Optimal Feature-set Selection Controlled by Pose-space Location

Klaus Müller, Ievgen Smielik and Klaus-Dieter Kuhnert
Institute of Real-Time Learning Systems, Department of Electrical Engineering & Computer Science,
University of Siegen, Hölderlinstr. 3, 57076 Siegen, Germany

Keywords: Feature Selection, Feature Combination, Model-based Pose Recovery, Pose Retrieval, Correspondence-based Pose Recovery.

Abstract: In this paper a novel feature subset selection method for model-based 3D-pose recovery is introduced. Many different kind of features were applied to correspondence-based pose recovery tasks. Every single feature has advantages and disadvantages based on the object's properties like shape, texture or size. For that reason it is worthwhile to select features with special attention to object's properties. This selection process was the topic of several publications in the past. Since the object's are not static but rotatable and even flexible, their properties change depends on there pose configuration. In consequence the feature selection process has different results when pose configuration changes. That is the point where the proposed method comes into play: it selects and combines features regarding the objects pose-space location and creates several different feature subsets. An exemplary test run at the end of the paper shows that the method decreases the runtime and increases the accuracy of the matching process.

1 INTRODUCTION

Model-based pose recovery is widely researched. A lot of the model-based recovery methods are correspondence-based and employ features to match query- and training-poses (see (Pons-Moll and Rosenhahn, 2011)). Many different types of features were introduced and finding best suitable one for a special problem has become part of research as well. Several comparisons of features for pose estimation have been published and many tests and comparisons have been done (Rosenhahn et al., 2006) (Chen et al., 2010) (Amanatiadis et al., 2011) (Kazmi et al., 2013). In so far it is known to the authors all the most of these methods apply global features or global feature sets for the whole pose space. In this paper a novel method is introduced, which selects and combines features depending on the location in pose-space. The idea behind this approach is, that the object's look is diverse depending on pose configuration. The bigger the pose-configuration changes, the more the look of the object varies and the probability that another feature set suits better increases. Based on this fact, the proposed method maps the feature space to the pose space and searches for the most discriminative subset depending on the region of the pose space. Finally it produces several feature subsets out of a pool of features. These subsets lead to a more accurate result while the dimensionality of the subset is reduced. That causes shorter computational time into two ways: on the one hand it reduces the feature extraction time, because just the features of the subset have to be extracted. On the other hand the matching time is reduced, since the dimensionality of the query- and the training-vector are downsized.

In the first part of the paper the theory of the method is explained. In the following a simple example is conducted and the results are evaluated.

2 RELATED WORK

There have been many publications on image based pose recovery. A lot of them deal with extraction of human poses. The introduced method is not specialized to humans pose extraction, but rather for general pose recovery of rigid and non-rigid objects. For this purpose many different kind of features were used: edge- or corner-based features (Choi et al., 2010) (Hinterstoisser et al., 2007), shape or silhouette based descriptors (Reinbacher et al., 2010) (Poppe and Poel, 2006), keypoint descriptors (Choi et al., 2010) (Collet et al., 2011) or self defined features like (Payet and Todorovic, 2011) and (Hinterstoisser et al., 2007).

Due to the fact that most of the descriptors are

Müller, K., Smielik, I. and Kuhnert, K-D.
Optimal Feature-set Selection Controlled by Pose-space Location.
DOI: 10.5220/0005724902000207
In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 4: VISAPP*, pages 200-207
ISBN: 978-989-758-175-5

all-purpose and not designed for a special problem, some of their components are irrelevant for pose retrieval but produce additional computational time. In order to increase the efficiency and the accuracy of the pose recovery process few of them combine and select features to a new compact problem specific descriptor. The method introduced by Chen et al. combines several shape descriptors and selects the components which are suitable for the current problem. They apply a variation of Adaboost to select descriptor components in several rounds to get an optimal compact descriptor(Chen et al., 2008). The method described in (Chen et al., 2011) also searches for the best global feature subset by using a more efficient feature selection method. Rasines et al. (Rasines et al., 2014) proposed a method to combine contour-based, polygon, blob and gradient-like features for hand pose recognition. They applied a sequential growing search algorithm to maximize the accuracy (F1 score) of the used classifier, while minimizing the size of the feature vector.

