2010 International Conference on Pattern Recognition

Dynamic Hand Pose Recognition using Depth Data

Poonam Suryanarayan
The Pennsylvania State University
University Park, PA
Email: poonam@psu.edu

Anbumani Subramanian, Dinesh Mandalapu
Hewlett-Packard Labs
Bangalore, India
Email: {anbumani, dinesh.mandalapu}@hp.com

***Abstract*—Hand pose recognition has been a problem of great interest to the Computer Vision and Human Computer Interaction community for many years and the current solutions either require additional accessories at the user end or enormous computation time. These limitations arise mainly due to the high dexterity of human hand and occlusions created in the limited view of the camera. This work utilizes the depth information and a novel algorithm to recognize scale and rotation invariant hand poses dynamically. We have designed a volumetric shape descriptor enfolding the hand to generate a 3D cylindrical histogram and achieved robust pose recognition in real time.**

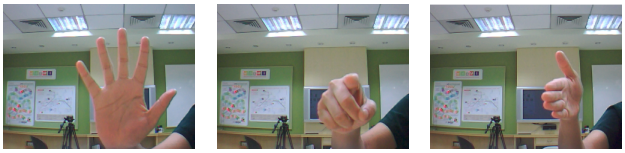
***Keywords*-Depth Camera, Gesture, Shape Descriptor, SVM.**

I. Introduction

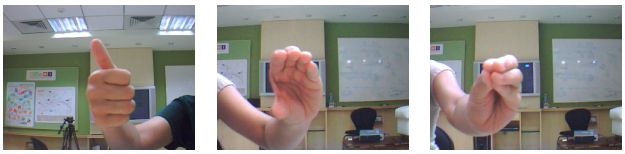
The classical appearance-based approach to the problem of vision-based hand pose estimation was re-visited by the introduction of depth cameras by companies like 3DV Systems [1] and Canesta [2]. These cameras can generate relative depth maps at 30 fps without the need for calibration. These depth maps can be utilized in real-time to achieve faster and accurate pose recognition than some of the optimization-based and template-based approaches mentioned in the survey article [3].

In this work, we have explored various methods to recognize six signature hand poses derived from a hand gesture vocabulary which was inspired by the intuitiveness to interact with the computer for command and control. The ZCam camera from 3DV Systems used in our experiments generates an RGB image as well as the depth map (8 bit) of objects present in its view. The interaction distance of the user from the camera is approximately set to 1.5 m and is independent of the ambiance and the user. The basic hand poses and their nomenclature observed in our gesture vocabulary are shown in Figure 1.

Our goal is to detect these signature hand poses irrespective of their change in position, orientation or scale. This is a challenging task since the hand is highly deformable and inferring the hand shape from 2D image data can be severely under constrained. We have tried to achieve our goal by augmenting the 2D image data with depth information. A new volumetric shape descriptor was developed which was inspired by the 2D shape descriptor introduced by Belongie et al. [4]. This volumetric shape descriptor gives the depth embedding at various levels. We argue that this multi level depth embedding along with the shape description can help us classify poses better. We have also explored a compressed version of a volumetric shape descriptor by storing the localized average depth of a collection of pixels.



(a) Palm pose (b) Point pose (c) Lateral pose



(d) Thumb pose (e) Open pose (f) Close pose

Figure 1. Signature hand poses

Since the depth cameras are yet to be widely commercialized, only a few studies have been conducted so far using depth for hand pose recognition. The algorithm proposed in [5] models a hand pose as a combination of a set of basic finger poses and finger inter-relations. The algorithm requires user initialization, limits the rotation of the hand to ± 30 degrees and is heavily dependent on the signature finger poses and may fail to recognize the pose which cannot be represented by the combination. The algorithm suggested in [6] uses PCA aligned data points on the hand as features. The downside however is, they require an extensive synthetic training set and it is not very clear if the system can handle scale variation. Depth data has also been used in gesture recognition in [7], [8]. The problem addressed by each of the above algorithms are heavily constrained and do not tackle scale and rotation invariance successfully.

We present a bird's-eye view of various stages involved in our algorithm in Section II. Section III describes in detail the volumetric shape descriptors. Section IV discusses our results and we conclude in Section V.

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.760

II. Our Algorithm

We believe that there is a very high mutual information between the depth and the shape of the hand. In accordance with our belief and to show the merits of using depth information, we have conducted separate tests with 2D shape descriptor, compressed volumetric shape descriptor and finally the 3D volumetric shape descriptor as features and compared the results. The stages involved in our algorithm are described below.

