Mental Image Transformation and Matching using Surface Reconstruction Neural Networks

Jenq-Neng Hwang, Yen-Hao Tseng

Information Processing Laboratory
Department of Electrical Engineering, FT-10
University of Washington
Seattle, WA 98195

ABSTRACT

Invariant 2-D/3-D object recognition and motion estimation under detection/occlusion noise and/or partial object viewing are difficult pattern recognition tasks. On the other hand, the biological neural networks of human are extremely adept in these tasks. It has been suggested by the studies of experimental psychology that the task of matching rotated and scaled shapes by human is done by mentally rotating and scaling gradually one of the shapes into the orientation and size of the other and then testing for a match. Motivated by these studies, we present a novel and robust neural network solution for these tasks based on detected surface boundary data or range data. The method operates in two stages: The object is first parametrically represented by a surface reconstruction neural network (SRNN) trained by the boundary points sampled from the exemplar object. When later presented with boundary points sampled from the distorted object without point correspondence, this parametric representation allows the mismatch information back-propagate through the SRNN to gradually determine (align) the best similarity transform of the distorted object. The distance measure can then be computed in the reconstructed representation domain between the surface reconstructed exemplar object and the aligned distorted object. Applications to invariant 2-D target classification and 3-D object motion estimation using sparse range data collected from a single aspect view are presented.

1 Introduction

In 2-D/3-D pattern recognition applications, we usually encounter the following problem: given a set of boundary points (testing data) extracted from the boundary or the surface of a 2-D or 3-D object, which is distorted by a *similarity transform* (translation, rotation, and scaling), corrupted by random detection/occlusion noise, and/or suffered from partial data acquisition due to a limited number of aspect views, from one of several exemplar objects, we want to estimate the amount of translation $\mathbf{t}$, rotation θ, and scaling s of the testing object with respect to the corresponding exemplar object (absolute orientation problem or motion estimation [9, 16]), or to classify the distorted testing object (invariant pattern recognition) to one of the exemplar class. A useful and general solution to this problem can find applications in a wide variety of fields: e.g., design automation, manufacturing automation, terrain mapping, vehicle guidance, and intelligent robots, etc.

Most algorithms proposed so far for absolute orientation or motion estimation problems adopted the *point correspondences* assumption, where the size of testing data is the same as that of reference data, and each point in the testing data has been pre-matched with (corresponded to) one point in the exemplar data. This assumption results in a least squares problem, and can be solved by an iterative algorithm [11], or the singular value decomposition of a covariance matrix of the data [1, 24]. In reality, there are missing points and noise corruption in most pattern recognition tasks. Therefore, unequal sizes of the two data sets are expected and the correspondence of data points from two objects are thus difficult to determine. To solve these problems, some efforts have been tried. For example, by correlating the Fourier transforms, a representation independent of where the points are sampled, of the functions defined on the coordinates of the root points from two objects, it is possible to estimate the similarity transform parameters [16].

Applying neural networks techniques to invariant pattern recognition tasks is emerging due to the potential relationship between artificial neural networks and biological neural networks, which are extremely adept in invariant pattern recognition tasks. Existing techniques for establishing neural networks for invariant recognition can be grouped into 3 categories [2]: (1) The *invariance by structure* methods create network connections which force transformed version of the same input to have the same output. These methods require prohibitively large number of connections for most practical applications. (2) The *invariance by training* methods train a network with different examples (aspect views) of the same object. These methods require prohibitively large number of training samples of different translations, rotations, and scaling for the classifiers to generalize well. (3) The *invariant feature spaces* methods extract input features which are invariant to transformations. The performance of these methods highly depend on the preprocessing techniques in extracting the invariant features.