The proposed method of this paper also aims to combine features and select subsets. In contrast to the other methods it defines multiple subsets distributed all over the pose space. In order to get an optimal discriminative subset the entropy of each feature is calculated regarding the pose space location. In consequence the algorithm gives different subsets for discriminative poses.

3 FEATURE SUBSET SELECTION

The proposed method aims to select subsets $\tilde{F}_k$ of a feature set F with features f_1 to f_n for estimating 3D-poses from a 2D-pose space (rotation around pitch- and yaw-axis). The selection process is based on the features' entropy. Due to the fact that the entropy distribution is not uniform over the whole feature range, the method defines multiple local subsets in the feature space. The subset is chosen for every incoming request individually. For noise-cancellation the feature space is normalised by noise.

The method is divided in two parts: in the first offline step, the training set, consisting of images which describe all possible pose configurations (in constant step size s of a few degrees) with their ground truth data, is analysed and all features are calculated. Afterwards the feature values are mapped to the pose space and the entropy of each is calculated. In an additional step, subsets of features with the highest entropy combination are defined regarding the mapped pose space (see 3.1 - 3.3). Besides the definition of the subsets the feature with the best global entropy is selected. This feature is the key to choose the subset. In the second online step the "key"-feature is extracted out of the query image and depending on the result the subset is selected. All features of the subset are extracted out of the query image and a matching method searches for the closest neighbour (see 3.4).

This method is also capable for multidimensional features and feature descriptors and can also handle multidimensional pose-spaces. For reason of understandability 1-dimensional and 2-dimensional features are used. This makes visualisation (plotting) of features in the pose space possible.

3.1 Feature Normalization

Tests have shown that very noisy features can mislead to a high entropy and consequently to a wrong feature set. Due to that, a normalisation by noise is necessary to avoid misinformation.

The following method is applied to measure the standard noise deviation. Since the used pose space depends on two parameters (pitch and yaw angle), every feature f_n is plotted to the 2-dimensional parameter space. In figure 1 a fish model is used to demonstrate the 2D-pose space. The result matrix M_{f_n} is smoothed with the help of a normalized boxfilter and results in $\tilde{M}_{f_n}$. The standard deviation of the n-feature's noise σ_{f_n} is calculated as following. e and f describes the number of steps along the x- and y-axis of the plotted feature and $h = e \times f$ the size of the training set. $f_{n_{8,15}}$ for example means the value of feature f_n when the object is rotated $8 \times s$ around pitch-axis and $15 \times s$ around yaw-axis. s is the rotation angle step size in degree:

$$N_{f_n} = M_{f_n} - \tilde{M}_{f_n}, \quad N_{f_n}(e \times f) = \begin{pmatrix} f_{n_{1,1}} & f_{n_{1,2}} & \cdots & f_{n_{1,f}} \\ f_{n_{2,1}} & f_{n_{2,2}} & \cdots & f_{n_{2,f}} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n_{e,1}} & f_{n_{e,2}} & \cdots & f_{n_{e,f}} \end{pmatrix} \quad (1)$$

$$\overline{n}_{f_n} = \frac{1}{ef} \cdot \sum_{i=0}^{e} \sum_{j=0}^{f} f_{n_{i,j}} \quad (2)$$

$$\sigma_{f_n} = \sqrt{\frac{1}{ef} \cdot \sum_{i=0}^{e} \sum_{j=0}^{f} (f_{n_{i,j}} - \overline{n}_{f_n})^2} \quad (3)$$

