Comparison of Local Image Descriptors for Full 6 Degree-of-Freedom Pose Estimation

Fredrik Viksten, Per-Erik Forssén, Björn Johansson, and Anders Moe

***Abstract*— Recent years have seen advances in the estimation of full 6 degree-of-freedom object pose from a single 2D image. These advances have often been presented as a result of, or together with, a new local image descriptor. This paper examines how the performance for such a system varies with choice of local descriptor. This is done by comparing the performance of a full 6 degree-of-freedom pose estimation system for fourteen types of local descriptors. The evaluation is done on a database with photos of complex objects with simple and complex backgrounds and varying lighting conditions. From the experiments we can conclude that *duplet features*, that use pairs of interest points, improve pose estimation accuracy, and that *affine covariant features* do not work well in current pose estimation frameworks. The data sets and their ground truth is available on the web to allow future comparison with novel algorithms.**

I. Introduction

Pose estimation, or estimation of the 6 degree-of-freedom *geometrical state* from a single 2D image is an important problem that has received considerable attention over the years [1], [2], [3], [4], [5], [6], [7]. Applications include industrial automation such as bin picking (see figure 1), support systems for augmented reality as well as a whole range of consumer products including toys and house-hold appliances. Important properties of a real-world system for pose estimation is robustness to *occlusion*, *changes in scale*, and *lighting*. Occlusion is usually handled by using local descriptors [8], [9], and robustness to scale is usually solved by some kind of scale-space approach [10]. Robustness to lighting changes seems to be the most challenging problem, as will be made evident in the experiments section, and most local descriptors attempt to deal with this by using normalised features based on derivatives of the intensity.

The local features typically used in view-based pose estimation have previously been evaluated for the purposes of view matching, and object recognition. In such evaluations, computation can be divided into three steps: *detection* of interest points, *descriptor construction*, and descriptor *matching* [9]. A pose estimation system, by necessity, has to contain two additional steps: *pose hypothesis generation*, and *pose clustering* [1], [6]. For this reason, a feature that is completely view invariant will be quite useful in view matching and object recognition, but useless in giving a pose estimate. This justifies the need for specific evaluation of local features in the pose estimation framework.

Fig. 1. Bin picking.

In this paper we will detail a framework and data sets for a performance evaluation of full 6 degree-of-freedom pose estimation. We do also evaluate the performance for fourteen local descriptors in this framework. The test is performed with a strong relation to industrial automation, where several instances of the same object need to be distingusihed, see figure 1. Therefore this particular test is constrained to recovering the pose for one specific object at a time, without attempting to determine the object type. There is however no fundamental limitation to one object, and indeed a similar framework has been used with several objects (albeit without determining pose) in [11]. The framework and two of the local descriptors we test have previously been proven to work in a system for bin picking in [1].

The data-set we have produced for the evaluation is divided into two sub-sets. The first sub-set consists of 16 objects seen from $0 - 180°$ rotation in one degree-of-freedom and $0 - 90°$ in another, both sampled in steps of $5°$. All 16 objects are available with and without a

This work was in part supported by SICKIVP and by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 215078, DIPLECS.

F. Viksten is with Division of Information Coding, Dept. of Electrical Engineering, University of Linköping, 581 83 Linköping, Sweden viksten@isy.liu.se

P.-E. Forssén is with Computer Vision Laboratory, Dept. of Electrical Engineering, University of Linköping, 581 83 Linköping, Sweden perfo@isy.liu.se

B. Johansson and Anders Moe are with SICK-IVP, Wallenbergs gata 4, 583 35 Linköping, Sweden bjorn.johansson/anders.moe@sickivp.se

cluttered background. One of the 16 objects is available in two additional lighting settings, i.e. in total that object has three full rotation sets with black background and three with a cluttered background. All in all, the first sub-set consists of 25 308 color images of mega-pixel size, tagged with rotation angles in their file names. The second sub-set consists of a series of 600 images for one of the objects captured with another camera under varying zoom. The combined data-set allows for controlled evaluation of *off-image-plane object rotation*, under *varying lighting conditions*, *scale change*, both with and without *cluttered background*. We also provide images of a stationary object with a *moving light source*. The data-set could besides being used for evaluation of local descriptors in pose estimation also be used for evaluation of *interest point* (IP) detectors, or complete frameworks for pose estimation. The entire data-set is available for download at [12].

A. Related Research

Evaluation of interest point detectors and local descriptors have previously been done on the wide-baseline stereo task [13], [9], [14], and in the setting of recognition of objects or object class [15], [16]. The object pose estimation problem is however sufficiently different from wide baseline stereo and general object recognition to require a separate feature evaluation. In object recognition and wide baseline stereo, view invariance for features is a good thing. In the object pose estimation application it is on the other hand important that a descriptor can be distinguished within a large database of descriptors, many of which were generated from visually similar image patches. In other words, the features need to be view specific if they are to tell one view from another. For this reason it is not obvious that a pose estimation evaluation will rank local descriptors in the same way as wide-baseline and object recognition tests.

