Proceedings VISAPP International Workshop on Robotic Perception 2008, pp. 23–32.

Pose Clustering From Stereo Data

Ulrich Hillenbrand

Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
82234 Wessling, Germany
Ulrich.Hillenbrand@dlr.de

Abstract. This article describes an algorithm for pose or motion estimation based on clustering of parameters in the six-dimensional pose space. The parameter samples are computed from data samples randomly drawn from stereo data points. The estimator is global and robust, performing matches to parts of a scene without prior pose information. It is general, in that it does not require any particular object features. Empirical object models can be built largely automatically. An implemented application from the service robotic domain and a quantitative performance study on real data are presented.

1 Introduction

Estimation of the pose of known objects in unknown scenes is a basic problem within many robotic applications. It arises not only in the context of object manipulation but also of self localization and navigation in a known environment. Moreover, the problem of pose estimation is mathematically and algorithmically equivalent to the one of motion estimation of objects or of the sensor itself relative to the environment. The only difference between the two problems is that for pose estimation, one of the two data sets that have to be registered is a priori known as the model of the object or the environment.

Data acquired from a natural scene do not exclusively contain a single known object, and two data sets in a motion sequence do not completely overlap. The estimator hence needs to be robust in the statistical sense, that is, it must select in the estimation process that part of the data that does match between two sets. Moreover, usually the detailed correspondences, on the level of features or data points, even between the overlapping parts of two data sets, are not reliably known and must be established during estimation. The combinatorics of the correspondence problem can give rise to a very high proportion (approaching 100%) of effective outliers.

The estimation problem considered in this paper takes the correspondence and outlier problems to the extreme, in that i) no prior knowledge of object pose is assumed, i.e., we aim at global pose estimation, and ii) no distinctive or high-level object features are used, such that one is faced with a very large number of possible correspondences between simple data points. Such simple data points can be, e.g., image points with a gradient value above a threshold, edge and corner points [1], or range-data points from stereo or laser imaging. The advantage of using simple data rather than distinctive features is the general applicability to all kinds of objects and under a broad range of imaging conditions. Distinctive features, on the other hand, require either the objects of interest to possess certain characteristics (sharp edges, corners, colors, etc.) or that the imaging conditions (viewpoint, lighting) do not significantly change (appearance-based features, SIFT [2] and its variants).

There are various ways in which the present estimation algorithm relates to methods used previously in pattern recognition. In fact, it may be understood as

– a continuous version of a randomized, generalized Hough transform,
– a density estimator in parameter space,
– a clustering procedure for parameter hypotheses.

Like parameter clustering in general, the method is based upon robust statistics in parameter space [3–8], as opposed to methods that rely on statistics in data space [9]. The technique belongs to the class of non-parametric estimators, as no parametric models of the underlying probability densities are assumed.

In the following section, the algorithm is described in some detail. The section on experiments gives an example application to object manipulation that regularly runs in our lab and a quantitative analysis of its performance on stereo data.

2 The algorithm

The algorithm discussed in this paper is based upon range data. Other algorithmic variants of the same principle of pose clustering may process other types of data. The range data is here obtained from a stereo algorithm that computes local correspondences in an image pair from five partly overlapping correlation windows [10]. The outcome of stereo processing is a point cloud with data points largely restricted to surface creases, sharp bends, and depth discontinuities. Typical artifacts from correlation-based stereo processing are also present, such as blurring of depth discontinuities and unstable depth values for edges nearly parallel to the epipolar lines; see figs. 2 and 3.

The algorithm for object detection and pose estimation may be described as a sequence of three distinct steps:

1. model generation,
2. parameter sampling,
3. parameter clustering.

Model generation is an empirical process that runs offline and may hence be regarded as a training step for the algorithm. Sampling and clustering of parameters are the actual processing steps for scene data that need to run in real time. This section describes each of these three steps in turn.

2.1 Model generation

In a training phase, data of the sought objects are collected by the same sensing process that is used later for recognition. In the present case, we collect range data points produced by the stereo algorithm. Depending on the object's complexity, between two and, say, ten different views of the object are acquired. Different views are registered in an object coordinate system by an external pose measurement device, e.g., a robot. Alternatively, given sufficient overlap between the data sets, a registration of the different views may be achieved by the very same algorithm used later for pose estimation.