The feature spaces are normalized with help of σ_{f_n} and the value range is shifted to zero

$$\hat{f}_{n_{e,f}} = \frac{(f_{n_{e,f}} - min(f_n))}{\sigma_{f_n}}. \quad (4)$$

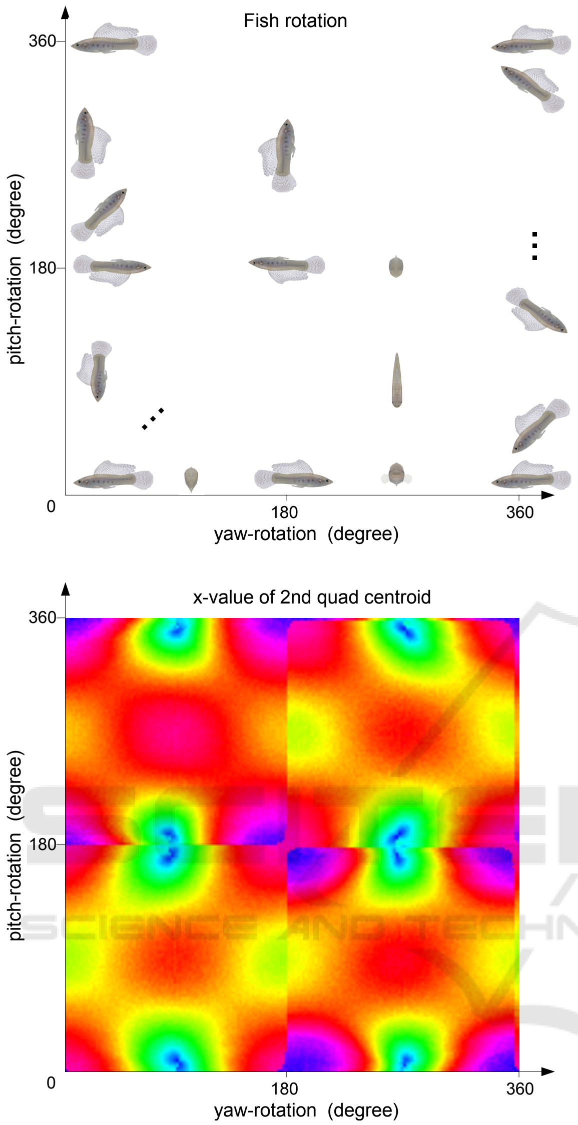

Figure 1: Overview of the pose space (top). Normalized feature vector (x-value of centroid) visualized along pitch and yaw rotation of the object (bottom).

The normalized values are stored in vector $\hat{f}_n$

$$\hat{f}_n = \begin{pmatrix} \hat{f}_{n_{0,0}} \\ \hat{f}_{n_{0,1}} \\ \vdots \\ \hat{f}_{n_{e,f}} \end{pmatrix} = \begin{pmatrix} \hat{f}_{n_0} \\ \hat{f}_{n_1} \\ \vdots \\ \hat{f}_{n_h} \end{pmatrix} \tag{5}$$

and brings the final normalized feature matrix $\hat{F}$.

$$\hat{F}(h \times n) = \begin{pmatrix} \hat{f}_{1_1} & \hat{f}_{2_1} & \cdots & \hat{f}_{n_1} \\ \hat{f}_{1_2} & \hat{f}_{2_2} & \cdots & \hat{f}_{n_2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{f}_{1_h} & \hat{f}_{2_h} & \cdots & \hat{f}_{n_h} \end{pmatrix} \tag{6}$$

After normalisation noisy features with little information have a small value range.

3.2 Feature's Entropy

The presented method is based on the feature's entropy in relation to the 2D-parameter space. At first a histogram with k different bins b_{n_k} is calculated out of the normalized feature vector $\hat{f}_n$, which has j_n elements. Every bin b_{n_k} has l_{n_i} elements. In a next step the entropy is calculated with

$$p_n = \sum_{i=0}^{k} \frac{l_{n_i}}{j_n} log_2(\frac{l_{n_i}}{j_n}). \tag{7}$$

The number of bins k is defined for each feature individually. Features with high information and less noise have a high number and noisy, weak features a little number of bins. Its number is calculated considering the normalized feature's value range. A manually chosen constant c (e.g. 100 bins) defines the bin number for the feature with the widest feature range. By increasing c the number of subsets raises and the number of pose configurations per bin falls. c should be chosen considering the size of training set. In the shown example c is around $\frac{1}{300}$ of the size of the training set. The bin size $|b|$ for all features is defined as following:

$$|b| = \frac{max(max(\hat{f}_i) - min(\hat{f}_i))}{c} \qquad with\ \ \hat{f}_i \in \hat{F} = (\hat{f}_0 \quad \hat{f}_1 \quad \ldots \quad \hat{f}_n). \tag{8}$$