To the best of our knowledge there are no publically available datasets that allows for controlled evaluation of pose estimation with sufficient accuracy for grasping. The annual PASCAL VOC datasets only provide pose information in the form of the view tags 'frontal', and 'side'. The COIL database [17] which has 100 images of 72 neatly centered objects at low resolution, only has one rotational degree of freedom. The closest to a pose estimation dataset is the one recently provided by Savarese and Fei-Fei [18]. This dataset has 8 angles along one rotational degree of freedom, 2-3 elevation angles, 3 scales, and 9-11 instances for 10 different objects. The purpose of this dataset was however not pose estimation for grasping, and consequently their sampling along the rotational degrees of freedom is much too sparse. For instance in [16] it is shown that for non-planar objects, all local features tested required at least a sampling density of about 25°.

Interestingly, the upper bound of about 25° on view change found in [16] fits well with work on modelling human vision. In [19] it is shown that the human visual system works as if it used a view-based lookup function when recognizing objects, and is robust up to about 20° view change. This is also in line with our choice of step size for training in the manipulator angles.

B. Contributions

We here present an evaluation framework for local features in 6 degree-of-freedom view-based pose estimation, and also make the used data-set publicly available on the web [12].

In the setting of pose estimation from a single image we expand upon previous publications in the following ways: 1. we include 14 descriptors from [3], [4], [7], [8], [9], [20], [21], [11], most of them untested in the setting. 2. the tests are made more extensive by using 16 different objects with sufficiently high sampling density and additionally they include a cluttered background. 3. we add a new light change sequence with three fixed light sources and known groundtruth. This new sequence does not have as large variations as the the previously used freely moving light source test [21] to which we again add many more descriptors.

II. Pose Estimation Framework

This presentation uses a match–vote–cluster scheme for performing view-based pose estimation. The approach is common in the literature [6], [21], [11]. The description of the framework is divided into the two modes it is run in; *training* and *query* (or evaluation) mode. As an example the SIFT descriptor will be used as part of the explanation. Adaptations with regard to other descriptors are described for each of them in section III.

A. Pose Representation

The 6 degrees-of-freedom (DOF) pose of an object in the camera coordinate system, $\mathbf{G} = (X\ Y\ Z\ \theta_X\ \theta_Y\ \theta_Z)^T$, consists of the object position along three orthogonal axes, and three rotation angles about these axes. Since this representation is useful for grasping, we will refer to it as *grasping coordinates*.

When estimating an object pose from local image features, it is however convenient to use a different representation, $\mathbf{E} = (x\ y\ \Delta\alpha\ \Delta s\ \phi\ \theta)^T$, which we will refer to as *estimation coordinates*. Two DOFs can be determined from the image plane location of the object (x, y). Another two DOFs are given by the relative image plane rotation ($\Delta\alpha$), and the relative scale change (Δs), both in relation to a reference view. The two remaining DOFs are represented by the two object rotation angles (ϕ, θ), see figure 2. Provided that image rectification has been performed, that the camera calibration parameters are known, and that the pose in the reference view is known, conversion between the two pose representations is straightforward [7].

B. Training

The system is trained using a set of images of an object sampled from different viewing angles. It can be argued that the more physical state attributes the method/system is invariant to, the fewer samples are needed. The features we use are invariant to position, image plane rotation and to

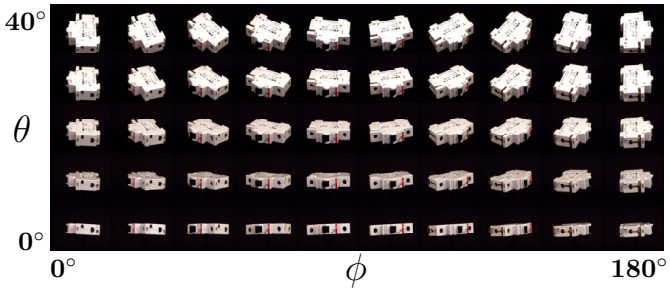

Fig. 2. An object sampled over the two pose angles.

some extent scale. We therefore only need to sample images by varying the two viewing angles ϕ and θ, see figure 2. We will in this presentation refer to these angles as *pose angles*.

During training, the system does the following for each training image:

1) Detect interest points (IP)
2) Extract local descriptors
3) Store each descriptor together with *auxiliary information*.

In the case of the SIFT descriptor the auxiliary information consists of the pose angles (ϕ, θ) the position in the image where the IP was found (x, y), the scale where it was found (s), and the reference direction specified by the SIFT descriptor (α). For conversion to grasping coordinates (see section II-A) we also need to store the image position of one or more reference points on the object, and the distance to the object (or alternatively, the object size).

Collecting and storing data in this manner for later use in e.g. interpolation, is called as *lazy learning* or *memory based learning* [22]. Interestingly, the human vision system also appears to work as if it used database look-up functions when recognizing objects [19].

C. Querying / evaluation

Once the system has been trained, we want to use it to estimate the geometrical state of an object (represented by the estimation coordinates). The whole estimation procedure is illustrated in figure 3, and can be detailed according to:

1) Detect IPs
2) Extract descriptors
3) Find the k most similar features in the database
4) Retrieve the pose angles from the auxiliary information
5) Compute the rest of the pose estimate using the feature location and scale, and the auxiliary information.
6) Cluster in the 6 dimensional *vote space* (where the estimation coordinates live) to find the most likely pose estimate.

