Generalised Pose Estimation Using Depth

Simon Hadfield and Richard Bowden

Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, England, GU2 7XH
{S.Hadfield,R.Bowden}@surrey.ac.uk

Abstract. Estimating the pose of an object, be it articulated, deformable or rigid, is an important task, with applications ranging from Human-Computer Interaction to environmental understanding. The idea of a general pose estimation framework, capable of being rapidly retrained to suit a variety of tasks, is appealing. In this paper a solution is proposed requiring only a set of labelled training images in order to be applied to many pose estimation tasks. This is achieved by treating pose estimation as a classification problem, with particle filtering used to provide non-discretised estimates. Depth information extracted from a calibrated stereo sequence, is used for background suppression and object scale estimation. The appearance and shape channels are then transformed to Local Binary Pattern histograms, and pose classification is performed via a randomised decision forest. To demonstrate flexibility, the approach is applied to two different situations, articulated hand pose and rigid head orientation, achieving 97% and 84% accurate estimation rates, respectively.

Keywords: pose, depth, stereo, head, hand, classification, particle filter, gesture, lbp, rdf, background suppression, object extraction, segmentation.

1 Introduction

In this paper, the problem of performing pose estimation on complex objects using classification is addressed. This is a difficult problem due to the variability of objects, which may be rigid, deformable or articulated. To solve this problem, the pose space is segmented into regions, and the problem is treated as one of classification.

The proposed framework generates depth via dense stereo point correspondence. These depth maps are used in several ways, to suppress image clutter by removing pixels at depths above or below the detected objects depth, to estimate the expected scale of objects, and to provide an additional channel of features during pose classification.

In [13] and [7] pose estimation is performed in a model based framework. When applied to articulated objects such as the human hand, this allows estimation of each joint angle individually. However, a model based framework is unsuitable

K.N. Kutulakos (Ed.): ECCV 2010 Workshops, Part I, LNCS 6553, pp. 312–325, 2012.

for generalised pose estimation, because of the need to build and integrate a specific object model.

In [3] pose estimation is treated as a regression problem, where the output of the regressor corresponds to the pose parameters. This requires only labelled training data in order to be applied to a new problem, making it a more suitable approach for generally applicable pose estimation. Unfortunately if the pose to be estimated has multiple parameters, regression is not simply applied. Few regressors are able to output multiple parameters, meaning a regressor must be used for each output. Again this limits generalisation, if the tasks are sufficiently different.

In this paper a classification methodology is used. As with regression, this means a system may be retrained to a new problem simply by providing labelled examples. However unlike regression the output values need not be continuous, allowing multi-dimensional poses with a single classifier. A pose space of any dimensionality may be segmented into regions, each assigned a label. The resulting tensor can then be flattened into a list of class labels. This way the output class of the classifier simultaneously encodes all pose parameters. The drawback of this, is the discretisation of the pose parameter outputs, this is countered using tracking techniques as discussed in section 2.5.

Two different, widely encountered, pose estimation tasks, are used to test the proposed framework. Head orientation estimation involves a rigid object, and can be used in gaze estimation, useful in studying consumers response to billboards [5], and employees behaviour during meetings [1]. Hand pose estimation, provides a problem with an articulated object, and is useful in Human-Computer Interaction and Sign Language Recognition. Although the head has fewer degrees of freedom (especially considering that roll does not affect gaze direction), a useful framework must be able to distinguish small movements of the head. This leads to a large number of classes with high inter-class similarities. On the other hand, the hand shape problem has a small number of classes each with very wide intra-class variations due to the objects articulation.

The remainder of this paper is structured as follows. Initially, the overview of the general pose estimation framework is explained in section 2, then each element is discussed in turn (2.1 to 2.5). The results, section 3, examines the performance of different feature variants, and the value of depth information in each of the two tasks. Then the application of particle filtering techniques 3.2 is discussed. Section 4 provides information on the interactive demonstration system. Finally conclusions are drawn on the general applicability of a pose estimation framework based on classification, and the use of depth.