Data from different views are fused by discarding all points that are not stable under view variation. More precisely, an intersection is computed of each data set in turn with all the remaining data, allowing for a tolerance of a few millimeters for two data points to be considered the same. These intersection sets are collected into the final data set. Formally, given data sets $D_1, D_2, \ldots, D_n$ acquired from different object views, the model point set M is constructed as

$$M = \cup_{i=1}^{n} \big[D_i \cap \big(\cup_{\substack{j=1 \\ j \neq i}}^{n} D_j \big) \big] = \cup_{\substack{i,j=1 \\ i \neq j}}^{n} \big(D_i \cap D_j \big) \ , \tag{1}$$

where the intersection tolerates small point differences. This procedure effectively removes view-dependent artifacts of the sensing process, creating an idealized data point set of the object. The point set M along with the lines of sight for each point constitute the object model to be matched against scene data. The object model can be built largely automatically with a robot that moves the sensor on the viewing sphere around the object of interest.

2.2 Parameter sampling

In order to produce a number of pose hypotheses, data samples are drawn from a scene point set S and a model point set M from which pose parameter samples are computed. A pose hypothesis can be computed from a minimum subset of three scene points matched against a subset of three model points. The sampling proceeds thus as follows.

1. Randomly draw a point triple from S.
2. Randomly draw a point triple from M among all triples that are consistent with the triple drawn from S.
3. Compute the rigid motion between the two triples in a least-squares sense.
4. Compute the six parameters of the hypothetical motion.

The parameter samples thus obtained are collected into a spatial array or a tree of buckets, from where they can be efficiently retrieved for the subsequent clustering step. The sampling process stops as soon as a significant number of parameter samples has accumulated anywhere in parameter space. This condition is pragmatically taken as fulfilled when one of the buckets is full. In numbers, from stereo data sets with 10^4 to 10^5 points on the order of 10^6 point triples are drawn.

Corresponding data points from M and S can be found among geometrically consistent groups of points. For drawing consistent point triples in sampling step 2, one has to exploit the constraints that arise from rigid motion. These are i) approximate congruence of the triangles defined by the point triples and ii) view point consistency. The latter means that the plane defined by three simultaneously visible points on a non-transparent solid shape generally exposes the same side to the sensor. Exceptions may occur, e.g., for triples that span holes through a shape. Although the view point constraint does not hold for all points on all shapes under arbitrary motion, it is a useful criterion for guiding the sampling process.

The constraints are enforced by building a hash table of point triples from the model M, which may in fact be done offline as part of the model generation process. The table is accessed through a key that encodes a triple's geometry in relation to the sensor. Given three range data points $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3 \in \mathbb{R}^3$ and their lines of sight $\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3 \in \mathbb{R}^3$, a suitable key $(k_1, k_2, k_3) \in \mathbb{R}^3$ is

$$\begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix} = \begin{cases} \begin{pmatrix} ||\boldsymbol{r}_2 - \boldsymbol{r}_3|| \\ ||\boldsymbol{r}_3 - \boldsymbol{r}_1|| \\ ||\boldsymbol{r}_1 - \boldsymbol{r}_2|| \end{pmatrix} & \text{if } \left[(\boldsymbol{r}_2 - \boldsymbol{r}_1) \times (\boldsymbol{r}_3 - \boldsymbol{r}_1)\right]^T (\boldsymbol{l}_1 + \boldsymbol{l}_2 + \boldsymbol{l}_3) > 0 \ , \\ \begin{pmatrix} ||\boldsymbol{r}_2 - \boldsymbol{r}_3|| \\ ||\boldsymbol{r}_1 - \boldsymbol{r}_2|| \\ ||\boldsymbol{r}_3 - \boldsymbol{r}_1|| \end{pmatrix} & \text{else,} \end{cases} \tag{2}$$

where $|| \cdot ||$ denotes the Euclidean norm. When building the hash table, each model point triple is entered for the key (k_1, k_2, k_3) and its cyclic permutations $(k_2, k_3, k_1), (k_3, k_1, k_2)$; when sampling from the scene data, just one of the permutations is used for indexing into the hash table. Through the hashing procedure, consistent scene-model pairs of point triples can be efficiently drawn.