Steps one and two are the same as during training. Step three uses the Euclidean distance to compute the k nearest neighboring matches between the query and prototype descriptors. In step four the pose angles are found directly from the auxiliary information. Step five varies between the different descriptors and will be detailed in section III. In the case of the SIFT descriptor, image plane rotation is found as the difference in reference directions between the query and prototype features. Scale change is found as the quotient

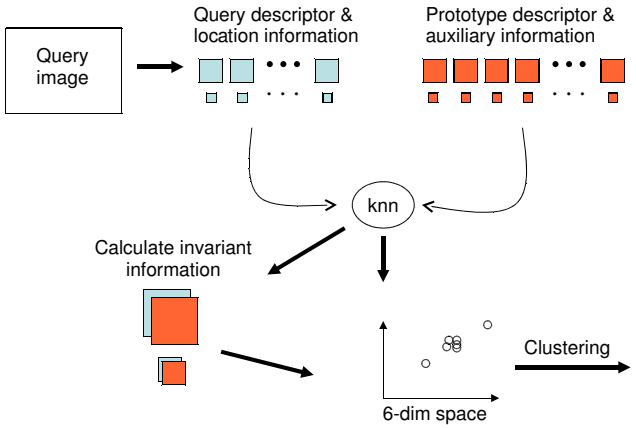

Fig. 3. Overview of the query mode.

between the scales for the query and prototype features. The relative position is then retrieved by de-rotating the location of the detected feature, and then subtracting the location of the prototype feature.

The last step is again the same for all methods in this presentation. The votes for the pose angles ϕ and θ, image-plane rotation α, relative position and scale change are inserted into a 6-dimensional space. To find local density peaks in this space and estimate a mean of such a peak, or cluster, mean-shift clustering [23] is used. Mean shift clustering outputs a cluster density value D_i for each cluster, with $D_i \geq D_{i+1}$ and from these we compute a certainty measure $c \in [0, 1]$, as

$$c = 1 - D_2/D_1 \,. \tag{1}$$

A high c value signifies that the highest peak D_1 in the pose estimate density is well above the second highest D_2, and is thus most likely the correct one. This means that the method can be used to search for several objects of the same kind as they will form different clusters. The computational complexity is linear in the number of detected features. This approach is quite common in the literature, see e.g. [6], [21], [4], [11].

III. Local Descriptors

This section describes the local descriptors that are evaluated in this presentation. The local descriptors can be divided into two classes; *singlets* and *duplets*. The singlets use a single IP to form the descriptor whereas a duplet uses two IPs to form one descriptor. In the case of duplets, the position of both IPs are saved together with the descriptor during training of the system.

A. Patch-duplets (PD & PDCC)

The first patch-duplet variant [21], referred to as PD, uses a sub-pixel Harris detector for IP detection. This method forms pairs between each IP and its three closest IPs.

The second variant [11] is referred to as PDCC for Patch-duplet from Colour Contour frames. It uses a colour modification of the Canny edge-detection and is the only IP detection method in this test that makes use of colour images.

The detected edges are split into contour segments that fit either a line, or an ellipse model. IPs are chosen as the two points furthest from each other. The combinatorics are not as bad as in the PD case, since each IP only forms a duplet with the IP to which it is connected by the line segment or ellipse.

Both patch-duplet variants use a descriptor computed from the double angle representation [24] of the local orientation in box-shaped area around each IP. The connection of two IPs gives both an orientation for the boxes as well as a size for the area which depends on the distance between the IPs.

Both patch-duplet variants extract IPs and descriptors for only two scales of the input image.

Duplets use the distance between its two points to recover scale. Rotation and position of the object uses the center point on the line connecting the two IPs in the duplet.

B. Scene-Tensor Duplets (ST)

The Scene-Tensor (ST) [3] extends on the orientation tensor [24] by using information on location as well as gradient direction. This makes it possible to extract information on line segments within the estimation window of a specific ST. IP detection is done in 2 scales and is refined by information extracted from the tensor. Once the information has been extraced from the ST, IPs are connected much like in the case of the Patch-duplet in section III-A and a very basic duplet with only 4 values for invariant angles is formed. Pose recovery for the ST duplets in query mode is identical to the PD.

C. Log-polar Sampled Patches (LP & LPSI)

Log-polar sampled patches are related to geometric blur [25] but were designed specifically for pose estimation [4]. In [4] it is stated that each detected IP can be seen as a point of fixation for a steerable camera that then uses foveal sampling as a means of focusing processing in the area close to that point. After a patch with edge information in double angle representation [24] is sampled using the log-polar sample pattern it is normalized. Each extracted descriptor gets transformed using the discrete Fourier transform before it is stored, thus transferring the information on scale and rotation to the phase of the patch. The auxiliary information stored with the descriptor is IP position and the scale at which it was found. The use of two IP-detectors for the log-polar patches gives two variants in this presentation.

The first variant [4], referred to as LP in this presentation, uses the same sub-pixel improved Harris IP detector as PD. Like in the PD case, the IPs are detected in two scales.

The second variant, suggested in [7], uses difference-of-Gaussians (DoG) for IP detection. The size of the log-polar sampled region is controlled by the scale at which the IP is found and it is called LPSI (SI for scale invariant). The DoG implementation was taken from [26].