2 Framework Overview

The proposed pose estimation framework, makes extensive use of depth data, which provides fast and simple background suppression [6] and a useful prior on object scale. During testing, the usefulness of depth as an additional channel for generating object features is also demonstrated. As in figure 1, a pair of cameras capture a left and right image of the scene. Stereo point correspondence is

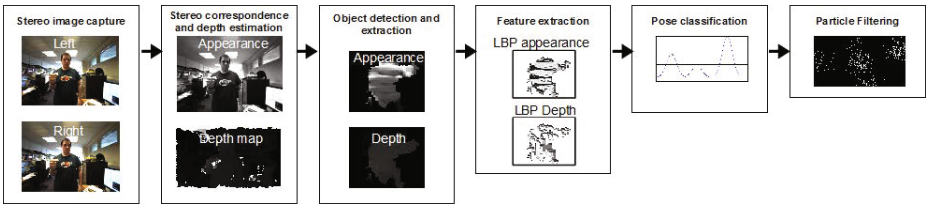


Fig. 1. Proposed framework, for real time, generalised pose estimation. Example appearance and depth images are included at each stage.

then performed to generate the depth image. Object detection extracts object candidates, and background suppression is performed using the depth map. The appearance and depth images for the extracted object are then converted to a Local Binary Patterns (LBP) [9] texture representation. This texture representation is input to a previously trained randomised decision forest classifier [2].

Due to the ambiguity of adjacent poses, the discretised pose classification can then be integrated into a particle filter framework [4], to apply temporal constraints and provide a continuous output estimate.

2.1 Stereo Correspondence and Depth Estimation

The depth information is extracted via stereo point correspondence, from a PointGrey Bumblebee2 stereo camera system. The mask size used causes an unfortunate trade-off between sparsity and accuracy. Smaller masks are harder to match, but provide finer details. In order to provide a more dense depth image, stereo reconstruction is performed with various mask sizes. The images are then combined, using the smallest mask size wherever possible. Figure 2 demonstrates this idea, showing a sequence of images each of which has had unmatched pixels from the previous image, filled in by a depth map captured at larger mask size.

A set of depth maps D was generated from the set of stereo masks S by performing stereo point matching on the left and right images (L and R respectively). Where $S = \{15 \times 15, 7 \times 7, 5 \times 5, 3 \times 3\}$ and $stereo_{S_i}$ represents stereo matching with the ith mask.

$$D_i = stereo_{S_i}(L, R) \tag{1}$$

The output depth map O is then created by selecting each pixel value O^p from the corresponding pixel values D_i^p, where D_i is the depth map from the ith stereo mask.

$$O^p = \begin{cases} D_i^p & D_i^p \neq NULL \\ D_{i+1}^p & otherwise \end{cases} \tag{2}$$

Fig. 2. Combining multiple depth maps. Each successive image is the previous image, combined with an of higher mask size.

2.2 Object Detection and Extraction

If the object whose pose is to be estimated, is a subregion of a larger image, then initially the object must be detected. This step is task specific. In the example experiments, head location is extracted using the well known, cascade of boosted haar-feature classifiers technique [12].

For hand detection a similar detector could be used, however due to the variability possible in human hands it requires large amounts of data to train, and performs significantly worse than with faces [11]. Many other hand detectors simplify the problem, by using segmentation techniques. Segmentation can be performed using background suppression, coloured gloves, motion detection, or skin segmentation [8]. In every case this imposes a restriction on general applicability. Instead, in this paper depth images are used to segment the hand, utilising the fact that when gesturing at the system, the hand is extended in front of the body.

Using the weak perspective camera model the scale (S) of the object in the image plane stretches between two depths (z_2 and z_1). Thus the resultant scale of an object in the image plane, can be determined by the distance in depth, from an object of known image scale, if their base scale ratio (B) is known, as in equation 3, where f is the focal length of the camera. In this case the base scale ratio from the face to the hand is taken as 1.2, based on the measurements of the "Vitruvian Man".

$$S = B\left(1 + f\left(\frac{1}{z_2} - \frac{1}{z_1}\right)\right) \tag{3}$$

In both tasks, the depth is then used for background suppression. After an object is detected, the median depth of that object is taken. Every image point, with a depth distance further from the median than the expected object size, is suppressed in both the intensity and depth images. This simple heuristic allows operation in noisy and cluttered scenes, without the need for more complicated detection strategies. Background clutter of similar depth to the object is not suppressed by this method, however the objects location and scale have already been estimated, so there is generally little clutter within the small region of interest. See section 3.2 for the specific performance increase using background suppression.