In 3-D invariant pattern recognition or motion estimation, one important issue is partial viewing of the object due to the limited number of aspect view angles. More specifically, only the visible surface (or surfaces) which constitutes part of surfaces of the whole object is available, and it is generally assumed that no information available beyond that. This issue further complicates the research for 3-D invariant pattern recognition and motion etimation. Some probabilistic approaches [4, 6] have been proposed to solve this limited-aspect data collection problem. In these approaches, the 3-D objects (e.g., the manufacturing objects) are approximated by a small number of patches of planes, cylinders, and spheres. Detection probabilities were computed based on the collected (range) data and transform parameters were estimated for each patch of data in a maximum likelihood manner. Another approach combining splines interpolation and depth estimation using Bayesian model techniques has also been introduced for estimating observer's motion [21]. All these methods suffer from *a priori* assumption about the standard building patches for objects as well as the form of the probability distribution to be used in the probabilistic parameter estimation.

This paper proposes a novel surface reconstruction neural network (SRNN) approach to overcome the shortcomings encountered in the existing methods discussed above. Our method can be supported by the experimental human psychology studies which show that *the task of matching rotated and scaled shapes is done by mentally rotating and scaling one of the shapes into the orientation and size of the other and then tested for a match.* Furthermore, it was also reported that the mental transformations of size and orientation appeared to be carried out by gradually alternating small steps of sizes and orientations [8, 19].

Our proposed SRNN approach consists of two stages of efforts. The object shape (as well as the whole 2-D/3-D image space) is first represented by a surface reconstruction neural network (SRNN) trained by the boundary points sampled from the exemplar object. The SRNN parametrically reconstructs (interpolates), in a least squares sense, the surface of the exemplar object as well as the profile of the whole space based on the available sparse boundary data. More specifically, every point (in continuous domain) in the 2-D/3-D space will be assigned a value. If the point falls on the reconstructed boundary, it is assigned a value "0". If a point is outside (or inside) of the reconstructed object, it has positive (or negative) value with the magnitude linearly proportional to the perpendicular distance of this point to the object boundary (constant slope profile). When later presented with those boundary points of the distorted object, this parametric SRNN representation allows easy accumulation of the mismatch information from all the distorted testing data. More importantly, the mismatch information can back-propagate through the SRNN to iteratively determine the best similarity transform of the distorted object (similar to the gradual mental transformation). The distance measure can then be computed in the reconstructed representation domain between the exemplar object and the aligned distorted object.

One point worthy of mentioning is that most conventional shape description using parametric functions (e.g., polynomials [23], splines [3], deformable functions [25, 20]) which specifically describe the shape of the objects without modeling the whole image space. Our proposed SRNN representation, on the other hand, gives a continuous description of the whole image space and the shape is implicitly defined as the subspace that gives $\phi(\mathrm{x}) = 0$.

The organization of this paper is as follows: In Section 2, we discuss the learning procedures involved in training an SRNN. Section 3 proposes our mechanism of estimating the similarity transform parameters by incorporating the trained SRNN through a mental image transform-like approach. Section 4 discusses the application of the SRNN approach to 2-D invariant target recognition. Section 5 presents the the application of the SRNN approach to 3-D motion estimation when the testing object is described by a set of sparse range data collected from one single aspect view angle. Concluding remarks are given in Section 6.

2 A Surface Reconstruction Neural Network Representation

Given a set of data points $\{\mathrm{x}\}$ (in $\mathcal{R}^2$ or $\mathcal{R}^3$) that lie on or close to the 2-D curve or 3-D surface of an object, the goal of surface reconstruction is to reconstruct the 2-D curve or 3-D surface of the object, expressed by a smooth and continuous function $\phi(\mathrm{x})$ such that the training data set is given as sample root points ($\mathbf{B} = \{\mathrm{x}^{(i)},\ i = 1, \ldots, n\}$) of $\phi(\mathrm{x})$, i.e., $\phi(\mathrm{x}) = 0,\ \mathrm{x} \in \mathbf{B}$

2.1 A Standard Two-layer Perceptron Neural Network

A standard 2-layer (one hidden layer) perceptron neural network, which has been proved to be a universal functional approximator [10], is used for surface reconstruction (see Figure 1). In this SRNN, the activation values of the k^{th} neuron of the l^{th} layer is denoted as $a_k(l)$, $l = 0, 1, 2$, where $l = 0$ specifies the input layer (consists of 2 or 3 input neurons with values equal to x); $l = 1$ specifies the hidden layer (consists of H hidden neurons with commonly used sigmoid