Finally k is calculated by

$$k_i = \lceil \frac{\max(\hat{f}_i) - \min(\hat{f}_i)}{|b|} \rceil \qquad with\ \ \hat{f}_i \in \tilde{F} = (\hat{f}_0 \quad \hat{f}_1 \quad \ldots \quad \hat{f}_n). \tag{9}$$

3.3 Definition and Entropy Calculation of Feature Subsets

In a preprocessing step feature subsets for the whole configuration space are defined. At the beginning of the selection process the feature with the highest entropy is searched:

$$\hat{f}_p = \max(p_i) \qquad with\ \ i \in (0, 1 \ldots n) \tag{10}$$

Every bin b_{p_k} of $\hat{f}_p$ contains a set of pose configurations C_{p_k}. Vice versa all pose configurations C_{p_k} which have a similar value for feature $\hat{f}_p$ are combine in bin b_{p_k}. An example is shown in figure 2 b). Each color defines a bin b_{p_k} of $\hat{f}_p$ drawn over the pose configuration space. In the next step all configurations

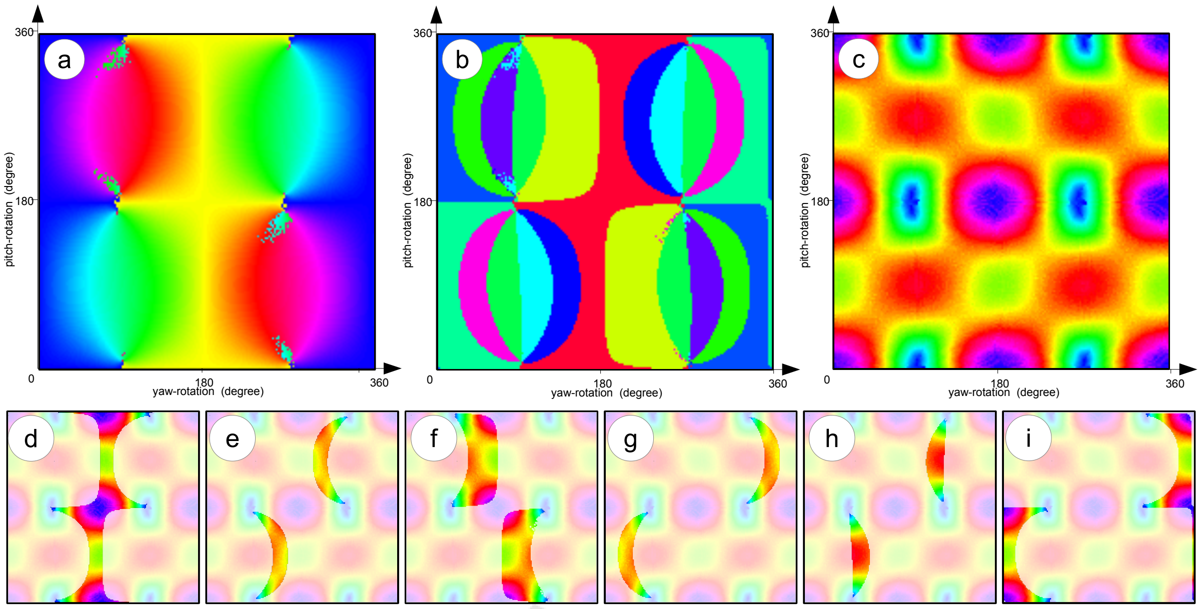

Figure 2: Process of local entropy calculation: (a) angle of major segment axis (here feature 1) (b) feature 1 is segmented by bins; segments are marked by different colors (c) deviation of feature 2 (d)-(i) bins of feature 1 are used to segment local regions of feature 2. The entropy of the segmented area is calculated.