Figure 3 illustrates the hand detection and segmentation. In the first image, the face and closest region of depth are detected, represented by the red circle and yellow dot respectively. The scale of the hand is estimated from the depth difference, and represented by the green box. The second image shows the intensity after background suppression is performed on the 2 objects.

(a) (b)

Fig. 3. Hand detection and segmentation: (a) Unsegmented depth image showing face detection (red circle) and nearest point detection (yellow dot), with estimated hand scale (green box). (b) Hand and face appearance after background suppression via depth.

2.3 Feature Extraction

For feature extraction, Local Binary Pattern (LBP) texture features were selected, providing invariance to monotonic value changes, translating to resistance to illumination changes in appearance and object distance in depth. These features are highly customisable, with the possibility for rotational invariance[10], tunable accuracy and multiple scales. Feature extraction is performed in both the appearance and depth channel.

LBPs describe an image in terms of a histogram of micro-texture components (edges, corners, dark points and light points in the intensity channel, ridges, contours, peaks and depressions in the depth channel). For basic LBP features, every pixel in the image is labelled by taking a 3×3 neighbourhood and thresholding each point by the value of the centre pixel. The result is an 8 bit long binary number labelling the pixel.

$$LBP = \sum_{i=0}^{7} \begin{cases} 2^i & f_i \geq f_c \\ 0 & otherwise \end{cases} \tag{4}$$

LBP features were extended to capture texture components at different scales, and also to allow for variable accuracy. The operator LBP(P,R) indicates that, rather than a 3×3 neighbourhood, P points are sampled uniformly around the centre, at a radius R. So R controls feature scale detected, and P controls the length of the output label (and so the size of the feature vector). However there is a limit on the detail possible in the features, dependant on the scale. If P is greater than the number of distinct pixels falling along a circle of radius R, then the new bins being added to the feature histogram are redundant

It was also shown that for most images, 90% of the LBP labels tend to belong to a small subset of the 2^P possible patterns. These patterns were termed "uniform" LBPs and are characterised by having at most two transitions between 0 and 1 in their binary representation. Ojala et al. claim that the removal of these unstable histogram bins also improves classification performance, however our experiments show that if the dataset is large enough, their removal decreases performance.

Another variant of the LBP operator is to add rotational invariance. In order to achieve this, the LBP for every pixel is bit-shifted until the minimum value is found, and this minimum value is used as a label. Equation 5 defines this conversion, where $shift_i$ represents a binary shift of i bits.

$$LBP^{ri} = min_{i=0}^{P} \left(shift_i \left(LBP^{(P,R)} \right) \right) \tag{5}$$

This gives an even greater reduction in feature vector size than uniform LBPs. It is also possible to apply both variants, and use rotationally invariant, uniform LBPs. Histograms of LBP features, for a single LBP variant v, are labelled LBP_v. Several different variant histograms may be concatenated, to provide additional features. These multi-variant histograms may be computed across a subregion r of the object, providing a description of the local texture in that region labelled HR_r. Concatenating these region histograms together forms the feature vector HI_i for the image i. Finally concatenating image histograms for both the depth and appearance images gives the objects feature representation H.

$$\begin{aligned} HR_{ir} &= \{LBP_0, \ldots, LBP_v\} \\ HI_i &= \{HR_0, \ldots, HR_r\} \\ H &= \{HI_0, HI_1\} \end{aligned} \tag{6}$$

In section 3, the exact effects of the specific feature variants on performance in different tasks is demonstrated. Additionally, by normalising the histogram of textures, the features become invariant to the scale of the detected object.

2.4 Pose Classification

A random forest is an ensemble classifier where a large number of decision trees are grown based on random subsets of the data. This allows each of the trees to capture different aspects of class separability. The outputs of these weak classifiers are then combined to act as a strong classifier. In this paper the randomised forest toolkit from alglib.net was used, with a forest of 100 trees, grown at a ratio of 0.6.