Figure 1: An SRNN based on a standard multilayer perceptron.

nonlinearity, $\frac{1}{1+e^{-u}}$); and $l = 2$ specifies the output layer (consists of a single output neuron with slightly modified sigmoid nonlinearity, $\frac{2}{1+e^{-u}} - 1$). The interconnection weight linking the k^{th} neuron of the l^{th} layer with the j^{th} neuron at the $l-1^{th}$ layer is denoted as $w_{kj}(l)$ (with $w_{k0}(l)$ denoting the offset of the neuron).

To train this 2-layer neural network to learn the unknown function $\phi(\mathrm{x})$ is not so straightforward, since the desired teacher value for the training point is $t^{(i)} = 0,\ \forall i$. If the standard cost function (i.e., the lease-squares error) is minimized [18]:

$$E \equiv \frac{1}{2} \sum_{\mathbf{x} \in \mathbf{B}} [t - \phi(\mathrm{x})]^2 = \frac{1}{2} \sum_{\mathbf{x} \in \mathbf{B}} \phi^2(\mathrm{x}) \tag{1}$$

then the trained network will yield a trivial solution $\phi(\mathrm{x}) = 0$. Therefore, there is need for a more proper cost function.

2.2 A New Cost Function to be Minimized

The new cost function to be minimized should reflect the fact that $\phi(\mathrm{x}) = 0$ for boundary points and the implied $\phi(\mathrm{x}) \neq 0$ for nonboundary points. One intuitive way is to assign $\phi(\mathrm{x})$ to be positive on one side (e.g., exterior of the object) of the contour and negative on the other (e.g., interior of the object). This requires an additional criterion that would force the function $\phi(\mathrm{x})$ to have large gradients at all the root points [22]. The following cost function incorporates both criteria, that is, small values of $\phi(\mathrm{x})$ and large gradients $\|\nabla_{\mathbf{x}} \phi(\mathrm{x})\|$ at the roots (i.e., $\mathrm{x} \in \mathbf{B}$) [22, 14]:

$$E \equiv \frac{1}{2} \sum_{\mathbf{x} \in \mathbf{B}} \frac{\phi^2(\mathrm{x})}{\|\nabla_{\mathbf{x}} \phi(\mathrm{x})\|^2} \tag{2}$$

Mathematically, minimizing this new cost function is equivalent to minimizing the *mean-square distance DIST* between the data and the reconstructed surface [5]. More specifically, given a point x and its (neighboring) normal projection x' ($\phi(\mathrm{x}') = 0$) on the reconstructed surface, a first order Taylor series expansion gives

$$\begin{aligned} \phi(\mathrm{x}) &\approx \phi(\mathrm{x}') + \nabla\phi(\mathrm{x}') \cdot (\mathrm{x} - \mathrm{x}') \\ &\approx \phi(\mathrm{x}') + \nabla\phi(\mathrm{x}) \cdot (\mathrm{x} - \mathrm{x}') \\ &= \nabla\phi(\mathrm{x}) \cdot (\mathrm{x} - \mathrm{x}') \end{aligned} \tag{3}$$

where a constant slope is assumed in the vicinity of x'. Therefore, $\phi^2(\mathrm{x}) = \|\nabla_{\mathbf{x}}\phi(\mathrm{x})\|^2 \cdot \|(\mathrm{x} - \mathrm{x}')\|^2$, and the distance is

$$DIST = \|(\mathrm{x} - \mathrm{x}')\|^2 \approx \frac{\phi^2(\mathrm{x})}{\|\nabla_{\mathbf{x}}\phi(\mathrm{x})\|^2} \tag{4}$$

Having chosen the cost function to be minimized, it is now straightforward for us to apply the standard gradient descent based back-propagation learning (sequential updating) to train the network [14]. Our experiments on training the SRNNs indicated that the recursive (sequential updating) back-propagation was often stuck in local minima where all weights and offsets reached zero. Therefore, we adopted the Levenberg Marquardt version of the batch Gauss Newton method for the training of our SRNN [12]. This algorithm takes advantage of the sum of products of the first order Jacobian matrix to explicitly approximate the Hessian matrix of the second order derivatives.