In step 3 of the sampling procedure, the least-squares rotation $\boldsymbol{R}^* \in SO(3)$ and translation $\boldsymbol{t}^* \in \mathbb{R}^3$ between two point triples $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3 \in M$ and $\boldsymbol{r}'_1, \boldsymbol{r}'_2, \boldsymbol{r}'_3 \in S$ are computed, i.e.,

$$(\boldsymbol{R}^*, \boldsymbol{t}^*) = \arg \min_{(\boldsymbol{R}, \boldsymbol{t}) \in SE(3)} \sum_{i=1}^{3} ||\boldsymbol{R}\,\boldsymbol{r}_i + \boldsymbol{t} - \boldsymbol{r}'_i||^2 \ . \tag{3}$$

The method in [11] provides a solution, based on quaternions, that is specifically tailored to the three-point case and hence more efficient than the general ones, like the popular method based on singular value decomposition. If the three point pairs $(\boldsymbol{r}_i, \boldsymbol{r}'_i)$ are approximately corresponding between M and S, the motion hypothesis $(\boldsymbol{R}^*, \boldsymbol{t}^*)$ will be close to the true object pose.

The choice of parameterization of motions in sampling step 4 is relevant for the clustering of motion hypotheses. Indeed, the result of clustering depends upon the parameter space used. A proper choice is one that respects the topology of the Euclidean group $SE(3)$ [8]. Let $\boldsymbol{\alpha} \in \mathbb{R}^3$ be the exponential/canonical parameters of the rotation $\boldsymbol{R}^*$, that is, $||\boldsymbol{\alpha}||$ is the angle and $\boldsymbol{\alpha}/||\boldsymbol{\alpha}||$ the oriented axis of $\boldsymbol{R}^*$. The parameters $\boldsymbol{\rho} \in \mathbb{R}^3$ of rotations used here are related to the canonical parameters through

$$\boldsymbol{\rho} = \left(\frac{||\boldsymbol{\alpha}|| - \sin ||\boldsymbol{\alpha}||}{\pi} \right)^{1/3} \frac{\boldsymbol{\alpha}}{||\boldsymbol{\alpha}||} \ , \tag{4}$$

that is, a non-linear re-mapping of the rotation angle. Its desirable properties derive from the fact that the invariant Haar measure of the rotation group $SO(3)$ is uniform in the parameters $\boldsymbol{\rho}$, such that there is no bias of pose clustering incurred from the group topology. This kind of parameterization has therefore been called consistent [8]. Translations are consistently parameterized simply by their three vector components $\boldsymbol{\tau} = \boldsymbol{t}^*$.

2.3 Parameter clustering

Significant populations of scene points in S matching a rigid motion of the model points M will produce many parameter samples $\boldsymbol{p} = (\boldsymbol{\rho}, \boldsymbol{\tau}) \in \mathbb{R}^6$ that coincide approximately. The goal of parameter clustering is hence to estimate the location in parameter space of the maximum probability density underlying the obtained parameter samples $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_N\}$ [8]. A practical realization, derived from kernel density estimation, is through the mean-shift procedure [12]. More precisely, a sequence of pose parameters $\boldsymbol{p}^1, \boldsymbol{p}^2, \ldots$ is obtained through iterative weighted averaging

$$\boldsymbol{p}^k = \frac{\sum_{i=1}^N w_i^k \, \boldsymbol{p}_i}{\sum_{i=1}^N w_i^k} \ , \tag{5}$$

$$w_i^k = u\big(||\boldsymbol{\rho}^{k-1} - \boldsymbol{\rho}_i||/r_{\mathrm{rot}}\big) \, u\big(||\boldsymbol{\tau}^{k-1} - \boldsymbol{\tau}_i||/r_{\mathrm{trans}}\big) \ . \tag{6}$$

Here u is a unit step function,

$$u(x) = \begin{cases} 1 \text{ if } x < 1, \\ 0 \text{ else,} \end{cases} \tag{7}$$

such that the averaging procedure (5) operates just on a neighborhood of $\boldsymbol{p}^{k-1}$. The required parameter samples can be efficiently retrieved from buckets indexed by $\boldsymbol{p}^{k-1}$; cf. sec. 2.2. The radii r_{rot} and r_{trans} of the rotational and translational extensions, respectively, of the averaging procedure can be adapted to the local parameter density: a higher density affords smaller radii.