The advantage of a random forest, is that it provides a likelihood distribution L over all classes c, given the input observations H. This allows likelihoods to be estimated between classes, somewhat mitigating the drawback of a classification based approach. This likelihood distribution also proves to be an advantage in section 2.5 where it is used in a particle filtering framework.

$$L(c) = P(H|c) \tag{7}$$

2.5 Particle Filtering

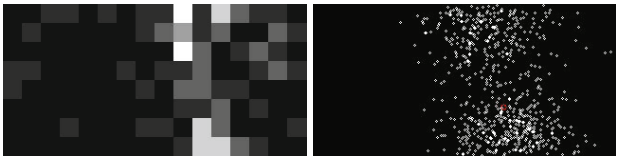
The particle filter takes the output of the classification stage as an observation likelihood, and combines it with the prior probability of the class $P(c)$, based on the previous system state and system dynamics. From Bayes theorem, the probability of each class given the new observation, is given by:

$$P(c|H) \propto L(c) P(c) \tag{8}$$

The particle filter approximates $P(c)$ with a number of weighted hypotheses, which are modified from the previous state based on the dynamics of the system with some stochastic diffusion. A resampling step is used to ensure that the higher probability portions of the distribution are more accurately estimated at the next iteration, using a larger number of hypotheses. Each hypothesis in the previous iteration generates a number of new hypotheses, based on it's normalised weight. Equation 9 illustrates the resampling technique, where $Quant_i$ represents the ith quantile of a distribution. W is the function of normalised hypothesis weights, n is the total number of hypotheses, and S_t is the set of hypotheses at time t.

$$S_{t+1}(i) = S_t \left(Quant_{i/n} \left(\int W \right) \right) \tag{9}$$

Figure 4 shows an example output from the pose classification system (a), being applied to the particle filter. Initially the particles are uniformly distributed. After the classification output is applied, the particles converge towards the peaks of the distribution (b), with more particles centred around higher peaks. This pose tracking allows the pose estimate to be continuously valued, despite initially using a discrete classification methodology.



(a) (b)

Fig. 4. The likelihood distribution (a) across the pose classes, is applied to the pose tracker. The positions of the hypotheses after application of the new likelihoods is shown in (b).

3 Results

Datasets were captured for each task, as there are few pre-existing pose datasets containing appearance and depth information. Both datasets are comprised of subjects from various ethnicities and genders. Performance was measured using

5 fold cross validation, with a random split of 70% training, 30% test images. The training set in each case was enriched by adding small amounts of scale and translation variation to each image. Specifically, each image was translated in all 4 directions by 5% and 10% of it's size, creating 8 additional images, and then the image was enlarged and shrunk by 5% and 10% producing an additional 4 images. Specific details about the individual datasets are provided at the start of the following two sections.

3.1 Hand Pose Classification Results

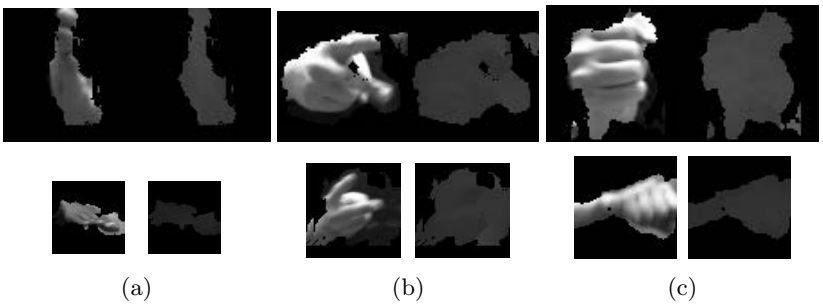
A test situation for hand pose was required, where the lexicon consisted of a small number of static gestures. A Rock, Paper, Scissors game was determined as a suitable candidate for the trial (see section 4). A dataset of depth and appearance images was created for each of the 3 poses. Seven subjects, including male and female Caucasians, one Indian, one Nigerian and one Asian were asked to create the specific gesture at different orientations and positions. In total 2100 appearance and depth image pairs were captured per symbol (before enrichment). A random selection of image pairs from this dataset is shown in figure 5. Performance was measured with a number of different feature variants, as shown in table 1.