A. Preprocessing

The depth image generated by the camera is scaled to 0-255. Otsu's thresholding algorithm is applied to the depth histogram to segment the hand from the rest of the image. After thresholding the image, the pixel co-ordinates (x, y) and the corresponding unscaled depth values (d) of the segmented hand region are extracted. Let $\mathbf{X}$ be the set of all the points extracted from each frame. Unscaled depth values are used to avoid the non-linear behavior of the depth map introduced by the camera. Figure 2 shows the ZCam camera, depth image of the Palm pose and its corresponding 3-dimensional view as a heat map. The heat-map like view show the points with the highest depth value (closest) as red and the lowest depth value (farthest) as blue.

(a) Zcam (b) Depth image (c) Heat-map

Figure 2. Camera and depth data

This point cloud is then centered by subtracting its mean. The centered data points ($\overline{\mathbf{X}}$) are transformed to the PCA space which results in rotation invariance. Projection on to the PCA space can cause severe ambiguities in the 3D case where the directions are uncontrolled, though the transformed points are rotation invariant, without a consistent polarity, the invariance hardly makes sense in reality. We use a novel method which is detailed in Section III to identify the principal component along the depth direction and to uniformly align it across all the frames. The PCA transformed points are then ported into cylindrical co-ordinate system (r, φ, z) for the feature extraction.

B. Features Extraction

Experiments were carried out to analyze the performance of various shape descriptors. The 2D shape descriptor, compressed 3D shape descriptor and 3D volumetric shape descriptor are discussed in Section III. These features were averaged over 3 frames to feed more information into the classifier. This idea was inspired by the concept of structure-from-motion. It counteracts the problem faced due to occlusion in a few poses. For example, the lateral pose suffers from extreme occlusion when the hand is held perpendicular to the view of the camera.

C. Training

We collected 10 - 15s of video for each pose with the hand held at various angles and distances from the camera, numbering to around 400 frames per pose to train the classifiers. The training samples were gathered in both sitting and standing poses to make the detection view-invariant and to register the possible tilt in the data. We chose to use Support Vector Machines (SVM) for training the hand model. The model parameters were generated by cross validation procedure over the training images. SVM was used to generate probabilistic estimates to obtain the confidence of classification. We have used the LIBSVM module developed by Chang et.al [9] in our implementation.

D. Evaluation

A new set of videos different from the training videos varying in scale and orientation were used for testing the model. Every frame of the video was thresholded, transformed into the PCA space. After feature extraction the learned SVM model classifies the data into one of the six possible poses with a probability greater than 0.8. A confidence of 0.8 was chosen empirically since the classification probability of SVM for a correctly classified frame was found to range between 0.85-0.99. A decision is made for every frame of the video. If the classification probability is below the threshold, the frame is displayed with no pose detection.

III. Feature Descriptors

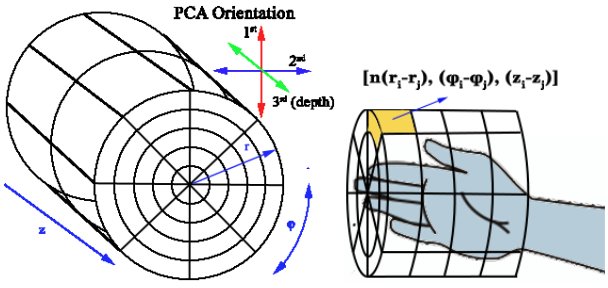
In this section we describe in detail the feature descriptors used for evaluation.

A. 2D Shape Descriptor

The 2D shape descriptors are generated by counting the spatial distribution of points on a set of concentric circles divided into sectors. We empirically chose to use 5 concentric circles divided into 8 sectors in our implementation. Only (x,y) co-ordinates are considered for PCA and cylindrical co-ordinate transformation. The scale invariance is achieved by dividing the range of r value into 5 equi-partitioned circles. Hence, irrespective of the size of the user's hand, the percentage of pixels in a sector patch will remain constant over a pose. The range of φ is always 0 - 2π and is divided equally into 8 parts (sectors). The percentage of data points in each bin is stored as its corresponding feature. The 5-fold cross validation accuracy obtained for the training set was 95.54%. A high accuracy on the training set suggests redundant samples.

B. Compressed 3D Shape Descriptor

Compressed 3D shape descriptor stores the average depth of the data points in every bin along with its 2D shape descriptor. The depth value varies non-linearly as the hand moves closer or away from the camera and has to be normalized. Without calibration, the relationship between the distance and the depth variation within the point cloud cannot be estimated. By experiments, we have found that the variation of depth range within a pose is very low. Hence, we re-scaled the depth values from 0 to $[max(d) - min(d)]$, where d is the set of all the depth values of the data points in $\mathbf{X}$. In other words, we assume that the $[max(d) - min(d)]$ (range of depth) value remains more or less constant across a given pose. For each of the bins in the 2D concentric histogram, we also store the average of the re-scaled depth values as a feature. We need to note that only the (x, y) co-ordinates are used for PCA transformation, while the depth dimension is untouched after scaling. The 5-fold cross validation accuracy on the training data was found to be 98.27%, which is consistent with our assumption that depth information increases accuracy. But this method might fail in a few cases where the basic assumption of constant depth variation does not hold good.