In matching, only the magnitude of the descriptor is used, thus making it invariant to scale and rotation. Scale and rotation are recovered by correlating the matched patches after an inverse Fourier transform has been applied to them. Compared to LP, LPSI also uses the quotient between the scale at which the IPs were detected and multiplies that with the scale factor found by correlation between patches. This means that LPSI does not have to recover as large changes in scale by correlation alone, which should make it more robust to scale changes. Position is calculated by first transforming the prototype descriptor position to the vote space using the scale and rotation from above and then taking the difference to the query position.

D. SIFT

A very good and detailed presentation of the scale invariant feature transform can be found in [27]. For details on how SIFT is used in pose estimation, see section II. We have used the implementation provided by Lowe [28].

E. SURF

The speeded up robust feature, or SURF [20], includes both an IP detector and a descriptor. In this report the SURF implementation from [29] was used, only slightly modified to output scale and orientation instead of the covariance matrix which is the default. We only matched descriptors with the same Hessian sign just as it is described in [20], but used the same matching code as for all the other descriptors. Except for the Hessian sign being used as auxiliary information, pose estimation is done like for SIFT.

F. Affine Covariant Features

We have also used a number of feature detectors and descriptors from a binary provided on the web by K. Mikolajczyk [30]. We have used the features GLOH, Cross Correlation (CC), Differential Invariants (DI), Shape Context (SC), SIFT+HarrisAffine (SIFTH), Spin Images (Spin), and Steerable Filters (SF). These features are described in [9]. All these features are used in the same manner as described for SIFT in section II.

G. Descriptor size

Besides the performance, it is also of interest to compare the number of elements in each descriptor, see table I. A larger descriptor (as for SIFT, GLOH, and SIFTH) means more storage requirements, and thus, at equal performance, a smaller descriptor is usually preferred. Note also the very small descriptor size for ST. Due to differences in implementations (Matlab, C-code, implemented by varying people) it is hard to estimate complexity variations between descriptors. It is however our subjective loose analysis that their complexities are similar.

	LP	LPSI	PD	PDCC	ST	SIFT	SURF
#	110	110	64	64	**4**	**128**	64
	GLOH	CC	DI	SC	SIFTH	Spin	SF
#	**128**	81	12	96	**128**	50	14

TABLE I
NUMBER OF ELEMENTS IN DESCRIPTOR

IV. POSE ESTIMATION EXPERIMENT

Our pose estimation experiments are similar to the ones in [21] but extended upon.

A. Methods of Evaluation and Parameters

At the end of section I we presented the data-sets that were produced for this publication. In the evaluation we decided to use only a sub-set of the available images. The interval we have used is seen in the composite image of figure 2. This speeds up the evaluation, and for e.g. the object shown in figure 2, the omitted views can, due to object symmetry, be obtained by image reflections.

Fig. 4. The 16 objects used in the tests and object #6 with cluttered background.

All objects were learned at 10° intervals for both the pose angles. The evaluation is then done at the sample positions in-between, yielding a worst-case in regards to image distortions from the training poses. This gives that there are 95 training views and 72 evaluation views. The test is performed both without a background, which is similar to having objects on a conveyor belt in a factory, and with a heavily cluttered background, see example in figure 4.

The background was created from a high-resolution image of random objects, including some used in the test. This image was then printed on paper and fastened to the turntable background. All the cluttered background sequences use the same background image.

The captured images are of much higher resolution than the ones used in the evaluation, so all images are down-sampled. What is noted as scale equal to one in all tests is the scale at which the objects were trained. Each evaluation view is subjected to a random scaling between 1.3 and 0.7 as well as a random image plane rotation between -180° and 180°. For all scale values, downsampling from the original high resolution images is used. All interpolations are done by bi-cubic interpolation.

The vector of random scaling and rotation for each view was created once and then reused for all objects and all descriptors. This ensures a fair comparison. We also ran the experiment several times with varying random vectors and they, as far as we have seen, all give roughly the same results.

The error values presented in the experiment results are distances between estimates and ground-truth. The two pose angles are combined into a vector on the unit sphere so only one value is presented. The error value is given by the space angle between the vector for the ground-truth and the vector for the measured pose angles. In the same plot as the object state values, we show the measure of certainty.

We also show plots of the percentage of found descriptors yielding good matches under the four constraints; correct *pose angles*, correct *pose angles and rotation*, correct *pose angles and scale* as well as *full state* estimate (pose angles, scale and image rotation) correct. These plots can show if there is a specific weak point of the method, e.g. if matching descriptors or recovering rotation is a major weak point of the method. Correct pose angle refers to the descriptor matching to any of the 4 pose angles in the database closest to the ground truth. Correct matches are, for evaluation purposes labeled as *good*, and the remaining matches are labeled *bad*. For scale and image plane rotation the matches needed to be within ± 0.1 and $\pm 10^\circ$ respectively of the ground-truth to be considered as good.

The labeling allows us to measure the separation between good and bad matches using Fisher's quotient J given by

$$J = \|\mu(B) - \mu(G)\|^2 / (\sigma^2(B) + \sigma^2(G)) \qquad (2)$$

were B and G are sets of Euclidean distances for bad and good matches. The larger the separation between good and bad matches the more distinctive the descriptors are.