The first 3 rows of the table illustrate the value of depth. Testing entirely without the influence of depth is impossible in this task, as it is required for object detection, however shape features may be removed from the classification stage. Classification based on depth and appearance features both achieve respectable performance levels, while the combination of the two improves over either alone.

Standard LBP features provide excellent performance. Utilising features across scale does provide slightly improved performance. In this task, class discrimination is based upon finger location, which may be poorly represented at higher scales. Using Uniform LBPs caused little change, implying that micro-texture components useful for determining finger positions are mostly uniform patterns. This is useful, as removing these patterns means a smaller feature vector, improving both training and running times for the classifier.

Rotationally invariant LBPs perform significantly worse in all cases, compared to their rotationally variant counterparts. This is likely because rotational variations are so well represented in the dataset, that implementing the invariance within the features is unnecessary.

The confusion matrix is shown in table 2. The performance on the rock and paper class is significantly higher than on the scissors class. Although scissors examples suffer from higher class confusion, few rock or paper images are classified as scissors. The most prominent features of the scissors class are the two extended fingers. Due to pose, often only the tips of these fingers are visible. So the number of image points useful for identifying a scissors shape may be low.



(a) (b) (c)

Fig. 5. Two randomly selected appearance and depth image pairs from the dataset for (a) Paper, (b) Scissors and (c) Stone. The scale variation between images of the dataset is apparent here.

Table 1. Hand pose classification, operating with different variants of LBP features. LBP^U are uniform, and LBP^R are rotationally invariant LBPs.

Feature type	Average correct classification	Standard deviation
Un-enriched, Greyscale channel	0.8929	0.0079
Un-enriched, Depth channel	0.8623	0.0054
Un-enriched, Both channels	0.9083	0.0040
LBP(8,1)	0.9689	0.0006
LBP^U(8,1)	0.9656	0.0013
LBP^R(8,1)	0.8865	0.0018
LBP^{UR}(8,1)	0.8593	0.0014
LBP^U(8,1) and LBP^U(8,2)	0.9693	0.0043
LBP^R(8,1) and LBP^R(8,2)	0.8932	0.0022

Table 2. Confusion matrix of hand classification, using uniform, multi-scale (8,1) (8,2) LBPs. Rows are predicted classes and columns are true classes.

	Rock	Paper	Scissors
Rock	0.9740	0.0102	0.0188
Paper	0.0200	0.9822	0.0315
Scissors	0.0060	0.0076	0.9497

3.2 Head Orientation Results

The head pose parameters affecting pose direction are pan angle and tilt angle, these 2 dimensions were segmented into a series of classes at 10 degree intervals. Five subjects, including male and female Caucasians, A Nigerian, and a Middle-eastern subject, were required to sit in a fixed position and look at markers placed at each class angle. Haar feature cascades picked out the faces and the background was suppressed using depth. This dataset was far sparser than the

Table 3. Head pose estimation on isolated images, using different types of LBP features and with different usage of depth. LBP^U are uniform, and LBP^R are rotationally invariant LBPs.

Test mode	Average exact classification	Classification within 10 degrees	Standard deviation
No seg. colour features	0.1464	0.6584	0.0202
No seg. depth and colour features	0.1911	0.7225	0.0069
Seg. colour features	0.1691	0.6801	0.0145
Seg. depth features	0.2052	0.7064	0.0114
Seg. depth and colour features	0.2010	0.7398	0.0113
LBP(8,1)	0.2010	0.7398	0.0113
LBP^U(8,1)	0.2845	0.8364	0.0052
LBP^R(8,1)	0.1981	0.6957	0.0103
LBP^{UR}(8,1)	0.2817	0.8107	0.0062
LBP^U(8,1) and LBP^U(8,2)	0.2870	0.8362	0.0094
LBP^R(8,1) and LBP^R(8,2)	0.2043	0.7017	0.0156

hand data, with 153 different classes, and 1-3 images per subject, per class (2200 pairs of appearance and depth images in total). This sparse dataset makes the task far more difficult, and reinforces the need for a classification based method, capable of operating with little training. As discussed above, situations with sparse datasets such as this, may use feature customisation to incorporate some invariances which are not in the dataset, directly into the feature representation.