C_{p_k} get an own feature matrix $\tilde{F_c}$. This matrix includes all features besides $\hat{f}_p$ in the range of C_{p_k}. An example is shown in figure 2 d)-i). Each figure d)-i) marks the values of a sample feature in the range of C_{p_k}. Finally the feature matrices $\tilde{F_c}$ get sorted by the entropy of each included feature vector f_{k_n} (see (7)).

$$\begin{aligned} &\tilde{F_c} = \begin{pmatrix} f_{c_0} & f_{c_1} & \dots & f_{c_r} \end{pmatrix} \\ &with \ \ c \in (0,1\dots k_p) \ \ and \ \ p(f_{k_0}) > p(f_{k_1}) \end{aligned} \tag{11}$$

In order to define feature subsets $\tilde{F_k}$ the number of features per subset $s \in (0,1...n-1)$ has to be defined manually. The higher s is, the higher is the computational power. The number of subsets k_p is equal to the number of $\hat{f}_p$-bins.

$$\tilde{F_k} = \begin{pmatrix} f_{c_0} & f_{c_1} & \dots & f_{c_s} \end{pmatrix} \tag{12}$$

3.4 Searching for the Right Feature Subset

After defining subset matrices $\tilde{F_k}$ in 3.3, the right subset has to found. Therefore the feature with the highest entropy $\hat{f}_p$ has to be calculated out of the query image. The subset $\tilde{F_d}$ is selected by sorting the result feature value d in the right bin b_{p_k}.

$$\tilde{F_d} = \tilde{F_k} \ \ if \ \ d \in b_{p_k} \tag{13}$$

4 EXPERIMENT

For a test a training set of 32400 artificial fish images showing fish rotated around the pitch- and yaw-axis, were used. The fish model was created in a project described in (Müller et al., 2014). Each image has a size of 500 x 400 pixels. A test set of 1000 images, showing randomly rotated fish, was applied in order to find the nearest neighbour within the 32400 images of the training set. Simple self defined texture and shape-based features (see 4.1) were chosen in order to make this experiment easy to understand.

4.1 Features

For the test several simple texture- and silhouette-based features were selected. These are shown in figure 3. In total 21 feature values were generated. A short description of the features is given in the following subsections.

4.1.1 Angle of Major Segment Axis

With the help of the image moments the angle of the silhouette's main component is calculated. After segmenting the silhouette the second image moment of the segment is calculated and is used to compute the orientation angle α of the major segment axis (see figure 3(a)).

4.1.2 Ratio of Width and Height

After rotating the segment around α the segment expansion in x- and y-direction is measured. The ratio between the lengths is used as feature (see figure 3(b)).

4.1.3 Centroids of Quadrants

As shown in 3(c)) the centroid of the segment's quadrants is used as another feature. At first the centroid of the rotated segment is calculated. This is used to separate the image in four quadrants: the quadrants are cut through the centroid parallel to x- and y-axis. Afterwards the centroid of each quadrant is calculated with the help of the image moments. The centroid's position is stored in reference to segment's centroid.

4.1.4 Size of Quadrants

Figure 3(d) shows the feature 'size of quadrants'. The area of every segment quadrant is calculated. In the figure each area has another color.

4.1.5 Position of Eye

With the help of a blob detector the eye of the sample fish is searched. The exact coordinate is finally defined by the centroid of the eye's area.

4.1.6 Snout and Positions around Eye

The last features describe the position of the snout (x,y) and the contour position above and below the eye's center. The points above and below the eye's center are defined by the intersection point of orthogonal of the major segment axis and the segment contour. All coordinates are stored in relation to the segment's centroid.

4.2 Implementation

The test application was implemented in *C++* and ran on an *Intel i-7* cpu with 3.4 Ghz. For the test all features of the training set were generated and stored in a vector. The feature with the highest entropy ('angle of major segment axis') was chosen as initial feature and 100 feature subset ranges were calculated regarding 3.3. Afterwards all 1000 test images were tested with subsets of size 5, 10, 15 and 20 and without subsets. Therefore the features' angle of major segment axis' was computed and the right subset was chosen by this feature value. With the help of a brute force algorithm the best fit training feature was searched. The quality of the match was defined by the rotational error between the test and the training image of a match. The runtime of feature extraction and matching was measured separately.