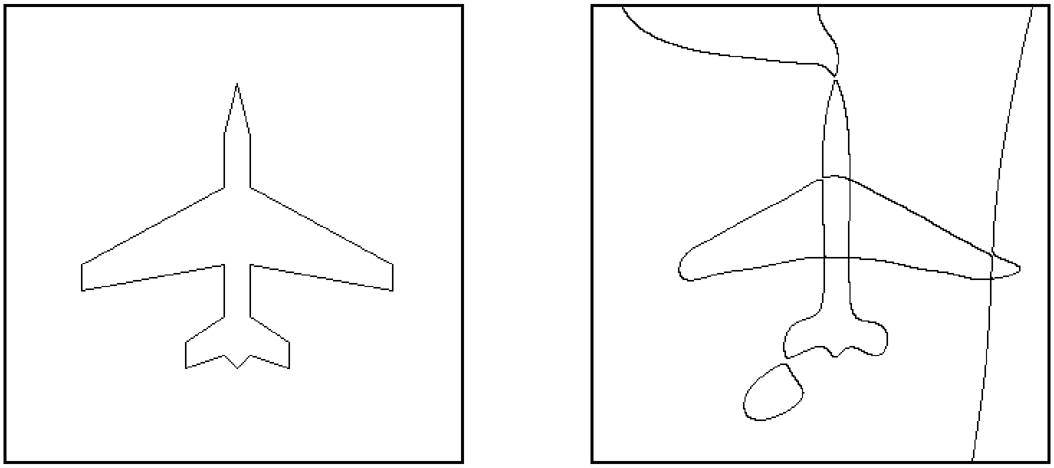
Unfortunately, simulations have indicated the presence of fake boundary curves (or surfaces) in the exterior part of the object due to the difficulty of approximating abrupt changes of some boundary curves. These additional fake curves sometimes degrade the matching accuracy when used in invariant pattern recognition applications [14]. Figure 2 shows the simulation results of the proposed new cost function for modeling a 2-D airplane. Figure 2(a) shows an artificially created airplane. There are 500 boundary points *uniformly* sampled from the airplane used for training the SRNN. The surface reconstructed airplane by an SRNN of 24 hidden neurons is obtained by checking all the points in the 2-D space that correspond to $\phi(\mathrm{x}) = 0$ (see Figure 2(b)).

In addition to the fake boundary curves/surfaces, another problem was also observed about using this new cost function in Eq. (2). The magnitude of the positive and negative values created by SRNN for nonboundary regions are very small (e.g., 0.01) and quite flat (i.e., very small gradient values). Therefore, it is inappropriate to use this reconstructed representation for the measure of degree of matching between a similarly shaped object $\mathbf{B}'$ and the reconstructed object $\mathbf{B}$ regardless the similarity measure is based on $\sum_{\mathbf{x}\in\mathbf{B}'}\phi^2(\mathrm{x})$ or $\sum_{\mathbf{x}\in\mathbf{B}'}\frac{\phi^2(\mathrm{x})}{\|\nabla_{\mathbf{x}}\phi(\mathrm{x})\|^2}$.

2.3 A Regularized Cost Function to be Minimized

To attack the encountered problems of fake boundaries and the small valued flat nonboundary region, regularizing terms are effectively added into the new cost function in Eq. (2) [15]:

$$E \equiv \frac{1}{2}\sum_{\mathbf{x}\in\mathbf{B}}\frac{\phi^2(\mathrm{x})}{\|\nabla_{\mathbf{x}}\phi(\mathrm{x})\|^2} + \frac{1}{2}\sum_{\mathbf{x}\in\mathbf{O}}\left(\frac{1-\phi(\mathrm{x})}{1+\phi(\mathrm{x})}\right)^2 + \frac{1}{2}\sum_{\mathbf{x}\in\mathbf{G}}(\|\nabla_{\mathbf{x}}\phi(\mathrm{x})\| - 1)^2 \tag{5}$$