The sequence $\boldsymbol{p}^k$ converges to an estimate of the position of a local density maximum [12], even though the density of parameters is not explicitely estimated. By starting with $\boldsymbol{p}^0$ close to the dominant mode of the density, the sought pose estimate $\hat{\boldsymbol{p}} = \lim_{k\to\infty} \boldsymbol{p}^k$ is thus obtained. The region of the dominant mode, in turn, is estimated from a histogram of pose parameters. Further modes may be explored in an analogous fashion in order to identify additional object instances.

3 Experiments

3.1 An application scenario

The algorithm has been integrated into a humanoid robot system [13] which is used for studying bi-manual manipulation. In particular, scenes composed of carafes, bottles, jars, and glasses are visually analyzed to autonomously perform a sequence of actions such as preparing a drink; see fig. 1. This kind of demonstration runs regularly in our lab.

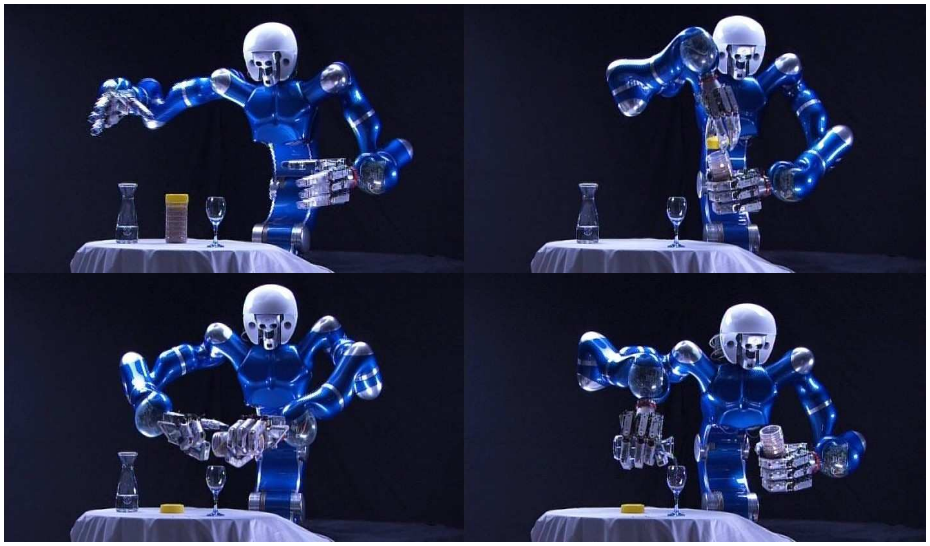

Fig. 1. DLR's humanoid robot 'Justin' (upper body only) preparing a drink: after recognition of the scene unscrewing the lid from a jar of instant tea, dropping some grains of tea into a glass, and adding water from a carafe.

Figure 2 shows a typical scene as viewed through one of the head-mounted stereo cameras and the stereo data with the recognized objects. Pose estimation is performed on the complete data set, that is, without prior segmentation into the three object components and the table component. The transparent objects are an example where reliable extraction of more distinctive features than the raw stereo data can be a severe problem.

3.2 A quantitative study

A quantitative analysis of pose estimation accuracy was carried out for a cardboard grid box (approximate dimensions: 170 × 170 × 50 mm) that can contain metal pieces; see fig. 3. For model building, an empty box was used. As test