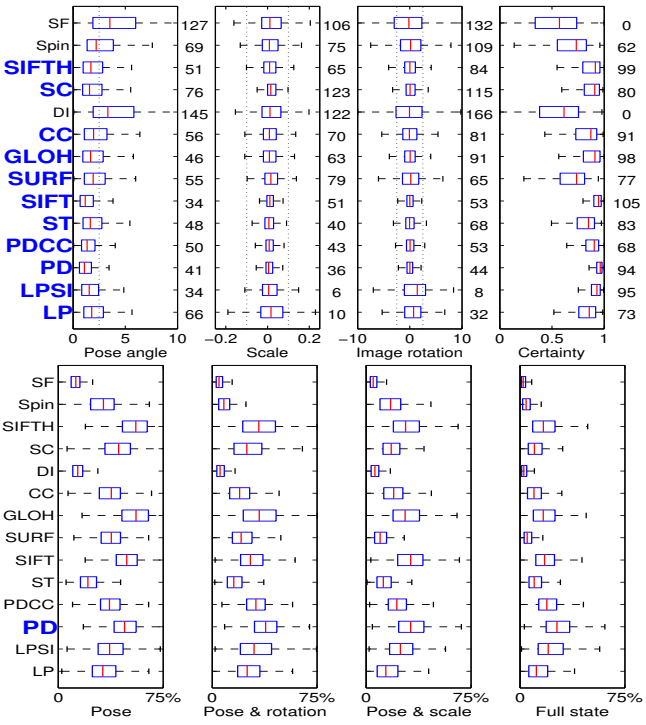

Fig. 5. Top: Error values for pose angles, image plane rotation and scale as well as the certainty. Bottom: Match accuracy plots. Fraction of good matches under the constraints *pose*, *pose and rotation*, *pose and scale*, and *full state* (pose angles, scale and image rotation).

B. Results using Black Background

In figure 5 we present pose-estimation results for objects on a black background. Each box-plot shows the first ($x_{.25}$) and third ($x_{.75}$) quartile by the solid box surrounding the line for the median ($x_{.50}$). The whiskers are located at $1.5 \times \mathrm{IQR}$ above or below the first and third quartile, where $\mathrm{IQR} = x_{.75} - x_{.25}$. Any values outside of the whiskers are considered as outliers. Outliers are not plotted, instead the number of outliers are printed on the right of each box. These results are

based on 16 objects with 72 evaluation views each. As an aid for comparison we have added dotted lines to the figures. The position of the dotted lines is chosen as a rough maximum of inaccuracies that a bin-picking application would tolerate ($2.5°$ for pose angle as well as image plane rotation and ± 0.1 for scale). Descriptors with their medians within the dotted lines will be marked in bold text to indicate an OK result. For the plots on accuracy, only the best performer is marked in bold text. From figure 5 we can see that most features perform OK. Exceptions are SF, Spin and DI. The best performer is PD closely followed by SIFT.

The bottom row of plots in figure 5 shows the percentage of the found query descriptors that give good matches under varying constraints on the meaning of good. This can be seen as an approximation to matching the correct descriptor. We can see that PD has the highest percentage of such good matches so in terms of accuracy it performs best. The singlet features SIFT, GLOH and SIFTH do well under the constraint of *pose angle*, but fall behid when rotation and scale are added. For interest-point based duplets, such as PD, scale and rotation estimates become increasingly more accurate the larger the point distance, as the primary noise source is in feature location. For PDCC, which uses contours instead of points, the advantage is less evident.

Table II shows the number of IPs detected for each local descriptor as well as the value for the Fisher quotient to show separation in matching distance for good versus bad matches. The value J in the table is for the *full state* case. Min is important for failure cases and max exposes uneven

#IP	LP	LPSI	PD	PDCC	ST	SIFT	SURF
min	7	9	16	12	**19**	6	7
max	110	260	336	212	345	260	138
μ	32	63	96	61	117	54	42
σ/μ	0.50	0.54	0.50	0.56	**0.44**	0.65	0.52
J	0.15	0.21	**0.62**	0.3	0.082	0.46	0.28

#IP	GLOH	CC	DI	SC	SIFTH	Spin	SF
min	1	1	1	1	1	1	1
max	255	266	266	**1348**	255	266	266
μ	48	54	54	**308**	48	54	54
σ/μ	0.63	0.61	0.61	0.64	0.63	0.61	0.61
J	0.19	0.12	0	0.03	0.2	0.08	0.04

TABLE II

BLACK BACKGROUNDS: IP & MATCHING STATISTICS.

computation requirements (e.g. true for SC). The mean (μ) shows the expected number of features and σ/μ indicates the reliability. As the images are similar in nature we would prefer a similar number of features to be detected in all images, and thus a low σ/μ value.