The other difficulty with this dataset is the inconsistency of the data. Ten degrees rotation is difficult to capture accurately for the human head, as subjects naturally tend to move their eyes, rather than their heads when looking at close, new objects. This means the dataset tends to have movement between classes of anywhere from 0 to 10 degrees, with the remainder made up by eye motion. Randomly selected example images from the dataset are shown in figure 6.

(a) (b) (c)

Fig. 6. Three randomly selected appearance and depth image pairs from the head orientation dataset. (a) -90 degrees pan, -10 degrees tilt. (b) +20 degrees pan, +30 degrees tilt. (c) +10 degrees pan, -20 degrees tilt. Note that scale variations are included in the dataset.

Head Pose Classification. Tests were initially performed on isolated images, using a range of feature variants (Table 3). Classification performance is listed for classifying within 10 degrees of the listed value, reflecting the probable range within the data, as mentioned previously. Using depth to suppress the background from detected objects improves performance by 1%-2% by removing clutter from the images. Using depth as the only feature channel, is more accurate than the standard appearance channel features. However the most effective system utilizes the combination of both feature channels to provide 4% improved performance.

Standard LBP features achieve a respectable 74% classification rate. As expected, the sparse dataset is unable to cover the variations in the classes. Customising the features to suit the task, yields improved results, with uniform LBPs providing the best performance. Due to the sparseness of the dataset, non-uniform feature bins are unstable, and when present, are mistakenly chosen as discriminatory.

As in the hand pose tests, the results show only a marginal improvement when using features from multiple scale, while using rotationally invariant LBPs causes a considerable drop in performance. This is to be expected as the test dataset does not contain roll variation, and so the rotational invariance is unnecessary.

Pose Tracking Framework. Head pose estimation was also performed on a continuous sequence, rather than a set of isolated images. For this test the particle filtering framework was enabled. The sequence contains partial and complete occlusions of the subjects face, and also frequent, sudden, changes in direction. The results are shown in table 4.

Table 4. Head pose estimation on a continuous sequence with and without pose tracking

Mode	Exact classification	Classification within 10 degrees	Average pan error	Average tilt error
Per frame classification	0.0885	0.4712	N/A	N/A
Pose tracking	0.1081	0.6414	10.0	10.6

As expected, applying temporal constraints is useful when determining the current pose. As a result, 15% more examples were classified correctly over isolated classification. In both dimensions the average error angle is roughly one class. Coupled with the fact that 64% of frames are classified within 10 degrees, it can be inferred that most miss-classified examples lie within two classes.

Figure 7 shows the confusion matrices before (a) and after (b) the pose tracking framework was used. The two dimensional arrangement of pan and tilt classes has been flattened into a vector. The tilt angle changes most rapidly, with the pan angle changing every 9 classes. This means that points which are 9 classes apart in the confusion matrix, are in reality only 10 degrees apart. This can be observed in the confusion matrix by the multiple diagonal lines, at 9 class intervals.

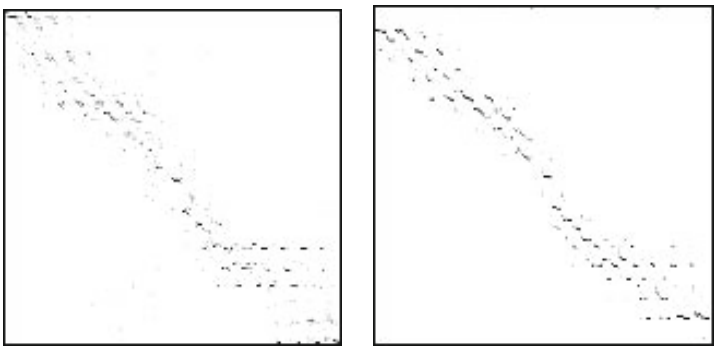


Fig. 7. Confusion matrices, (a) without and (b) with pose tracking, for the 153 class head pose task. Darker pixels indicate greater classification rates. The average correct classification rates (within 10 degrees) are 47%, and 64% respectively.

