Three-Dimensional Model-based Object Recognition and Pose Estimation using Probabilistic Principal Surfaces

Kui-yu Chang and Joydeep Ghosh
Department of Electrical and Computer Engineering
University of Texas
Austin TX 78712
U.S.A.

ABSTRACT

A novel scheme using spherical manifolds is proposed for the simultaneous classification and pose estimation of 3-D objects from 2-D images. The spherical manifold imposes a local topological constraint on samples that are close to each other, while maintaining a global structure. Each node on the spherical manifold also corresponds nicely to a pose on a viewing sphere with 2 degrees of freedom. The proposed system is applied to aircraft classification and pose estimation.

Keywords: pose estimation, 3-D object, 2-D pose image, probabilistic principal surface, aircraft classification, manifold, spherical, image silhouette

1. INTRODUCTION

The human vision system is inherently two-dimensional (2-D), but is exceptionally adept at recognizing 3-D objects. This holds true, to a somewhat lesser extent, even in situations where depth and shading information are absent (e.g. recognizing 3-D objects based on their 2-D silhouettes), indicating that humans make use of implicit 3-D information. In fact, it has been shown that humans outperform template-based classifiers in recognizing novel (unseen) 2-D projections of 3-D objects,[1] when both are trained on the same set of 2-D pose images. Thus it is within reason to expect a pattern recognizer operating only on 2-D silhouettes, with suitably incorporated 3-D cues/hints, to perform better then one without hints.

Suppose a 3-D object is only allowed two degrees of freedom for rotation (ignoring rotation about the viewing axis, i.e. 2-D image plane), corresponding to the elevation and rotation (azimuth) angles, respectively, then each pose can be characterized by a 3-D vector $\mathbf{x}$ on the viewing sphere, where $\|\mathbf{x}\| = 1$. Each 2-D pose image can be thought of as being generated by a camera aimed towards the center of the sphere and travelling along the longitudes and latitudes. The usual way in which humans learn to recognize a 3-D object, that is, observing it from a number of distinct angles, can be seen as selecting strategic locations on the spherical viewing surface. Four such locations are shown in figure 1.

Taking cues from shading, lighting, and most importantly a relatively intricate knowledge of 3-D objects, a few distinct views usually suffice for humans to "get the whole picture". On the contrary, a nearest neighbor classifier not utilizing any high-level 3-D structural information needs to "see" all possible views of the object in order to recognize it and provide reliable pose estimates. This approach does not make use of 3-D pose information and is also sensitive to noise and small deviations from the templates.

In this paper, a general pose estimation and classification framework using spherical manifolds that possess none of the aforementioned limitations, is proposed. Under this framework, a spherical manifold is iteratively fitted to all 2-D pose images of an object, and a smooth nonlinear mapping from image poses to image feature vectors is obtained for each object. Given a test image feature vector, it is classified as the class of the nearest spherical manifold in feature space, and its projection onto that manifold gives the pose estimate. Figure 2 shows an example of a spherical manifold in 3-D feature space.

Further author information: (Send correspondence to Kui-yu Chang)
Kui-yu Chang: E-mail: kuiyu@lans.ece.utexas.edu; WWW:http://lans.ece.utexas.edu/~kuiyu
Joydeep Ghosh: E-mail: ghosh@lans.ece.utexas.edu; WWW:http://lans.ece.utexas.edu/~ghosh

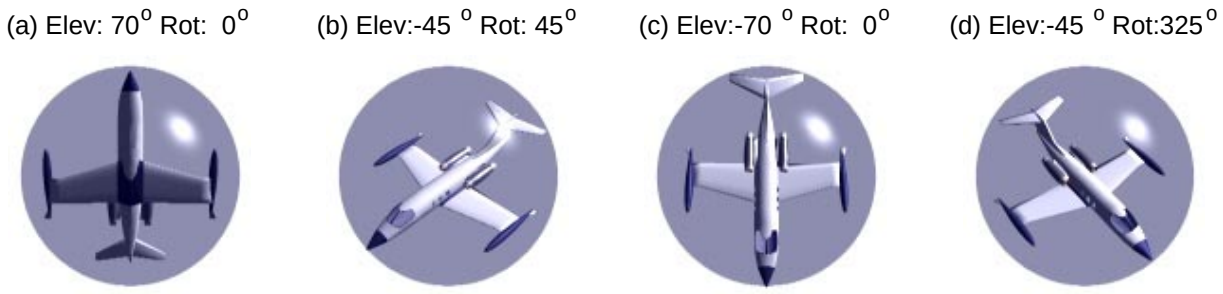

Figure 1. (a)–(d): Four sample poses of an object, shown encased within their respective viewing spheres. Poses are generated by traversing along the latitudes (rotation) and longitudes (elevation) of the sphere. By varying the elevation angle (Elev) from $[-90°, 90°]$, and the rotation angle (Rot) from $[0°, 360°)$, all possible unique views of the object may be generated. Rotations about the 2-D image plane are ignored.