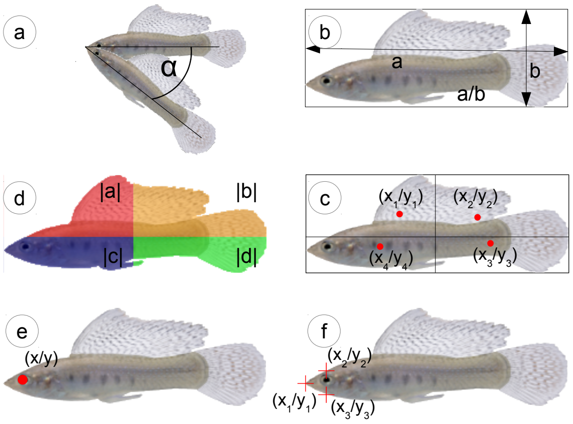

Figure 3: Features: (a) angle of major segment axis (b) ratio width - height (c) centroid of quarters (d) size of quarters (e) position of eye (f) position of snout contour pixel above and below eye.

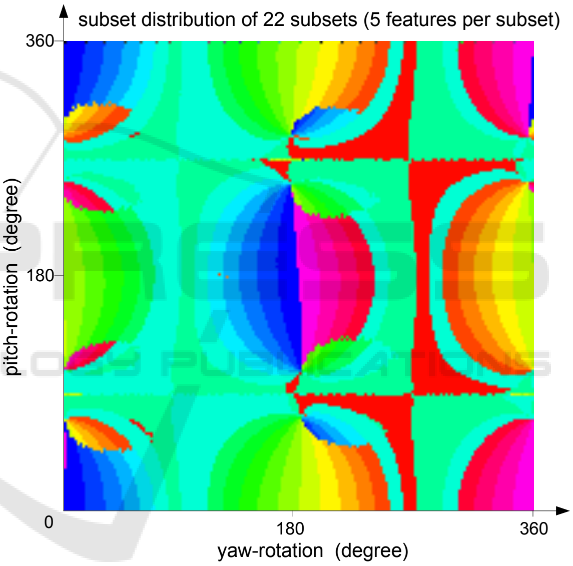

Figure 4: Distribution of subsets mapped to the pose space. Every color shows another subset.

4.3 Results

Besides the matching results the runtime of the matching process as well as the runtime of feature extraction is analysed in this section.

4.3.1 Pose Matching

As described in 4.2 in a first step 100 feature subset ranges were defined. Since different subset ranges can

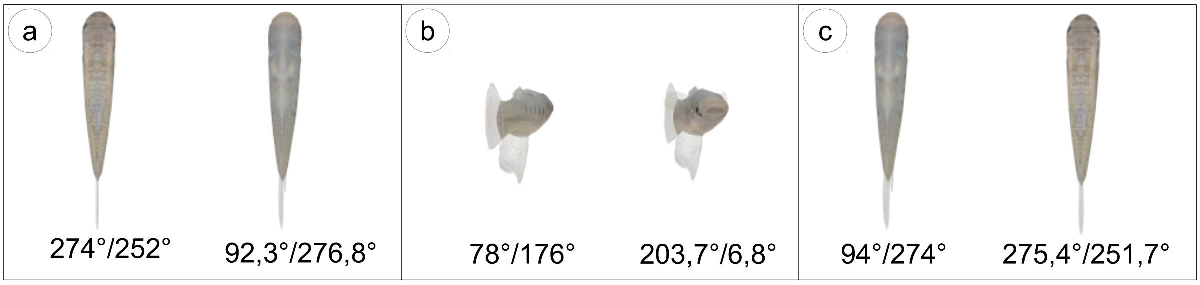

Figure 5: Outliers: some matches affect the result negatively. The images show some outliers, causing errors around 180 degrees. On the left side the training images and on the right side the test images are shown. The rotation angle of each images is found below (pitch-angle/yaw-angle).