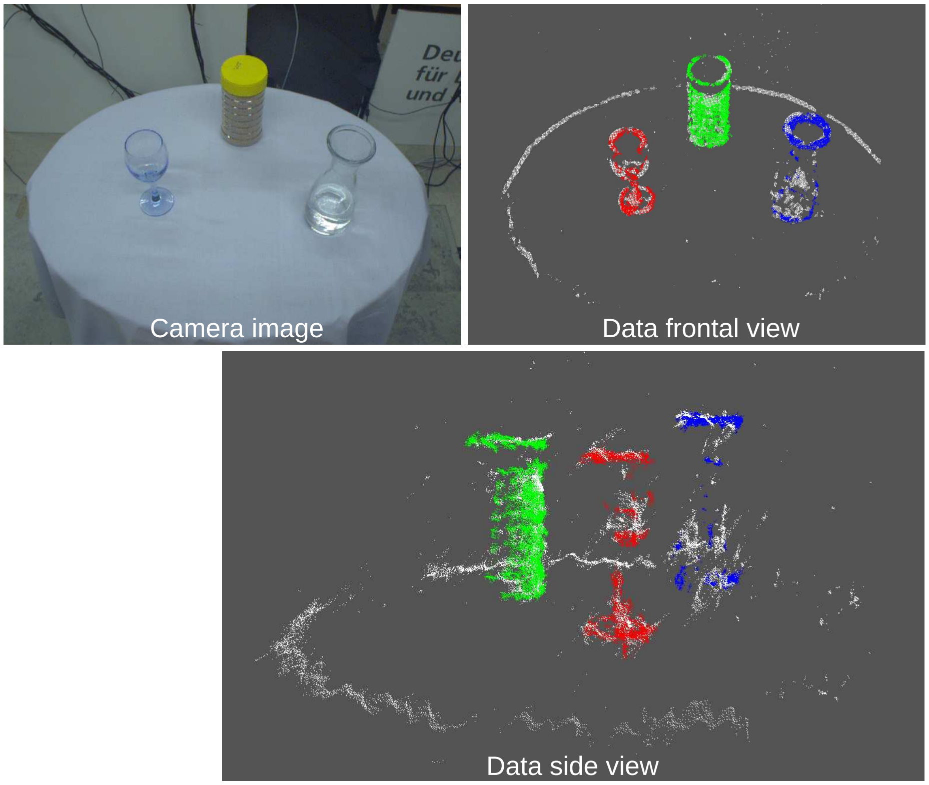

Fig. 2. Scene with objects used by the humanoid 'Justin' (cf. fig. 1) for preparing an instant-tea drink: camera image and stereo data points (white) superposed with model points of the recognized objects (colored).

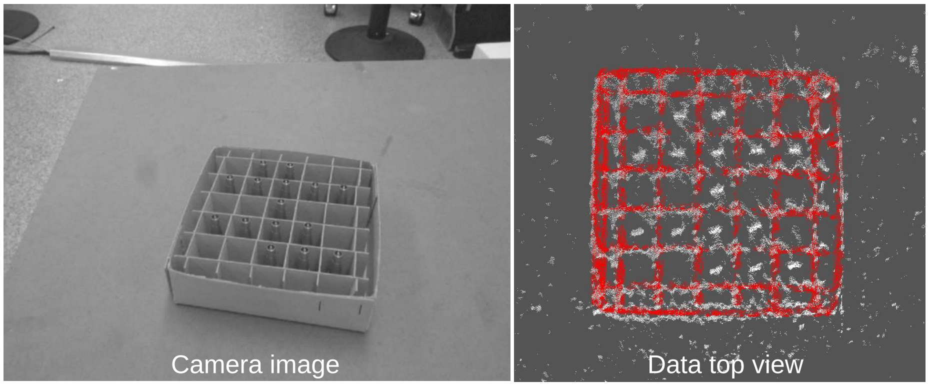

Fig. 3. Scene with a cardboard grid box used for quantitative evaluation of pose estimation accuracy: camera image and stereo data points (white) superposed with model points of the box (red).

scenes, stereo data of the partially filled box from 10 different views were acquired, while a robot moved the stereo cameras around. Using the known motion of the robot between views, all the data sets were registered in a common coordinate system. As a result, the pose that had to be estimated was numerically the same for all sets.

However, a ground truth for the box pose was not available, both for practical and for principle reasons. Practically, it is hard to rely on any other pose measurement to be significantly more accurate than the one we want to test such that it could define the reference pose. Moreover, in principle it is hard to define a true pose for a deformable object such as the cardboard box, as in any setting it will slightly differ in shape from the empty box seen at modeling time.