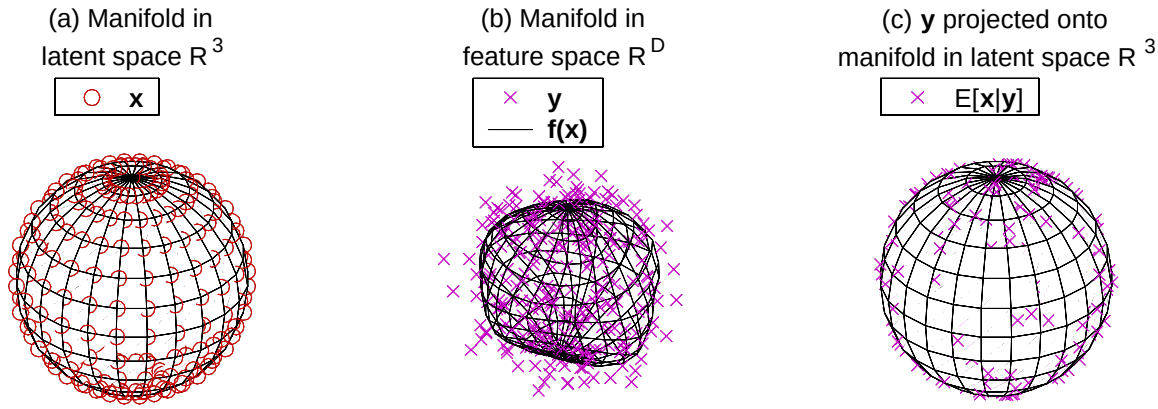

Figure 2. (a) The spherical manifold in $\mathbb{R}^3$ latent space. Each node $\mathbf{x}$ corresponds to a viewing position. (b) The spherical manifold in $\mathbb{R}^D$ feature space for $D = 3$ (usually $D \gg 3$). Each manifold node $\mathbf{f}(\mathbf{x})$ in feature space is related to $\mathbf{x}$ in latent space via a generalized linear transformation. Notice how the manifold approximates the data ($\mathbf{y}$), while maintaining the spherical topology. (c) Projection of data points $\mathbf{y}$ onto the latent spherical manifold. The projection of a test point onto the spherical manifold gives its pose estimate.

2. PREVIOUS WORK

One-dimensional (1-D) manifolds embedded in feature space have previously been used for pose estimation and object classification on simulated military vehicle images,[2] achieving better generalization (lower test classification error) compared to the multi-layer perceptron (MLP) neural network, the nearest neighbor classifier, and the radial basis function network.[3] However, that approach was restricted to classifying objects with a single degree of freedom. Also, the 1-D manifold used is sensitive to noise as it simply interpolates all data points without imposing any smoothness constraint. Somewhat related is the use of auto-associative MLP neural networks for embedding pose feature vectors onto a low-D manifold for classification.[4] Good classification results were reported for simple 3-D wire-frame objects (using 6 vertices as features), but the system cannot be used for pose estimation due to the lack of an explicit latent manifold model. An earlier effort on recognizing 3-D objects (but not their poses) from silhouette sequences using ART2 and aspect graphs, is described in the paper by Seibert and Waxman,[5] with further developments given in the paper by Grossberg and Bradski.[6]

Strong empirical evidence reported by Liu and Kersten[1] indicates that human object recognition depends on familiarity with the 2-D pose images of an object. Another key result is that classifiers restricted to rotations in the image plane of independent 2-D templates could not account for human performance in discriminating novel object views, implying the need for 3-D information. Image manifolds have also been frequently used in the past for other

applications such as face,[7] gesture[8] and handwriting recognition.[9] Most importantly, it was shown empirically that image manifolds may have large curvatures,[7] and therefore linear interpolation between two related 2-D pose images in pixel feature space may not result in a satisfactory image (as the interpolated image does not lie on the manifold). Much better results can be obtained if the interpolation is performed along the manifold.[8,7]

3. PROBLEM FORMULATION

Suppose there are C classes of 3-D objects, and each object has N_c training pose images. Let $\mathbf{y} \in \mathbb{R}^D$ denote the feature vector corresponding to a pose image and $\mathbf{x} \in \mathbb{R}^3$ its associated pose (co-ordinate on the viewing sphere), then the triplet $(\mathbf{y}, \mathbf{x}, c)$ represents a pose image of class c with pose $\mathbf{x}$. A spherical manifold $\mathbf{f}^c(\mathbf{x})$ is then fitted to all pose feature vectors $\mathbf{y}$ of a particular class c, independent of other classes. The feature vector $\mathbf{y}$ is extracted from each raw pose image by the following process:

1. The pose image is thresholded to yield a binary image.
2. The binary image is normalized for scale and translation invariance.
3. Zernike moments,[10] which are rotation invariant, are computed from the normalized image to form a feature vector.

The extracted feature vector $\mathbf{y}$ should be invariant with respect to image scale, translation and rotation. Rotation invariance is especially important in this context because it corresponds to removing the third degree of freedom from the viewing sphere, i.e. at each position on the viewing sphere, the hypothetical camera is allowed to rotate about its image plane without affecting the extracted feature vector $\mathbf{y}$. Scale invariance is somewhat tricky for aircraft images, and is described in detail next.

3.1. Image Pre-processing

The zero-th order geometric moment,[11] i.e. pixel area for a binary image, is commonly used to scale image objects to a uniform size, so as to achieve scale and translation invariance. However, this measure is not appropriate for images that vary significantly in shape skew, as in the case of aircraft poses. For example, the top-down view of the aircraft in figure 3(a) occupies significantly more screen real-estate compared to its frontal view in figure 3(e). If the zero-th order geometric moment (area) is used for scaling here, one of the following two results may occur: (1) a small-area object is magnified beyond the image boundaries and gets clipped, as shown in figure 3(f), or, (2) important details are lost from reducing a object too much, as exemplified in figure 3(c). Moreover, all available training pose images will have to be evaluated in order to select a suitable moment threshold, which may be problematic for unseen test pose images. In general, the situation gets worst with an increasing number of highly skewed aircraft pose images. Therefore for the pose estimation problem, it is better to achieve scale and translation invariance with respect to a fixed-size bounding shape. The proposed scheme is described as follows:

1. Compute the geometric centroid of object (unscaled image).
2. Perimeter-bound the object with the smallest bounding disc *bshape*, centered at the image centroid.
3. Scale the bounding *bshape* to the desired size/resolution (e.g. 128×128) via linear interpolation, thereby scaling the object contained within.
4. Create a new rectangular image out of the bounding box of the scaled *bshape*, in this case the centroid of the object will correspond to the center of the image.

Translation invariance is obtained by centering the bounding shape *bshape* at the object's centroid. Scale invariance is achieved via scaling the *bshape* to a constant resolution, say 128×128. The requirement that both the object centroid and *bshape* corresponds to the image center facilitates radial feature extraction; feature extraction algorithms such as Zernike moments consider only the portion of image lying within the unit circle. Figure 3 illustrates the advantages of this method compared to the geometric moment based normalization method on two typical aircraft poses.