(a) 3D shape descriptor (b) Overlaid on hand

Figure 3. 3D shape descriptors

C. 3D Volumetric Shape Descriptors

This feature utilizes the depth information to reconstruct the hand in a 3D view. The 3D shape descriptor is an extended 2D shape descriptor and is cylindrical in shape. The cylinder is segmented along its axis to generate a number of discs as shown in Figure 3(a). The axis of the cylinder is always assumed to be aligned with the principal component along the depth direction. Hence, we make the assumption that hand pose can be distinguished by looking at the point distribution along the depth axis. Transforming the point cluster into the Principal Component Analysis (PCA) space and constraining the third principal component (one with the least variance) to correspond to depth dimension is a non-trivial problem. The principal components are always chosen in the decreasing order of the variance. Predominantly, depth tends to be along the least variant direction, but there might be a few instances where the axes are flipped leading to an unconstrained PCA. The directions predicted by the PCA might not conform to reality and distort the data and hence we propose a novel method of altering their directions to get a consistent representation.

Let $\mathbf{V}_{[3X3]} = (\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3})$ be the eigen vectors of the data matrix, $\overline{\mathbf{X}} = (\overline{x}, \overline{y}, \overline{d})$ and $\mathbf{V'}_{[2X2]} = (\mathbf{v'_a}, \mathbf{v'_b})$ be the eigen vectors corresponding to $\overline{\mathbf{X}} = (\overline{x}, \overline{y})$ ignoring the depth dimension. We propose to find two eigen vectors in $\mathbf{V}$ which correlate with the eigen vectors in $\mathbf{V'}$ and hence isolating the eigen vector or the principal component along the depth direction. The correlation matrix $\mathbf{S}$ is calculated by $\mathbf{S} = |\mathbf{V'}^T_{[2X2]}.\mathbf{V}_{[2X3]}|$, where d dimension in $\mathbf{V}$ is ignored.

$$\mathbf{S} = |\mathbf{V'}^T_{[2X2]}.\mathbf{V}_{[2X3]}| = \begin{bmatrix} S_{a1} & S_{a2} & S_{a3} \\ S_{b1} & S_{b2} & S_{b3} \end{bmatrix}$$

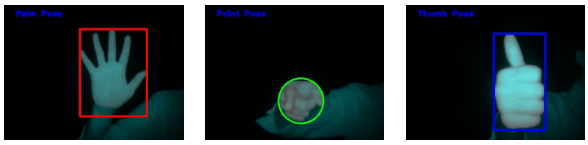
The first row of $\mathbf{S}$ corresponds to the correlation score of eigen vector $[\mathbf{v'_a}]$ with the eigen vectors $[\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}]$ hence the maximum of these correlation scores is used to find the correspondence i.e. if S_{a2} is maximum, $[\mathbf{v'_a}]$ in 2D corresponds to $[\mathbf{v_2}]$ in 3D. If the indicies turn out to be the same for both $[\mathbf{v'_a}]$ and $[\mathbf{v'_b}]$ then the absolute value of correlation score is compared for disambiguation. Hence, by finding these correspondences we are indirectly finding the eigen vector in the depth direction, which is assumed to be the third principal component always.

To ensure a consistent polarity, we verify if the direction from the minimum depth to the maximum depth remains the same even after the transformation into PCA space. If the direction is inverted (i.e. the minimum depth becomes maximum depth) we invert the direction of all the principal components to ensure consistency. The corrected data points are transformed into cylindrical co-ordinates and the percentage of data points in each of the 3D sector is stored as its corresponding feature, as shown in Figure 3(b). By varying the number of discs from 3 to 8, we have found the optimal number of discs to be 5. The 3D shape descriptors are very sparse and sparsity might lead to redundant dimensions and low classification accuracies. To enhance the precision and to simulate structure-from-motion in a crude way we averaged the bin values from the past 3 frames before classification. The 5-fold cross validation accuracy on the training data was found to be 99.79%.

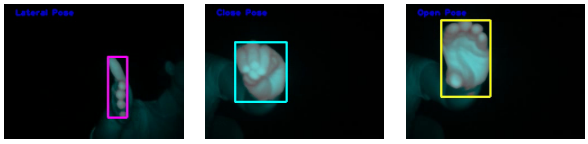
IV. Results

A new set of dynamic hand gesturing videos with approximately 200 frames per pose were used for testing and comparing the features detailed in Section III. The test results obtained for each of the descriptors is shown as a confusion matrix in Tables 1 - 3. The diagonal of the confusion matrix gives the absolute accuracy of the corresponding class. The real-time implementation of the algorithm developed on OpenCV takes an average of 60 ms (16 fps) on a 2 GHz Inter Core 2 Duo Processor to

classify each frame. Figure 4. shows some of the real-time recognition results.