include subsets with same features, the number of different subsets was smaller than 100. For subsets with 5 features the algorithm defines 22 different subsets (see figure 4), for 10 features 18 different subsets, for 15 features 21 different subsets and for 20 just 1 subset. This is caused by the fact, that 4 of the features have the worst entropy in all subset ranges and are never used. The quality of matching was measured by calculating the pose angle difference between test- and training image. The mean value of all angle differences was used to measure the quality of each method. Due to the fact that our training-database covers just rotations in 2-degree-steps, the expected mean error is 0.5 degrees. In general the mean-error-value of all methods was effected negatively by some outliers. Some of them are shown in figure 5. Since the fish in our test is very similar from top and bottom view, in some cases the process matched fish, which were rotated around 180 degrees. In order to reduce the influence of outliers in a second run the five nearest neighbours were searched and the best match was used to calculate the mean value. This is shown in figure 6.

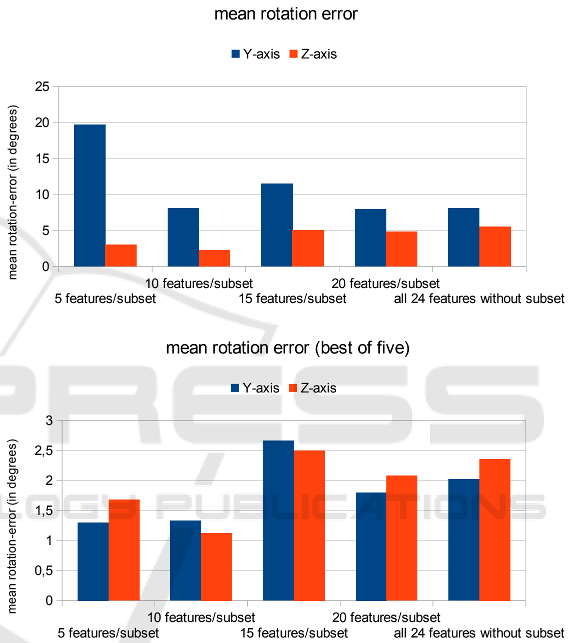

Figure 6: Mean matching error of all matches (top) and mean matching error of the best out of the five nearest neighbours (bottom).

4.3.2 Runtime

The shown method can save runtime in two different stages of the process. On the one hand it saves time during feature extraction. It is not necessary to extract all features out of the images, but only the features which are part of the chosen subset. The total runtime of the extraction process depends on the number of features per subset. On the other hand it helps to save runtime during the matching process. Depending on the matching technique the time reduction is more or less efficient, but in general matching algorithms are faster with less features. In the shown example a simple brute-force matching method was used. The runtime of different configuration was measured. At the top of figure 7 the runtime per query is shown. The brute-force-matcher needed up to 41% less runtime with a small subset than with all features. At the bottom of the figure the time for feature extraction is shown. Due to the fact that some of the used features depend on each other (see 4.1), in some configurations all features were calculated while not all of them was used. In spite of everything the runtime of feature extraction was reduced by more than 50% for subsets with 10 features and less.

5 CONCLUSION

In this work a novel approach for pose-depending subset feature selection is proposed. In contrast to the most other feature-based methods this method aims to