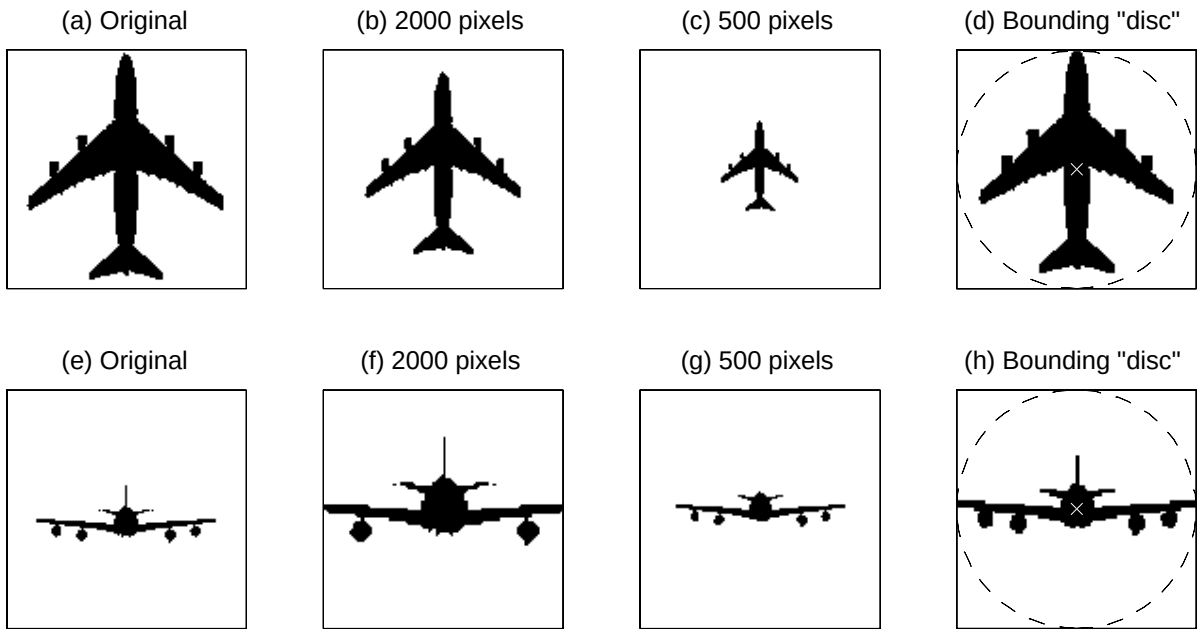

Figure 3. (a)(e) Original images, (b)(f) scaled to 2000 pixels, (c)(g) scaled to 500 pixels (d)(h) scaled via bounding disc method. Notice the clipping in (f), and the lost of details in (c). This example illustrates the difficulties encountered when pose images are made scale-invariant through the classical zeroth-order moment normalization method (constant pixel count). The situation gets worse when more pose images are taken into consideration. On the contrary, the bounding disc scaling method ensures a minimal loss of details, while maintaining translation and scale invariance (rotation invariance is also facilitated by making the centroid of the scaled object corresponds to the image center, as marked by the crosses in (d) and (h)).

3.2. Zernike Moments

After pre-processing, Zernike moments[10] are extracted from each image to form the feature vector $\mathbf{y}$. Zernike moments are the projection of the image function onto an orthogonal set of complex polynomials over the interior of the unit circle. It has been shown to be superior to other types of moment invariants for pattern classification tasks.[12] The Zernike moment of order a with repetition b for a discrete image function $I(c_1, c_2)$ is approximated as

$$Z_{ab} = \frac{a+1}{\pi} \sum_{c_1} \sum_{c_2} I(c_1, c_2) V_{ab}^{*}(\rho, \theta), \qquad c_1^2 + c_2^2 \leq 1$$

where the complex polynomial V_{ab} is given in polar co-ordinates as

$$V_{ab}(\rho, \theta) = \left[\sum_{s=0}^{0.5(a-|b|)} (-1)^s \frac{(a-s)!}{s! \left(\frac{a+|b|}{2} - s\right)! \left(\frac{a-|b|}{2} - s\right)!} \rho^{a-2s} \right] e^{jb\theta}$$

and

a : Order (non-negative integer)
b : Repetition (integer, subject to constraints $(a - |b|)$ is even and $|a| \leq b$)
ρ : Length of vector from image center to pixel co-ordinate (c_1, c_2)
θ : Angle between vector ρ and the horizontal axis.

The Zernike moments' magnitudes are invariant to image rotation, and thus can be used as rotation-invariant features. In fact, the Zernike moment Z'_{ab} of a rotated (by θ radians) image is simply related to that (Z_{ab}) of the unrotated

image by a phase change proportional to the degree of rotation, i.e.

$$Z'_{ab} = Z_{ab} e^{-jb\theta}. \tag{1}$$

This property is very useful for pose compensation, described later. Note that since $Z_{a,-b} = Z^*_{ab}$, only moments for $b \geq 0$ need to be considered. In addition, the first Zernike moment Z_{00} represents the pixel mass/count, which varies with poses (with respect to the scaling scheme used in this paper), and therefore is useful. The second Zernike moment Z_{11} is almost zero as it is proportional to the horizontal and vertical first-order geometric moments, which are both zero due to translation normalization, and are therefore not used. In this paper, up to 49 (absolute) moments are used.

4. SPHERICAL MANIFOLDS

The spherical manifold is comprised of M nodes $\{\mathbf{x}_m\}_{m=1}^{M}$ evenly distributed on the surface of a sphere as shown in figure 2(a). The next section briefly describes the probabilistic principal surface model which is used to construct the spherical manifold.