(a) Palm pose (b) Point pose (c) Thumb pose



(d) Lateral pose (e) Close pose (f) Open pose

Figure 4. Pose recognition result windows

A. Discussion

The confusion matrix can be used to deduce the confidence of the classifier for each class and hence a definite confidence threshold can be set instead of having an empirical threshold. The comparisons of the results show that 3D shape descriptor performs significantly better than the compressed 3D and 2D shape descriptor features. The compressed 3D shape descriptor does not produce good results as expected due to the non-linear variation in the depth range. The results for Palm, Open and Close poses are significantly enhanced proving our hypothesis of distinguishability using depth. The confusion between the Palm, Open and in some cases the Lateral poses can be justified because of the similarity in their shape during the transition to other poses.

V. Conclusion

In this paper we have presented a novel feature descriptor and algorithm for recognizing hand poses dynamically. We have also developed a new method to automatically constrain the PCA directions in 3D data. The strength of our method mainly lies in recognizing hand poses which are rotation and scale invariant in real-time without the need of extensive training/template sets. Since our method relies only on the depth information, we no longer face any limitations caused by the lighting conditions. By comparing the results we find that our new algorithm for detecting hand poses using 3D volumetric shape descriptor performs significantly better as compared to the traditional 2D shape descriptors. However the shortcomings of the algorithm, mainly lies in the assumption that the Z-direction (depth) is always discriminative which might not be true in some cases. Also, the algorithm cannot handle extreme occlusions. Our further study incorporates expanding our signature pose library and using structure-from-motion techniques to reconstruct the hand pose in real-time.

References

[1] 3DV Systems Ltd., "http://www.3dvsystems.com/."

[2] Canesta Inc., "http://www.canesta.com/."

[3] A. Erol, G. Bebis, M. Nicolescu, R. Boyle, and A. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 52–73, 2007.

[4] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 509–522, 2001.

[5] Z. Mo and U. Neumann, "Real-time hand pose recognition using low-resolution depth images," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1499–1505, 2006.

[6] S. Malassiotis and M. G. Strintzis, "Real-time hand posture recognition using range data," *Image Vision Comput.*, vol. 26, no. 7, pp. 1027–1037, 2008.

[7] X. Liu and K. Fujimura, "Hand gesture recognition using depth data," *IEEE International Conference on Automatic Face and Gesture Recognition*, vol. 0, p. 529, 2004.

[8] R. Muñoz-Salinas, R. Medina-Carnicer, F. Madrid-Cuevas, and A. Carmona-Poyato, "Depth silhouettes for gesture recognition," *Pattern Recognition Letters*, vol. 29, no. 3, pp. 319–329, February 2008.

[9] C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," 2001.

Table I
CONFUSION MATRIX FOR 3D SHAPE DESCRIPTOR

Class	Palm	Point	Thumb	Lateral	Close	Open
Palm	**0.9188**	0.022	0	0.0258	0.0185	0.0148
Point	0	**0.9524**	0.0280	0	0.0112	0.0084
Thumb	0.0102	0.0408	**0.8112**	0.1378	0	0
Lateral	0.0196	0.0078	0.0431	**0.9020**	0.0275	0
Close	0.0240	0.0080	0.1280	0	**0.8320**	0.0080
Open	0.0561	0.0093	0	0	0.0935	**0.8411**

Table II
CONFUSION MATRIX FOR COMPRESSED 3D SHAPE DESCRIPTOR

Class	Palm	Point	Thumb	Lateral	Close	Open
Palm	**0.7528**	0	0.1327	0	0	0
Point	0.0221	**0.9552**	0.0357	0	0.0080	0.0187
Thumb	0.0111	0.0224	**0.7602**	0.1412	0	0
Lateral	0	0.0056	0.0663	**0.9294**	0.0080	0.0187
Close	0.0443	0.0476	0.0051	0	**0.6960**	0.0748
Open	0.0221	0.0140	0.0102	0	0.0960	**0.7664**

Table III
CONFUSION MATRIX FOR 2D SHAPE DESCRIPTOR

Class	Palm	Point	Thumb	Lateral	Close	Open
Palm	**0.6691**	0.0706	0	0.0223	0.0446	0.1933
Point	0	**0.9410**	0.0112	0	0.0028	0.0449
Thumb	0	0.0205	**0.7949**	0.1846	0	0
Lateral	0.0433	0.0118	0.0866	**0.8465**	0.0039	0.0079
Close	0	0.0887	0	0	**0.6371**	0.274
Open	0.0093	0.1495	0.0093	0	0.1523	**0.6794**