Avoiding these problems, two kinds of error statistics are here presented. One uses the median of all pose estimates as the reference pose $\boldsymbol{p}_{\mathrm{ref}}$ to estimate the expected error $E(||\hat{\boldsymbol{p}} - \boldsymbol{p}_{\mathrm{ref}}||)$. The other is based on the covariance matrix

$$\boldsymbol{C} = E\Big\{ [\hat{\boldsymbol{p}} - E(\hat{\boldsymbol{p}})]\, [\hat{\boldsymbol{p}} - E(\hat{\boldsymbol{p}})]^T \Big\} \tag{8}$$

and estimates the square root of the total variation $\mathrm{Tr}\,\boldsymbol{C}$, which is a lower bound on the expected quadratic error

$$\begin{aligned} &E(||\hat{\boldsymbol{p}} - \boldsymbol{p}_{\mathrm{true}}||^2) = E[||\hat{\boldsymbol{p}} - E(\hat{\boldsymbol{p}})||^2] + ||E(\hat{\boldsymbol{p}}) - \boldsymbol{p}_{\mathrm{true}}||^2 \geq \\ &E[||\hat{\boldsymbol{p}} - E(\hat{\boldsymbol{p}})||^2] = E\{\mathrm{Tr}\,[\hat{\boldsymbol{p}} - E(\hat{\boldsymbol{p}})]\, [\hat{\boldsymbol{p}} - E(\hat{\boldsymbol{p}})]^T\} = \mathrm{Tr}\,\boldsymbol{C}\ . \end{aligned} \tag{9}$$

Here $\boldsymbol{p}_{\mathrm{true}}$ is the true pose parameter and the neglected term $||E(\hat{\boldsymbol{p}}) - \boldsymbol{p}_{\mathrm{true}}||^2$ is the squared bias of the estimator. Both error statistics are computed for the rotational and translational parameters separately, because of their different physical dimensions.

Pose estimates were computed either in a single stage or in two stages: the first stage estimated the box rotation and translation in their joint 6D parameter space; an optional second stage attempted to refine the translation estimate in its 3D parameter space while keeping the estimated rotation fixed. Different resolutions of parameter space analysis were investigated: the sizes of parameter buckets, histogram bins, and mean-shift radii took identical rotational values $\{0.02, 0.03, 0.04, 0.05\}$[1] with their translational value fixed at 30 mm for the first stage of estimation, and with identical translational values $\{5, 10, 15\}$ mm for the second stage. Estimator variants with only the first stage and with both stages were run on the test data, making a total of 16 tested variants. Run times of the estimators were recorded for a C++ implementation on a single CPU at 3.0 GHz. The measured times included building of the model hash table.

For each of the 10 box views, 100 data sets were acquired, yielding a total of 1000 pose estimates. Plots of the two kinds of error statistics, each for rotation and translation estimates, versus expected run time for the 16 estimator variants are presented in fig. 4. There are five main observations to be noted.

[1] The full rotational parameter range is the unit sphere; cf. eq. (4).

- The two kinds of error statistics agree, suggesting they are both reasonable error measures for the estimates.
- The accuracy of the estimator is sufficient for manipulation tasks across all tested variants.
- There is a trade off between rotational and translational accuracy.
- The highest rotational accuracy was achieved by estimators with the highest rotational resolution; the highest translational accuracy was achieved by the estimator with the lowest rotational resolution and the medium translational resolution in the second estimation stage.
- The run time increases for estimator variants with higher resolution of parameter space analysis; rotational resolution is much more expensive than translational resolution.