4.1. Probabilistic Principal Surfaces

Principal surfaces (curves)[13] are nonlinear generalizations of principal subspaces (components) that formalizes the notion of a low-D manifold passing through the 'middle' of a dataset in high-D space. The probabilistic principal surface (PPS),[14] a generalization of the generative topographic mapping,[15,16] is a parametric approximation of principal surfaces. The PPS manifold is comprised of M nodes $\{\mathbf{x}_m\}_{m=1}^{M}$ arranged typically on a uniform topological grid in latent (low-D) space $\mathbb{R}^Q$. The topology is consistently enforced via a generalized linear mapping from each latent node $\mathbf{x}_m$ in $\mathbb{R}^Q$ to it's corresponding data node $\mathbf{f}(\mathbf{x}_m)$ in data (high-D) space $\mathbb{R}^D$ (D is the data dimensionality),

$$\mathbf{f}(\mathbf{x}_m) = \mathbf{W}\boldsymbol{\phi}(\mathbf{x}_m)$$

where $\mathbf{W}$ is a $D \times L$ real matrix and

$$\boldsymbol{\phi}(\mathbf{x}_m) = \left[\begin{array}{ccc} \phi_1(\mathbf{x}_m) & \cdots & \phi_L(\mathbf{x}_m) \end{array} \right]^T,$$

is the vector containing L latent basis functions $\phi_l(\mathbf{x}) : \mathbb{R}^Q \rightarrow \mathbb{R}$, $l = 1, \ldots, L$. The basis functions $\phi_l(\mathbf{x})$ are usually isotropic Gaussians with constant widths. Each data node $\mathbf{f}(\mathbf{x}_m)$ actually corresponds to the mean of a Gaussian probability distribution with noise covariance parameter $\boldsymbol{\Sigma}(\alpha, \mathbf{x}_m)$ where α $(0 < \alpha < D/Q)$ denotes the amount of clamping in the tangential manifold direction. Note that the GTM is obtained for $\alpha = 1$, and $\alpha < 1$ has been shown to yield a better manifold in terms of reconstruction error.[14] The PPS is iteratively computed using a maximum likelihood optimization procedure. Figure 4 shows a 1-D PPS. Clearly, the spherical manifold can be trivially constructed using a PPS with latent nodes $\{\mathbf{x}_m\}_{m=1}^{M}$ arranged regularly on the surface of a sphere in $\mathbb{R}^3$.

4.2. Initialization

In order to achieve a one-to-one mapping between each pose image and manifold node, the number of nodes M is set equal to the number of samples, and the spherical manifold is initialized such that $\mathbf{W}$ approximately maps each latent node at position $\mathbf{x}$ to a pose image of pose $\mathbf{x}$. This can be done by solving for $\mathbf{W}$ in the least squares equation

$$\left[\begin{array}{ccc} \mathbf{y}_1 & \cdots & \mathbf{y}_M \end{array} \right] = \mathbf{W} \left[\begin{array}{ccc} \boldsymbol{\phi}(\mathbf{x}_1) & \cdots & \boldsymbol{\phi}(\mathbf{x}_M) \end{array} \right]$$

where the pose of image feature vector $\mathbf{y}_m$ is $\mathbf{x}_m$ for $m = 1, \ldots, M$.

4.3. Classification and Pose Estimation

Given a test image feature vector $\mathbf{y}$, it is first projected onto all C manifolds $\{\mathbf{f}^c\}_{c=1}^{C}$, and is assigned the class label of the nearest manifold. The co-ordinate of its projection onto that manifold gives a first-cut pose estimate $\mathbf{x}_p$. The projection can be performed probabilistically,[14] or simply using a nearest neighbor approach. In experiments, it has been found that probabilistic projections give better pose estimates. Note that due to the use of rotation invariant features, a rotational displacement may exist between the images corresponding to $\mathbf{y}$ and $\mathbf{f}(\mathbf{x}_p)$. In this case, the phase difference between the complex versions of $\mathbf{y}$ and $\mathbf{f}(\mathbf{x}_p)$ may be used to compensate for the rotation according to equation 1. Thus, although only the magnitudes of Zernike moments are used for classification and manifold construction, their original complex versions must also be stored and made available if pose compensation is desired.

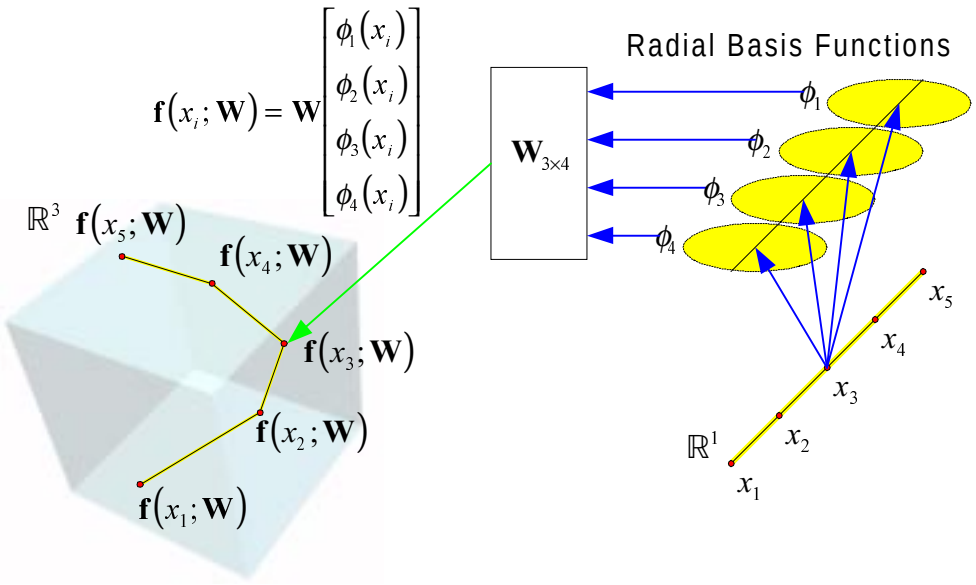

Figure 4. A 1-D PPS in $\mathbb{R}^3$ with 5 nodes and 4 latent bases.

