

As described in Section II, throughout the experiments three levels were used with a scale factor of $\Delta s = 0.75$. However, when assuming that the object never appears larger than the largest training view, then multiple levels are not needed for feature computation on the current view. It is sufficient to use multiple levels for the training view, so that the object can be recognized at smaller scales. This strategy significantly reduces the number of feature comparisons and therefore the run-time of the matching procedure.

The computation of the nearest neighbor for the purpose of feature matching is the most time-consuming part of the complete recognition and localization algorithm. To speedup the nearest neighbor computation, a kd-tree is used to partition the search space; one kd-tree is built for each object. In order to perform the search efficiently, the Best Bin First (BBF) strategy [24] is used. This algorithm performs a heuristic search and only visits a fixed number of $n_l$ leaves. The result is either the actual nearest neighbor, or a data point close to it. The parameter $n_l$ depends on the number of data points i.e. SIFT descriptors: The more SIFT descriptors the kd-tree contains, the greater $n_l$ must be to achieve the same reliability. Since each kd-tree only contains the features of one object, $n_l$ can be chosen to be relatively small. Throughout the experiments, $n_l = 75$ was used for feature sets consisting of not more than 1,000 features.

## V. Experimental Results

In this section, results of experiments for the evaluation of repeatability, accuracy, and speed are presented. The repeatability of the proposed features equals the repeatability of the Harris corner points within a scale interval of approx. $[0.87,\ 1]$, when using a scale factor of $\Delta s = 0.75$ ($\sqrt{0.75} \approx 0.87$). For measuring the repeatability, the

right image from Fig. 6 was rotated and scaled, and the Harris corner points were computed both on the original and the result image. The repeatability measure was computed with the formula given in [22]. When applying the Harris corner detector, three parameters are important: the quality threshold, the minimal distance between two feature points, and the maximal number of feature points to be calculated. Throughout all experiments, we used a minimal distance of 5 pixels. The quality threshold was set to 0.001 in order to produce many features. Fig. 7 shows the results for 500 and 1200 feature points, where 1200 was the maximum number of features that could be calculated with the chosen parameters. As can be seen, the repeatability at the scale 0.87 amounts to 73% for 500 points and 84% for 1200 points. Note that it is impossible to provide one representative value of the repeatability for a specific scale, since the repeatability always depends on the provided parameters and the image data.

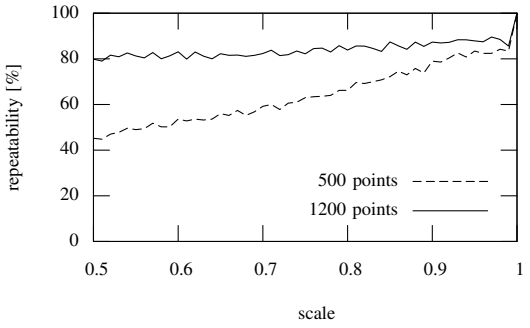


Fig. 7. Results of repeatability experiments.

The performance of the proposed features within our object recognition system and localization system is shown in Fig. 8. A difficult scene with skew and a low effective resolution of the object of interest was chosen. The 2D error was computed as the mean projection error into the current image. As can be seen, a low quality threshold for the Harris corner detector should be used.

The processing times given in Table I were computed using a trained object representation containing 700 SIFT descriptors and 230 SIFT descriptors were extracted from the current view. The processing times for matching and for homography estimation scales linearly with the number of trained objects. Furthermore, the matching time scales linearly with the number of features extracted from the current view. The system was implemented using the Integrating Vision Toolkit (IVT)[2], which, among many other features, offers a fast Harris corner detector (compared to OpenCV 1.0: 10 ms vs. 17 ms) and an efficient kd-tree implementation. The company *keyetech*[3] offers highly optimized implementations (e.g. Harris corner detection within less than 5 ms or nearest neighbor computation).

[2]http://ivt.sourceforge.net
[3]http://www.keyetech.de

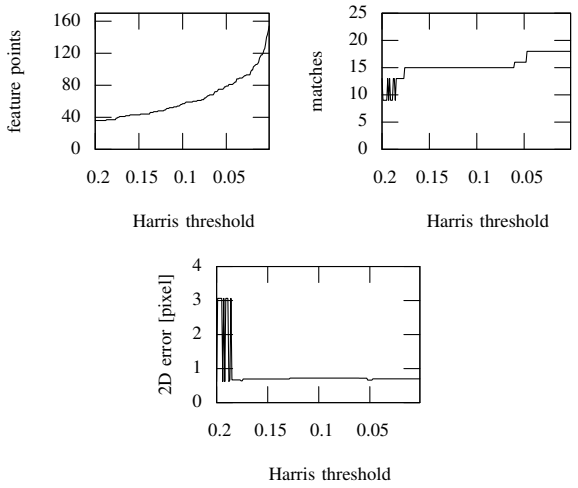


Fig. 8. Effect of the Harris quality threshold for an example with a low resolution of the object. One learned view was used containing 700 feature descriptors. The computation time of the Harris corner points took 13 ms in all cases.

Finally, exemplary recognition results on real image data aquired by the humanoid robot ARMAR-III [25] operating in a kitchen environment are shown in the Fig. 9 and 10. The video attachment shows the results of processing an image sequence with a moving object.

| | Time [ms] |
|---|---|
| Harris corner detection | 10 |
| SIFT descriptor computation | 6 |
| Matching | 12 |
| Iterative homography estimation | 3 |
| **Total** | **31** |

TABLE I

PROCESSING TIMES FOR THE PROPOSED OBJECT RECOGNITION AND LOCALIZATION SYSTEM. THE OBJECT REPRESENTATION CONSISTED OF 700 DESCRIPTORS AND THE CURRENT VIEW CONTAINED 230 DESCRIPTORS. THE TESTS WERE PERFORMED ON A 3 GHZ CORE 2 DUO.