C. Results using Cluttered Background

In figure 6 we present the pose estimation results for a cluttered background. The plots are presented in the same way as for black background. In this experiment SF, Spin, SIFTH, DI, CC, GLOH, SURF and ST have such poor performance that they would be completely useless in the application. For this reason figure 6 is scaled to focus on the other methods and we can see that LPSI, PD and SIFT are

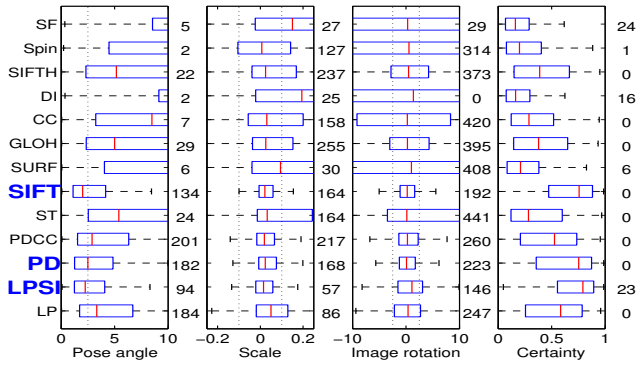

Fig. 6. Pose errors with cluttered background.

within the required bounds. SC is not to be found in this and the following tests since the binary had segmentation faults for most images.

#IP	LP	LPSI	PD	PDCC	ST	SIFT	SURF
min	59	73	189	130	**195**	107	48
max	661	1498	2013	**2492**	1974	1309	1556
μ	206	406	640	640	**683**	426	354
σ/μ	0.41	0.54	0.41	0.55	**0.37**	0.50	0.66
J	0.4	0.45	1.1	0.26	0.026	**2.2**	0.72

#IP	GLOH	CC	DI	SC	SIFTH	Spin	SF
min	93	108	108	fail	93	108	108
max	1503	1628	1628	fail	1503	1628	1628
μ	408	452	452	fail	408	452	452
σ/μ	0.48	0.47	0.47	N/A	0.48	0.47	0.47
J	1.4	0.73	0	N/A	1.6	0.56	0.11

TABLE III

CLUTTERED BACKGROUNDS: IP & MATCHING STATISTICS.

Table III again shows statistics for the detected IPs and for matching. Looking at table III we can see that the discriminating power of SIFT is really good. Closest is SIFTH, GLOH and PD.

V. SCALE CHANGE EXPERIMENT

To evaluate the robustness to scale change, we use a sequence of images of object #14 captured with a different camera and lens over the zoom range $[0.37 \;\; 2.85]$. The ground-truth has been produced by hand by analysing the sequence frame by frame. The first and last images of the scale change test are shown in the upper row of figure 7.

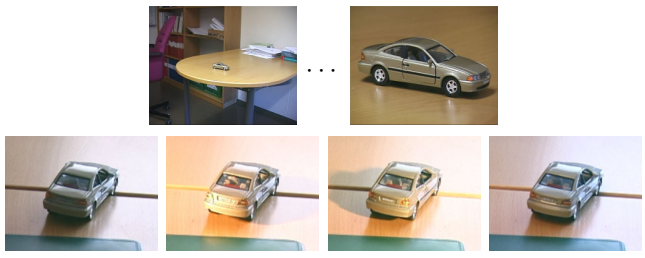

Fig. 7. Top: First and last image in zoom test. Bottom: Random images from moving light source test.

A. Scale Change Results

In figure 8 we see that the duplet PD does OK in estimating the pose over the range of zoom, again likely

due to it being a duplet feature. PDCC and SIFT actually do well for a large part of the zoom range.

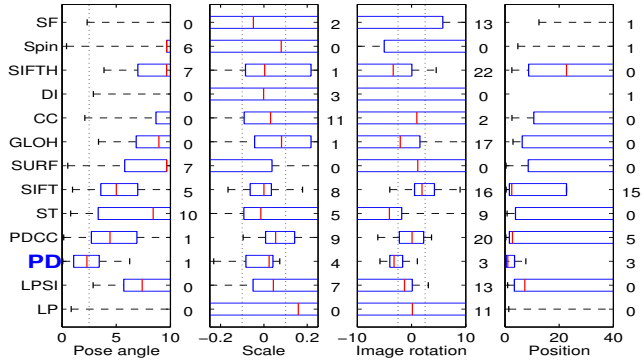

Fig. 8. Pose errors for zoom sequence. Certainty has been replaced with an image position error measure.

VI. Moving Light Source Experiment

To test robustness to light changes, an evaluation sequence with images of object #14 under varying lighting conditions was captured (with the same camera as in the zoom test). During capture, a light source was moved around the object by hand. Ground-truth was produced by hand for this test. Example images with cut-outs from images in the light change sequence can be seen in the bottom row of figure 7. In [21], a resampling to a scale of 0.7 was used, here it is 1.

A. Moving Light Source Results

The results from this test are shown in figure 9. This test is quite hard, as indicated by all descriptors showing quite poor results, except PD which fails only slightly.

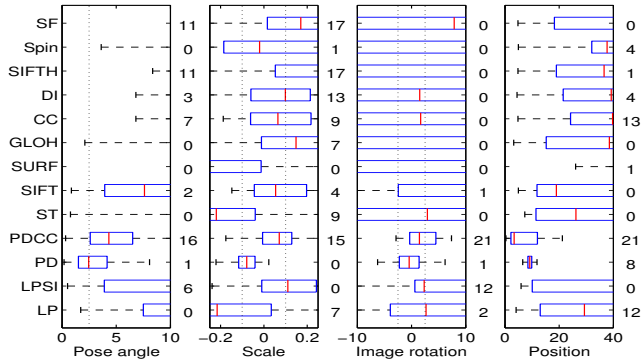

Fig. 9. Pose errors for moving light source sequence. Certainty has been replaced with an image position error measure.