5. EXPERIMENTS

5.1. LANS Aircraft Dataset

In practice it is difficult to obtain all possible pose images for any arbitrary aircraft, unless a computer 3-D model of the aircraft is available. Therefore in this paper, computer 3-D models were used to generate the training pose images of 10 aircrafts. A total of 684 pose images at 128×128 pixel resolution were generated/rendered for each aircraft such that the training poses range from

$$(elev, rot) : \begin{cases} ele = \{-90, -80, \ldots, 80, 90\} \\ rot = \{0, 10, \ldots, 350\} \end{cases} .$$

Note that only one training pose image is needed at the north and south poles ($elev = -90, 90$); since rotation at the poles is equivalent to rotation about the image plane, and the feature extraction method used here is rotation invariant. Similarly, another 648 pose images corresponding to poses

$$(elev, rot) : \begin{cases} ele = \{-85, -75, \ldots, 75, 85\} \\ rot = \{0, 10, \ldots, 350\} \end{cases} ,$$

were generated for testing purposes.

A 614-node spherical manifold with 83 latent bases (including bias) was computed for each aircraft using 29 Zernike moments and a clamping factor $\alpha = 1$. Tables 1 and 2 show the confusion matrix for the training and test set, respectively. Some interesting characteristics can be inferred from the test result by cross-referencing to the corresponding pictures in figure 5. First, the similarity between class 2 (b707) and 3 (b747) is symmetrical, as both are Boeing jets. Likewise, class 4 (cessna402) and 5 (dc3), which are both propeller driven, have a certain degree of symmetrical overlap. Last, class 7 (f15) is sometimes misclassified as class 8 (f18) and 10 (mig29m), which again is expected since all three aircrafts looked very alike (twin vertical rear wing stabilizers, twin undermounted jet engines). Moreoever, class 8 (f18), with its more compact fuselage, has a relatively high classification accuracy. On the other hand, class 10 (mig29m) is frequently mistaken for class 7 (f15), as both have wider bodies compared to class 8 (f18). It should be noted that the high classification rate obtained for class 7,8 and 10 is quite remarkable, considering the high degree of similarity between the three types of aircrafts and that only image silhouettes were used.

Table 3 shows the effect of using a larger number of Zernike moments and different clamping values (α) on the test classification rate and pose estimation accuracy. It can be seen that using a larger D (more Zernike moments)

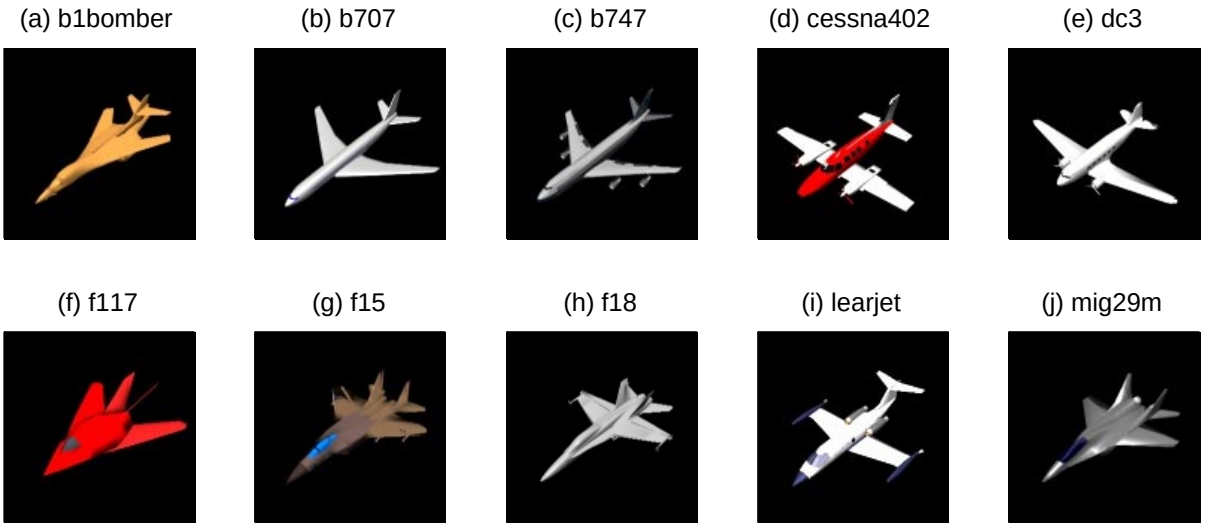

Figure 5. (a)–(j) The 10 computer rendered 3-D models used to generate the image poses. The models are shown at $-40°$ elevation and $40°$ rotation. A total of 614 pose images are generated for each aircraft.

True\Classified	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1 b1bomber	676					3	1		1	3
2 b707	1	663	11		8					1
3 b747		11	663	4	5			1	1	
4 cessna402		5		674	2	1				1
5 dc3	2	1	2	3	671	3	1		3	
6 f117	1	1				677	1			3
7 f15		1				1	678			3
8 f18		1	1		4			678		
9 learjet		3	3	1	1	3	1	1	670	1
10 mig29m		1		1	1		6	2		673
	Training Classification Error = 1.7 % (117 errors)									

Table 1. Training Classification Confusion Matrix.

resulted in noticeably lower test classification error, but only yielded marginal improvements in pose estimation accuracy. Setting $\alpha = 0.75$ also improved slightly both the classification and pose estimation accuracy. Thus it can be concluded that $\alpha = 1$ gives reasonable performance with the least computational complexity (setting $\alpha \neq 1$ requires extra computational effort).