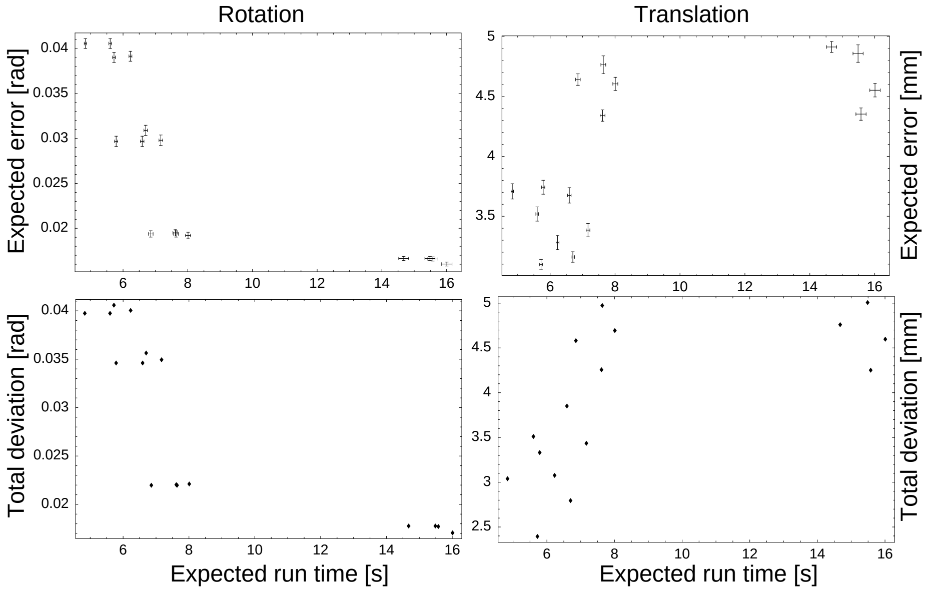

Fig. 4. Plots of error statistics versus run time statistics for the 16 tested variants of pose estimator: expected error $E(||\hat{\boldsymbol{p}} - \boldsymbol{p}_{\text{ref}}||)$ (bars indicate standard errors of the expectation values) and total deviation $\sqrt{\operatorname{Tr} \boldsymbol{C}}$, each for rotation and translation parameters.

The last observation can be explained by the fact that smaller buckets in parameter space need more sampling in order to get significantly filled. Sampling of full pose parameters is much more expensive than of just translations, as done in the second stage of estimation, which is why the translational resolution has less effect on run time than the rotational resolution in the present statistics.

A less intuitive result is the apparent trade off between rotational and translational accuracy. This suggests that, for the range of estimator variants here investigated, the allover estimation error does not greatly vary but is merely distributed to varying proportions between the rotational and translational degrees of freedom. As a consequence, one should use different variants of the estimator for rotational and translational parameters. This point, however, deserves further investigation.

It should be noted that the run times given are mainly a relative measure of the costs of the estimator variants. The absolute timings can be greatly improved by i) distributing parameter sampling and clustering across several CPUs, ii) building the model hash tables before execution of the estimators, and iii) some additional algorithmic optimizations.

Acknowledgments

I greatfully acknowledge help with acquiring the test data from Michael Suppa and Stefan Fuchs.

References

1. Harris, C., Stephens, M.: A combined corner and edge detector. In: Fourth Alvey Vision Conference. (1988) 147–151
2. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Intern. J. Comput. Vision **60** (2004) 91–110
3. Ballard, D.H.: Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition **13** (1981) 111–122
4. Stockmann, G., Kopstein, S., Benett, S.: Matching images to models for registration and object detection via clustering. IEEE Trans. Pattern Anal. Mach. Intell. **4** (1982) 229–241
5. Stockmann, G.: Object recognition and localization via pose clustering. CVGIP **40** (1987) 361–387
6. Illingworth, J., Kittler, J.: A survey of the Hough transform. CVGIP **44** (1988) 87–116
7. Moss, S., Wilson, R.C., Hancock, E.R.: A mixture model for pose clustering. Patt. Recogn. Let. **20** (1999) 1093–1101
8. Hillenbrand, U.: Consistent parameter clustering: definition and analysis. Pattern Recogn. Let. **28** (2007) 1112–1122
9. Stewart, C.V.: Robust parameter estimation in computer vision. SIAM Review **41** (1999) 513–537
10. Hirschmüller, H., Innocent, P.R., Garibaldi, J.: Real-time correlation-based stereo vision with reduced border errors. Int. J. Computer Vision **47** (2002) 229–246
11. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. J. Opt. Soc. Am. A **4** (1987) 629–642
12. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24** (2002) 603–619
13. Ott, C., Eiberger, O., Friedl, W., Bäuml, B., Hillenbrand, U., Borst, C., Albu-Schäffer, A., Brunner, B., Hirschmüller, H., Kielhöfer, S., Konietschke, R., Suppa, M., Wimböck, T., Zacharias, F., Hirzinger, G.: A humanoid two-arm system for dexterous manipulation. In: Proc. IEEE-RAS International Conference on Humanoid Robots. (2006) 276–283