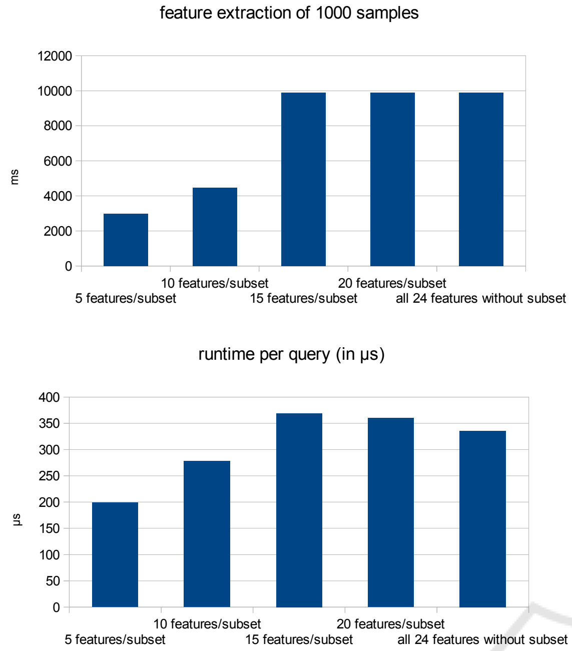

Figure 7: (Top) Runtime of a single matching query ordered by the number of features per subset. (Bottom) Runtime of feature extraction process with 1000 test images.

select pose-sensitive, significant local subsets, which fit optimal to the depending pose-space region. In a simple experiment it could be shown, that the method decrease the runtime of feature extraction up to 50% and of the matching process up to 41%. Depending on the number of features and matching-method efficiency can even be improved. During this test-run the accuracy of the pose matching was increased as well, in case the number of features per subset was not chosen to small. For future it is planned to apply this method to a fish tracking system with multiple degree-of-freedom fish models and contour- and keypoint-based features.

ACKNOWLEDGEMENTS

The presented work was developed within the scope of the interdisciplinary, DFG-funded project virtual fish of the Institute of Real Time Learning Systems (EZLS) and the Department of Biology and Didactics at the University of Siegen.

REFERENCES

Amanatiadis, A., Kaburlasos, V. G., Gasteratos, A., and Papadakis, S. E. (2011). Evaluation of shape descriptors for shape-based image retrieval. *Image Processing, IET*, 5(5):493–499.

Chen, C., Yang, Y., Nie, F., and Odobez, J.-M. (2011). 3d human pose recovery from image by efficient visual feature selection. *Computer Vision and Image Understanding*, 115(3):290–299.

Chen, C., Zhuang, Y., and Xiao, J. (2010). Silhouette representation and matching for 3d pose discrimination–a comparative study. *Image and Vision Computing*, 28(4):654–667.

Chen, C., Zhuang, Y., Xiao, J., and Wu, F. (2008). Adaptive and compact shape descriptor by progressive feature combination and selection with boosting. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

Choi, C., Christensen, H., et al. (2010). Real-time 3d model-based tracking using edge and keypoint features for robotic manipulation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4048–4055. IEEE.

Collet, A., Martinez, M., and Srinivasa, S. S. (2011). The moped framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, page 0278364911401765.

Hinterstoisser, S., Benhimane, S., and Navab, N. (2007). N3m: Natural 3d markers for real-time object detection and pose estimation. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–7. IEEE.

Kazmi, I. K., You, L., and Zhang, J. J. (2013). A survey of 2d and 3d shape descriptors. In *Computer Graphics, Imaging and Visualization (CGIV), 2013 10th International Conference*, pages 1–10. IEEE.

Müller, K., Schlemper, J., Kuhnert, L., and Kuhnert, K.-D. (2014). Calibration and 3d ground truth data generation with orthogonal camera-setup and refraction compensation for aquaria in real-time. In *VISAPP (3)*, pages 626–634.

Payet, N. and Todorovic, S. (2011). From contours to 3d object detection and pose estimation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 983–990. IEEE.

Pons-Moll, G. and Rosenhahn, B. (2011). Model-based pose estimation. In *Visual analysis of humans*, pages 139–170. Springer.

Poppe, R. and Poel, M. (2006). Comparison of silhouette shape descriptors for example-based human pose recovery. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 541–546. IEEE.

Rasines, I., Remazeilles, A., and Iriondo Bengoa, P. M. (2014). Feature selection for hand pose recognition in human-robot object exchange scenario. In *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*, pages 1–8. IEEE.

Reinbacher, C., Ruether, M., and Bischof, H. (2010). Pose estimation of known objects by efficient silhouette matching. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1080–1083. IEEE.

Rosenhahn, B., Brox, T., Cremers, D., and Seidel, H.-P. (2006). A comparison of shape matching methods for contour based pose estimation. In *Combinatorial Image Analysis*, pages 263–276. Springer.