Figure 6 plots the percentage of training and test samples with poses estimated within a given accuracy. The error (in degrees) is measured as the angle between the true pose vector and that of the estimated pose vector on the viewing sphere. From the figure, it can be seen that approximately 55% of the test samples can be estimated within $20°$ accuracy, whereas almost 70% of the samples can be accurately estimated within $40°$ accuracy. This is satisfactory considering the fact that the classifier was trained at $10°$ resolution, and that it uses only rotation invariant features extracted from object silhouettes. The average elevation estimation error is less than $20°$ whereas the average rotation estimation error is around $100°$. Thus, most of the pose estimation errors occurred in the rotation angle, and can be attributed to the use of rotation invariant features. The compensation method mentioned previously may be used to obtain further improvement in rotation angle estimation accuracy. It should be noted that simple compensation is only possible for large elevation values, i.e. $ele \simeq 90°$.

True\Classified	1	2	3	4	5	6	7	8	9	10
1 b1bomber	616			3	2	6	6	6	3	6
2 b707		583	34	5	10	3	1	1	4	7
3 b747		46	574	10	4	5	2	1	5	1
4 cessna402		13	5	562	31	3	5	15	8	6
5 dc3		11	6	30	581	4		2	6	8
6 f117	13	2	4		4	605	6	2	3	9
7 f15	5	1	1	2		8	571	28	4	28
8 f18	5	2	1	2	2		15	611		10
9 learjet	1	9	6	4	11	11	7	5	588	6
10 mig29m	10		1	3	1	8	31	11	1	582
	Test Classification Error = 9.4 % (607 errors)									

Table 2. Test Classification Confusion Matrix.

		Classification	Percentage (%) of pose estimates within given accuracy						
D	α	Error (%)	$< 10°$	$< 15°$	$< 20°$	$< 25°$	$< 30°$	$< 35°$	$< 40°$
29	0.5	9.69	32	48	55	60	63	66	68
	0.75	9.18	34	49	56	61	64	67	69
	1	9.37	33	49	56	60	63	66	68
39	0.5	6.85	34	50	57	61	65	67	69
	0.75	7.20	34	50	57	61	65	67	69
	1	7.08	34	50	57	61	64	66	69
49	1	5.93	36	51	57	62	65	67	70

Table 3. Test performance for different number of features (D) and clamping parameter values (α).

5.2. Real Aircraft Dataset

To evaluate how well the system performs on real-life data, a selection of images corresponding to the studied aircrafts were downloaded from the internet and fed into the trained classifier as inputs. Each image is segmented semi-automatically (using a combination of edge-detection, thresholding and other image processing techniques with human supervision) and normalized according to the scheme in section 3.1. Although the exact pose of each test image is unknown, the pose-estimate given by the spherical manifold classifier can be used to generate a pose-image from the 3-D model, which is then visually compared to the test image. Figures 7–13 show the classified images with their corresponding re-generated 3-D pose images.

6. CONCLUSION

A novel system using spherical manifolds for 3-D object classification and pose estimation from 2-D images is proposed and applied to the task of aircraft classification. The system was shown to generalize well on both unseen artificial and real images when trained only on a set of computer generated 2-D object silhouettes. Despite the existence of ambiguity pose images, the system was able to differentiate between very similar looking aircrafts (the sets: {f14, f18, mig29}, {b707, b747}, {cessna402, dc3}) most of the time. It was also capable of giving reliable pose estimates (within 40° accuracy) for almost 70% of the data, or within 20° for over 50% of the data, even when rotation invariant features were used. With pose compensation, the pose estimation accuracy can be improved further.

It is possible to perform image interpolation under the developed framework, either using Zernike moments or other invertible feature extraction methods such as wavelet transforms. If Zernike moments are used for this purposes, the amount of data storage requirement becomes quite significant as the same number of image basis as the number of features must be stored. More work is currently being done on coming up with a reliable compensation method, in addition to investigating other types of feature extraction schemes, and testing the system on more real photographs.

ACKNOWLEDGMENTS

This research was supported in part by Army Research contracts DAAG55-98-1-0230 and DAAD19-99-1-0012, and NSF grant ECS-9900353. The authors are also grateful to the following persons who provided computation and

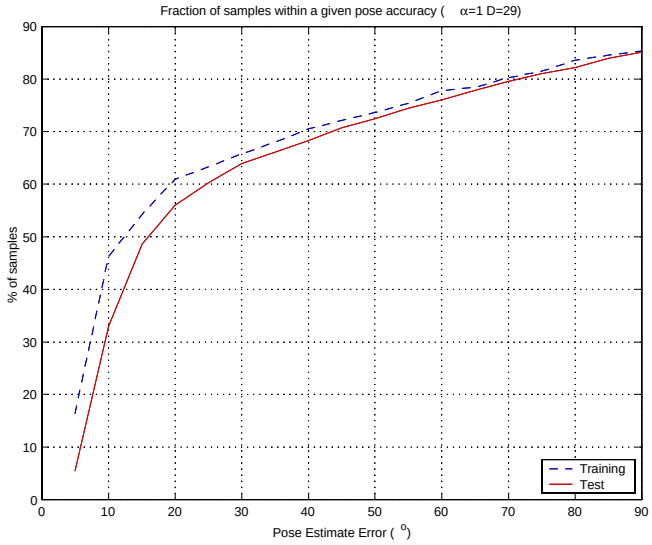

Figure 6. Plot of fraction of poses estimated within given accuracy.

networking resources while portions of this paper was being prepared during a trip to the Far East : Pao-kuo Chang, Pao-min Chang, Hai-Shing Chiang, Vui-Kwan Chong and Choong-Ching Lim.