VII. Light Change Pose Estimation Experiments

A second test for robustness to light changes uses object #5 from the first pose estimation test, but with two additional light settings, both less extreme than in the moving light test. Example images from each light setting can be seen in figure 10. In this test we trained for each light setting and then evaluated on the two other light settings. The evaluation is done with both black and with cluttered background.

Fig. 10. Left to right: Ambient, left, and right illumination of object #5.

A. Light Change Results

The results for the light change experiments on black background can be seen in figure 11. Only PDCC and SIFT are useful here. ST, PD and LPSI just barely fail in this test. From table IV we can see that SIFT and SURF are quite descriminative which seems reasonable for SIFT as it also has low error values in figure 11. SURF on the other hand, performs quite poorly in spite of a high value for J. This is evidence of a lot of matches just outside the limits for *good* matches and that the matches outside of the limits are of quite poor quality.

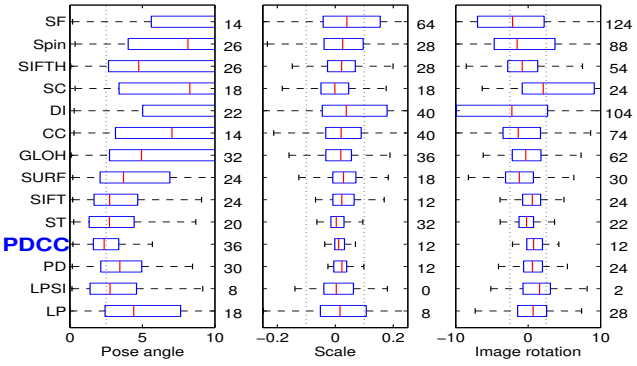

Fig. 11. Error values for object states from light change experiment on black background.

#IP	LP	LPSI	PD	PDCC	ST	SIFT	SURF
min	32	66	96	58	**102**	56	31
max	113	275	339	188	348	278	138
μ	64	154	192	119	207	140	75
σ/μ	0.30	0.34	0.29	0.28	**0.26**	0.36	0.35
J	0.23	0.21	0.66	0.26	0.18	**0.91**	0.87

#IP	GLOH	CC	DI	SC	SIFTH	Spin	SF
min	29	33	33	89	29	33	33
max	255	266	266	**1367**	255	266	266
μ	96	107	107	**531**	96	107	107
σ/μ	0.44	0.44	0.44	0.46	0.44	0.44	0.44
J	0.64	0.31	0	0.19	0.6	0.25	0.14

TABLE IV

Light change on black background: IP & matching statistics.

Results with cluttered background can be found in figure 12 and table V. All the descriptors fail in this test but PDCC remains the best performer. ST handled the changing light on black background quite well but a cluttered background once again proves too challenging for this low dimensional descriptor. We see that the low dimensional descriptors get a very poor separation between good and bad matches. The same discussion about SIFT and SURF is true for cluttered background as for black background.

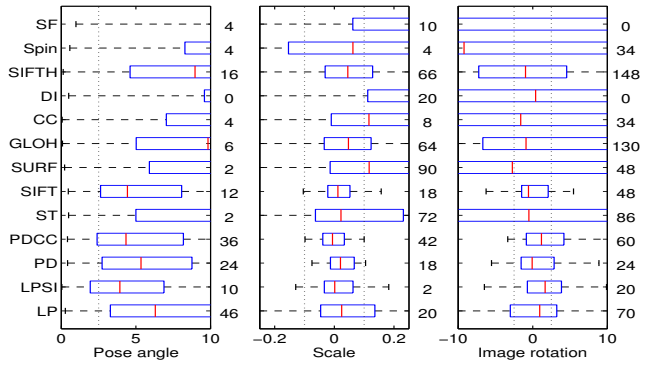

Fig. 12. Error values for object states from light change experiment on cluttered background.

#IP	LP	LPSI	PD	PDCC	ST	SIFT	SURF
min	101	166	321	224	**342**	196	86
max	305	663	945	854	**957**	730	415
μ	194	400	605	490	**613**	432	241
σ/μ	0.25	0.35	**0.24**	0.32	0.25	0.32	0.35
J	0.23	0.14	0.56	0	0.03	**1.2**	**1.2**
#IP	GLOH	CC	DI	SC	SIFTH	Spin	SF
min	143	148	148	fail	143	148	148
max	611	684	684	fail	611	684	684
μ	343	384	384	fail	343	384	384
σ/μ	0.30	0.29	0.29	N/A	0.30	0.29	0.2943
J	0.58	0.24	0	N/A	0.67	0.12	0.05

TABLE V

LIGHT CHANGE ON CLUTTERED SCENE: IP & MATCHING STATISTICS.

VIII. CONCLUDING REMARKS

From our experiments we can note that patch duplet methods do well (PD and PDCC), whereas the singlet methods do not (the exceptions are SIFT, that comes second in total, and to some extent LPSI). The reason for this appears to be that using two feature points allow more accurate estimates of image plane rotation and scale (see text around figure 5).