In the first image (without tracking) there are fewer diagonals visible, and each diagonal is more sharply defined. These two features relate to lower average confusion in tilt and pan respectively. In both cases there are very few extreme outliers, meaning the classification system is able to accurately find the correct region of pose space. A prominent feature of the confusion matrices is the increased number of diagonals present at extreme classes, compared to the central classes. From this it can be deduced that tilt angle is easily determined for a frontal face, but for profile faces (high pan angles) there is greater confusion in the tilt dimension.

4 Demonstration

In order to demonstrate the systems real-time performance, an interactive demonstration system was built around the hand pose task. This demonstration uses an animated avatar as an opponent for a user to play Paper, Scissors, Stone against. Figure 8 shows an image of the demonstration system in use. A video of the system is also available at http://www.youtube.com/watch?v=SRfQFOMSH3A.

Fig. 8. Interactive demonstration of hand pose estimation in a Paper, Scissors, Stone

5 Conclusions

In this paper, a method was demonstrated, for estimating continuous pose, by segmenting the pose space into classes and treating it as a classification problem. The applicability of such a framework to varied pose estimation tasks, and the rapid retraining time has also proved it a viable method for generalised pose detection. Such a framework has proven capable of real time performance, with this implementation, image capture and stereo reconstruction required roughly 200ms, while estimating the pose took on average 5ms.

The usefulness of depth data during pose estimation has been demonstrated, both as a tool for object extraction, and an additional channel for feature extraction, granting considerable improvements in both tasks. The possibility for systems built on this framework to be customised to handle inadequate training data is also apparent, by modifying the features to incorporate extra invariances, or remove noisy features. Finally, a method for using particle filtering to overcome the limitations of a classification based approach was proven to increase performance by incorporating temporal information into the pose estimate.

Acknowledgments. This work is supported by the EPSRC project LILiR (EP/E027946) and the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 231135 - Dicta-Sign.

References

1. Ba, S.O., Odobez, J.M.: Recognizing Visual Focus of Attention From Head Pose in Natural Meetings. IEEE T. Syst. Man. Cyb. 39, 16–33 (2009)
2. Breiman, L.: Random Forests. Mach. Learn. 45, 5–32 (2001)
3. de Campos, T.E., Murray, D.W.: Regression-based Hand Pose Estimation from Multiple Cameras. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 782–789. IEEE Press, New York (2006)
4. Isard, M., Blake, A.: CONDENSATION - Conditional Density Propagation for Visual Tracking. Mach. Learn. 29, 5–28 (1998)
5. Lablack, A., Maquet, F.: Visual gaze projection in front of a target scene. In: IEEE International Conference on Multimedia and Expo, pp. 1839–1840. IEEE Press, New York (2009)
6. Malassiotis, S., Strintzis, M.G.: Robust real-time 3D head pose estimation from range data. Pattern Recogn. 38, 1153–1165 (2005)
7. Marras, I., Nikolaidis, N., Pitas, I.: 3D head pose estimation in monocular video sequences by sequential camera self-calibration. In: IEEE International Workshop on Multimedia Signal Processing, pp. 1–6. IEEE Press, Brazil (2009)
8. Mitome, A., Ishii, R.: A comparison of hand shape recognition algorithms. In: Annual Conference of the IEEE Industrial Electronics Society. IEEE Press, Virginia (2003)
9. Ojala, T., Pietikainen, M., Harwood, D.: A Comparative Study of Texture Measures with Classification Based on Feature Distributions. Pattern Recogn. 29, 51–59 (1996)

10. Ojala, T., Pietikainen, M., Topi, M.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE T. Pattern Anal. 24, 971–987 (2002)
11. Ong, E.J., Bowden, R.: A boosted classifier tree for hand shape detection. In: IEEE International Conference on Automatic Face and Gesture Recognition, pp. 889–894. IEEE Press, Korea (2004)
12. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 511–518. IEEE Press, Hawaii (2001)
13. Zhenyao, M., Neumann, U.: Real-time Hand Pose Recognition Using Low-Resolution Depth Images. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1499–1505. IEEE Press, New York (2006)