REFERENCES

1. Z. Liu and D. Kersten, "2D observers for human 3D object recognition?," in *Advances in Neural Information Processing Systems*, M. I. Jordan, M. J. Kearns, and S. A. Solla, eds., vol. 10, pp. 829–835, 1997.
2. L. Neiberg and D. P. Casasent, "Feature space trajectory (FST) classifier neural network," in *Proc. SPIE*, vol. 2353, pp. 276–292, SPIE, 1994.
3. L. Neiberg and D. P. Casasent, "Feature space trajectory neural net classifier," in *Proc. SPIE*, vol. 2492, pp. 361–372, SPIE, 1995.
4. S. Suzuki and H. Ando, "Unsupervised classification of 3D objects from 2D views," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, eds., vol. 7, pp. 949–956, 1995.
5. M. Seibert and A. M. Waxman, "Learning and recognizing 3D objects from multiple views in a neural system," in *Neural Networks for Perception*, H. Wechsler, ed., vol. 1, pp. 426–444, Academic Press, Inc., 1991.
6. S. Grossberg and G. Bradski, "VIEWNET: A neural architecture for learning to recognize 3-D objects from multiple 2-D views," in *Proc. SPIE*, vol. 2353, pp. 266–275, SPIE, 1994.
7. H.-M. Lu, Y. Fainman, and R. Hecht-Nielsen, "Image manifolds," in *Proc. SPIE*, vol. 3307, pp. 52–63, SPIE, 1998.
8. C. Bregler and S. M. Omohundro, "Nonlinear image interpolation using manifold learning," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, eds., vol. 7, pp. 973–980, 1995.
9. G. E. Hinton, P. Dayan, and M. Revow, "Modeling the manifolds of images of handwritten digits," *IEEE Transactions on Neural Networks* **8**, pp. 65–74, Jan 1997.
10. A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**, pp. 489–497, May 1990.
11. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison Wesley, 1992.
12. C.-H. Teh and R. T. Chin, "On image analysis by the methods of moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**, pp. 496–513, Jul 1988.
13. T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American Statistical Association* **84**, pp. 502–516, Jun 1988.
14. K.-Y. Chang and J. Ghosh, "A unified model for probabilistic principal surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , p. (submitted 1999.10.29).

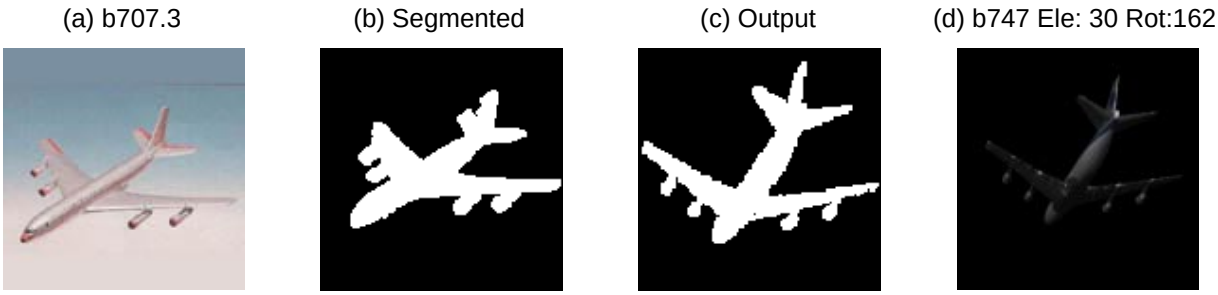

Figure 7. (a) input image, (b) normalized image, (c) image/pose from classifier output, (d) 3-D rendered image based on classifier output. This is a good example of the ambiguity caused by using image silhouettes; the silhouettes in (b) and (c) look alike, but actually correspond to planes pointing in opposite directions. The misclassification of (a) into a b747 is acceptable in this case, since it is difficult even for humans to make the distinction solely based on the information in (b).

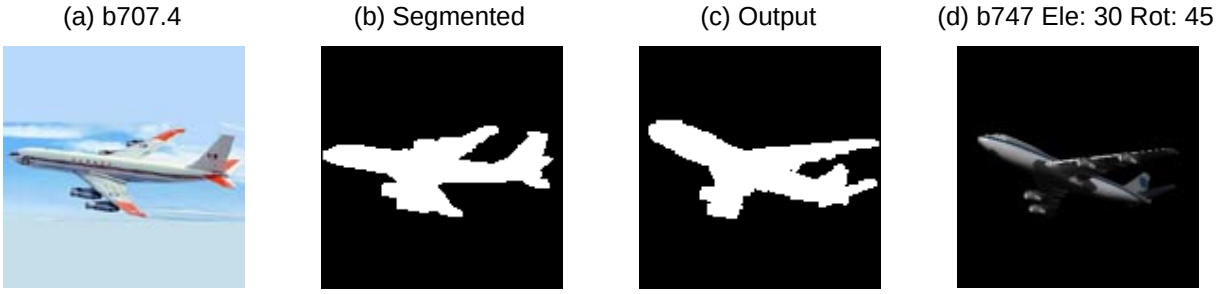

Figure 8. (a) input image, (b) normalized image, (c) image/pose from classifier output, (d) 3-D rendered image based on classifier output. This is another example highlighting the ambiguity of image silhouettes. The images in (b) and (c) are approximately related by an image rotation. Again, the classifier "thinks" that the image in (b) is a b747 instead of a b707.

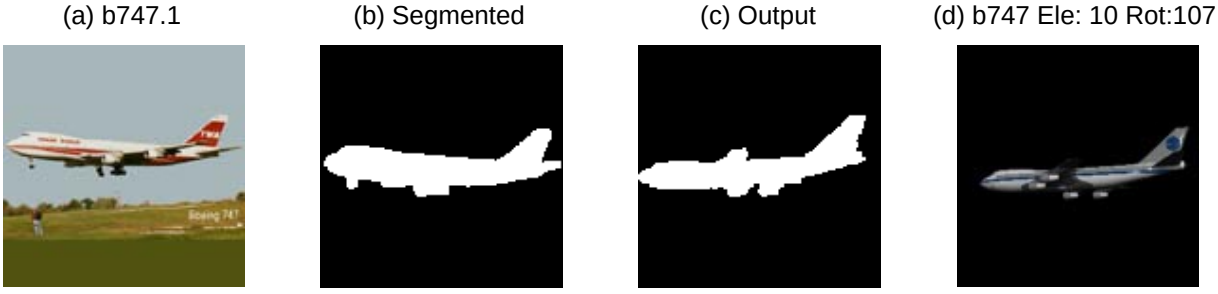

Figure 9. (a) input image, (b) normalized image, (c) image/pose from classifier output, (d) 3-D rendered image based on classifier output.