The results for the moderate light changes (see figure 10) in the second light change test showed that the PD variants, SIFT and LPSI had the best performance. The large light changes in the moving light source test (see figure 7) were too difficult for most of the descriptors. In this test we found that e.g. SIFT and PD extracted roughly the same amount of descriptors (again roughly as many as for the same object in the pose estimation test on black background), but SIFT had a much lower percentage of its found descriptors voting for the winning cluster than PD. This indicates that the descriptor matching was harder for SIFT. A possible explanation for this might be a less accurate scale estimate in the SIFT detector (and thus more changes in the descriptors).

We can also conclude that affine covariant features [9] did not do particularly well in our tests. It is possible that a modified framework could change this. Such a modification should use the affine deformation estimate as an additional measurement. This is however not straightforward, and has to be left for future work.

Finally we would like to stress that, in practise one should not use just one feature for pose estimation. Instead, features that do reasonably well in these tests should be used in combination, using e.g. the multicue integration described in [1].

IX. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution and input of Prof. Robert Forchheimer.

REFERENCES

[1] F. Vikstén, R. Söderberg, K. Nordberg, and C. Perwass, "Increasing Pose Estimation Performance using Multi-cue Integration," in *ICRA*, Orlando, Florida, USA, May 2006.
[2] R. Söderberg, "Compact representations and multi-cue integration for robotics," Linköping University, Sweden, Thesis, April 2005.
[3] R. Söderberg, K. Nordberg, and G. Granlund, "An invariant and compact representation for unrestricted pose estimation," in *Iberian Conf. on Patt. Rec. and Im. Analysis*, June 2005.
[4] F. Vikstén and A. Moe, "Local single-patch features for pose estimation using the log-polar transform," in *Iberian Conf. on Patt. Rec. and Im. Analysis*. Estoril, Portugal: IAPR, June 2005.
[5] B. Johansson and A. Moe, "ISY-R-2553: Patch-duplets for object recognition and pose estimation," Linköping, Sweden, 2003.
[6] B. Leibe and B. Schiele, "Interleaved object categorization and segmentation," in *British Machine Vision Conference*, 2003.
[7] F. Vikstén, "Methods for vision-based robotic automation," Dept. EE, Linköping University," Lic. Thesis, April 2005, thesis No. 1161.
[8] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999.
[9] K. Mikolajczyk *et al.*, "A comparison of affine region detectors," *IJCV*, vol. 65, no. 1-2, pp. 43–72, 2005.
[10] T. Lindeberg, *Scale-space Theory in Computer Vision*. Kluwer Academic Publishers, 1994, iSBN 0792394186.
[11] P.-E. Forssén and A. Moe, "Autonomous learning of object appearances using colour contour frames," in *CRV*, Canada, June 2006.
[12] F. Viksten, "CVL/ICG dataset for object pose estimation evaluation," 2007, http://www.cvl.isy.liu.se/research/objrec/posedb/.
[13] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *IJCV*, vol. 60, no. 1, pp. 63–86, 2004.
[14] ——, "A performance evaluation of local descriptors," *PAMI*, vol. 27, no. 10, pp. 1615–1630, 2005.
[15] K. Mikolajczyk, B. Leibe, and B. Schiele, "Local features for object class recognition," in *ICCV*, 2005, pp. 1792–1799.
[16] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3D objects," *IJCV*, vol. 73, no. 3, July 2007.
[17] S. A. Nene, S. K. Nayar, and H. Murase, "COIL-100," 1996, http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php.
[18] S. Savarese and L. Fei-Fei, "3d generic object categorization, localization and pose estimation," in *ICCV*, October 2007.
[19] S. Edelman and H. Bülthoff, "Modeling human visual object recognition," in *Proc. IJCNN*, vol. 4, September 1992, pp. 37–42.
[20] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded-up robust features," in *ECCV*, 2006.
[21] B. Johansson and A. Moe, "Patch-duplets for object recognition and pose estimation," in *CRV*, Victoria, BC, Canada, May 2005, pp. 9–16.
[22] C. Atkeson, "Using locally weighted regression for robot learning," in *ICRA*, Sacramento, CA, 1991, pp. 958–963.
[23] Y. Cheng, "Mean shift, mode seeking, and clustering," *PAMI*, vol. 17, no. 8, pp. 790–799, August 1995.
[24] G. Granlund and H. Knutsson, *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995.
[25] A. Berg and J. Malik, "Geometric blur for template matching," *CVPR*, vol. 1, pp. I–607–I–614, 2001.
[26] A. Vedaldi, "SIFT - an open implementation of SIFT," 2006, http://vision.ucla.edu/∼vedaldi/code/sift/sift.html.
[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
[28] D. Lowe, "Demo software: SIFT keypoint detector," 2005, http://www.cs.ubc.ca/∼lowe/keypoints/.
[29] E. T. H. Zürich, "SURF," 2006, http://www.vision.ee.ethz.ch/∼surf/.
[30] K. Mikolajczyk, "Binaries for affine covariant features," 2005, http://www.robots.ox.ac.uk/∼vgg/research/affine/.