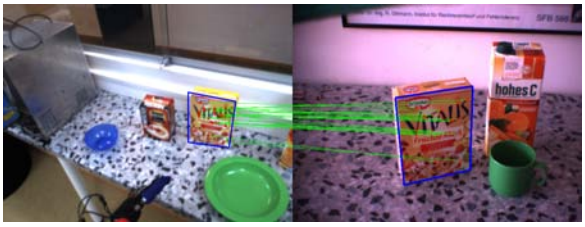

Fig. 9. Computed feature correspondences and recognition result for a difficult scene, featuring out-of-plane rotation and a low effective resolution of the object of interest.

## VI. DISCUSSION

We have presented a novel type of point feature, which combines the Harris corner detector with the SIFT descriptor. It was shown how scale-invariance can be achieved efficiently and effectively without a time-consuming scale

Fig. 10. Exemplary results with the proposed object recognition and localization system.

space analysis. Furthermore, the integration of the proposed features in our object recognition and localization system has been presented.

Results from experiments on simulated image data as well as on real image data from the humanoid robot ARMAR-III operating in a kitchen environment proved the practical applicability and performance of the proposed features. The features are computed within approx. 20 ms for an image of resolution 640×480; with the proposed system a single object can be tracked in real-time at frame rates of 30 Hz.

## ACKNOWLEDGMENT

The work described in this paper was partially conducted within the EU Cognitive Systems projects PACO-PLUS (IST-FP6-IP-027657) and GRASP (IST-FP7-IP-215821) funded by the European Commission, and the German Humanoid Research project SFB588 funded by the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft).

## REFERENCES

[1] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," in *IEEE International Conference on Computer Vision (ICCV)*, Kerkyra, Greece, 1999, pp. 1150–1517.

[2] ——, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[3] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," in *European Conference on Computer Vision (ECCV)*, Graz, Austria, 2006, pp. 404–417.

[4] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," in *British Machine Vision Conference (BMVC)*, vol. 1, London, UK, 2002, pp. 384–393.

[5] C. G. Harris and M. J. Stephens, "A Combined Corner and Edge Detector," in *Alvey Vision Conference*, Manchester, UK, 1988, pp. 147–151.

[6] J. Shi and C. Tomasi, "Good Features to Track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, 1994, pp. 593–600.

[7] A. Baumberg, "Reliable Feature Matching Across Widely Separated Views," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Hilton Head, USA, 2000, pp. 1774–1781.

[8] C. Schmid and R. Mohr, "Local Grayvalue Invariants for Image Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 19, no. 5, pp. 530–535, 1997.

[9] J. J. Koenderink and A. J. van Doorn, "Representation of Local Geometry in the Visual System," *Biological Cybernetics*, vol. 55, pp. 367–375, 1987.

[10] T. Tuytelaars and L. V. Gool, "Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions," in *British Machine Vision Conference (BMVC)*, Bristol, UK, 2000.

[11] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, Madison, USA, 2003, pp. 257–263.

[12] W. Freeman and E. Adelson, "The Design and Use of Steerable Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 13, no. 9, pp. 891–906, 1991.

[13] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets," in *European Conference on Computer Vision (ECCV)*, Copenhagen, Denmark, 2002, pp. 414–431.

[14] L. V. Gool, T. Moons, and D. Ungureanu, "Affine / Photometric Invariants for Planar Intensity Patterns," in *European Conference on Computer Vision (ECCV)*, vol. 1, Cambridge, UK, 1996, pp. 642–651.

[15] E. Murphy-Chutorian and J. Triesch, "Shared features for Scalable Appearance-based Object Recognition," in *Workshop on Applications of Computer Vision (WACV)*, Breckenridge, USA, 2005, pp. 16–21.

[16] V. Lepetit, J. Pilet, and P. Fua, "Point Matching as a Classification Problem for Fast and Robust Object Pose Estimation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, Washington, DC, USA, 2004, pp. 244–250.

[17] K. Welke, P. Azad, and R. Dillmann, "Fast and Robust Feature-based Recognition of Multiple Objects," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Genova, Italy, 2006, pp. 264–269.

[18] P. Azad, "Visual Perception for Manipulation and Imitation in Humanoid Robots," Ph.D. dissertation, Universität Karlsruhe (TH), Karlsruhe, Germany, 2008. [Online]. Available: http://digbib.ubka.uni-karlsruhe.de/volltexte/1000011294

[19] M. Özuysal, P. Fua, and V. Lepetit, "Fast Keypoint Recognition in Ten Lines of Code," in *European Conference on Computer Vision (ECCV)*, Graz, Austria, 2006, pp. 430–443.

[20] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," in *European Conference on Computer Vision (ECCV)*, Graz, Austria, 2006, pp. 430–443.

[21] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose Tracking from Natural Features on Mobile Phones," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, Cambridge, UK, 2008, pp. 125–134.

[22] K. Mikolajczyk and C. Schmid, "Scale & Affine Invariant Interest Point Detectors," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 1, pp. 63–86, 2004.

[23] P. Azad, T. Asfour, and R. Dillmann, "Stereo-based 6D Object Localization for Grasping with Humanoid Robot Systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, USA, 2007, pp. 919–924.

[24] J. S. Beis and D. G. Lowe, "Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Juan, Puerto Rico, 1997, pp. 1000–1006.

[25] T. Asfour, K. Regenstein, P. Azad, J. Schröder, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Genova, Italy, 2006, pp. 169–175.