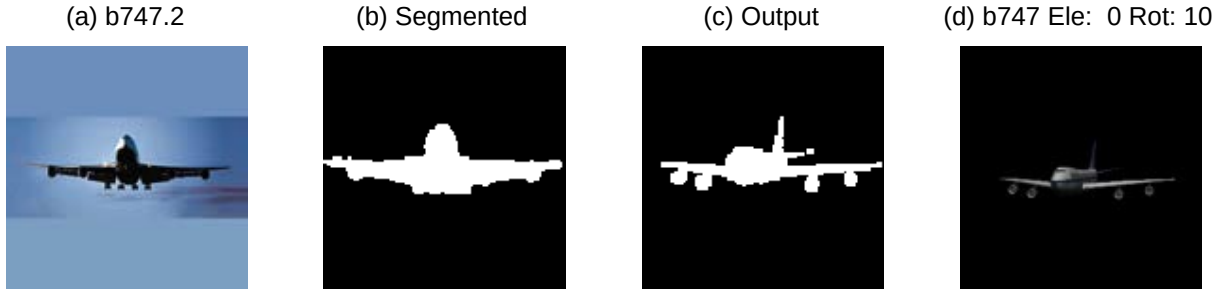

Figure 10. (a) input image, (b) normalized image, (c) image/pose from classifier output, (d) 3-D rendered image based on classifier output.

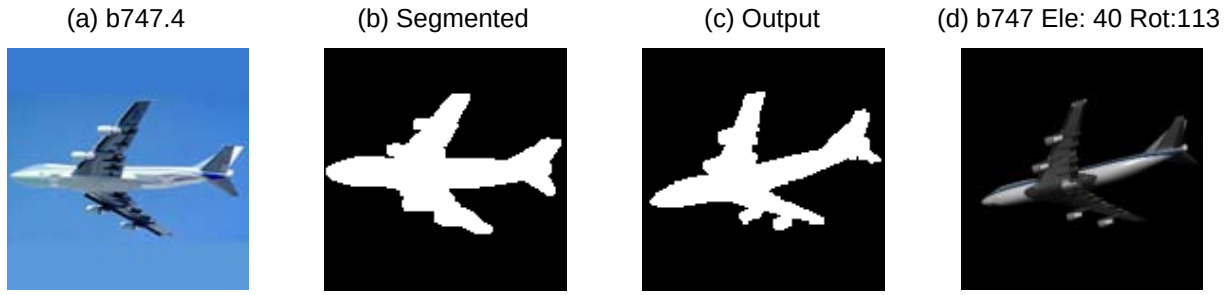

Figure 11. (a) input image, (b) normalized image, (c) image/pose from classifier output, (d) 3-D rendered image based on classifier output. The classifier output is near perfect, except for an image rotation (caused by using rotation-invariant features).

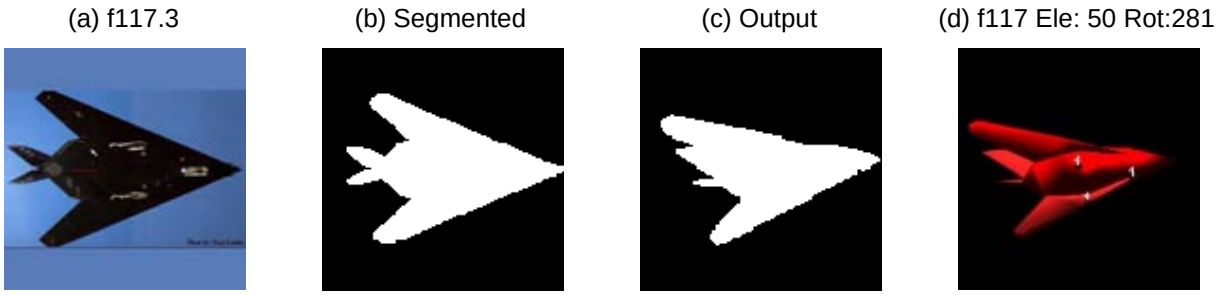

Figure 12. (a) input image, (b) normalized image, (c) image/pose from classifier output, (d) 3-D rendered image based on classifier output.

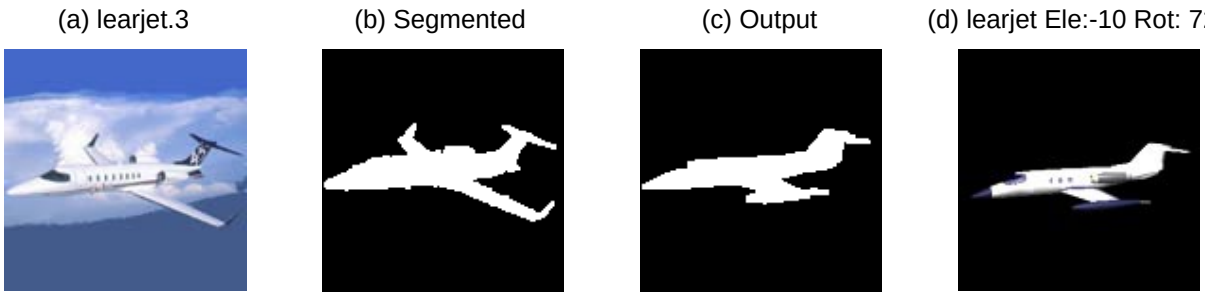

Figure 13. (a) input image, (b) normalized image, (c) image/pose from classifier output, (d) 3-D rendered image based on classifier output. Note that the test image in (a) represents a newer generation of the learjet, yet the system was able to classify and estimate its pose correctly.

15. C. M. Bishop, M. Svensén, and C. K. I. Williams, "GTM: The generative topographic mapping," *Neural Computation* **10**(1), pp. 215–235, 1998.
16. C. M. Bishop, M. Svensén, and C. K. I. Williams, "Developments of the generative topographic mapping," *Neurocomputing* **21**, pp. 203–224, 1998.