(a) (b)

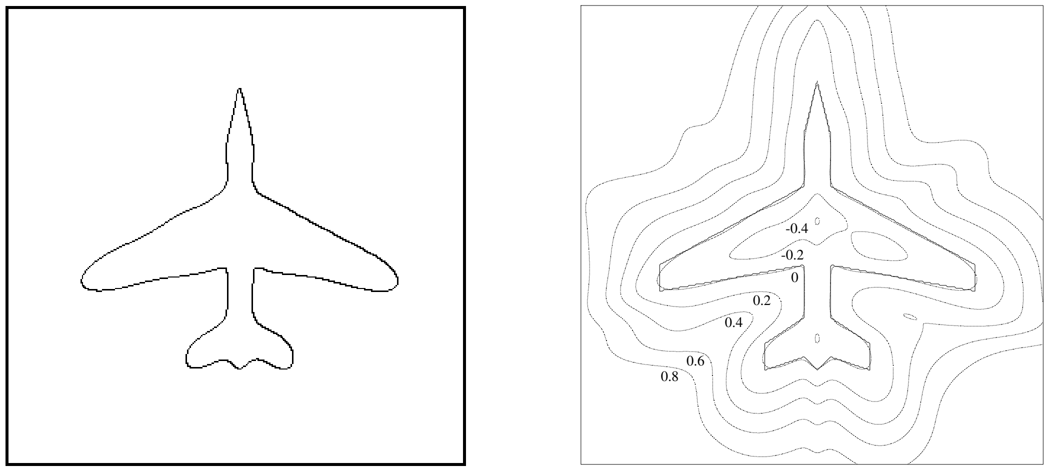
Figure 2: (a) An artificially created airplane. (b) The surface reconstructed airplane by an SRNN of 24 hidden neurons using 500 uniformly sampled boundary points.

where **O** is the set of data points uniformly sampled from a large enough sphere to enclose the whole exemplar object. **G** is the set of data points sampled along the direction of gradient in a constant step starting from each boundary point "after" the SRNN is trained by the data of **B** and **O** sets.

Regularization to Remove Fake Boundary Curves/Surfaces: Note that all the fake boundary curves/surfaces are created due to the inability of SRNN to form a closed object description to some sharp corners. These fake curves/surfaces always extend to infinity at some point in the space. The first regularizing term ($\mathbf{x} \in \mathbf{O}$) is thus added to force the $\phi(\mathbf{x})$ value of the ponits outside the object to be one consistently so that a no-zero-crossing zone outside the object is ensured. This is done by uniformly sampling additional points from a large enough sphere which encloses the exemplar object. In our simulations, the large enough sphere has a radius which is twice of the maximum Euclidean distance of all points in **B** to their centroid. We also found that the convergence rate of training the SRNN using the regularized cost function (using **B** and **O**) is much faster than that of no regularization (using **B** only).

Note that after this regularizing training, the set of boundary points **B** is converted into a whole 2-D/3-D image space description, which is somewhat like a binary image. More specifically, value "1"for exterior region of the exemplar object and negative value of very small magnitude for interior region of the object.

Regularization to Form a Constant Slope Profile: The second regularizing term ($\mathbf{x} \in \mathbf{G}$) is imposed *after* the training of the SRNN using data sets of **B** and **O**. This regularizing term further converts the somewhat binary image description to a constant slope profile image description with respect to the reconstructed boundary. This kind of profile construction is normally achieved by a (chamfer) distance transform in a discrete and nonparametric manner



(a) (b)

Figure 3: (a) The reconstructed curves of the 2-D airplane based on the proposed regularized cost function. (b) The reconstructed constant slope profile of the whole 2-D image given by the trained SRNN.

that approximates the Euclidean distance of each discrete image pixel from its nearest boundary pixel [7]. More specifically, this regularization creates a reconstructed image representation which has a constant slope (e.g., $\|\nabla_{\mathbf{x}}\phi\| = 1$) along the radial direction of the object boundary instead of using very sharp cutoff in representation. In our simulations, the sample points in $\mathbf{G}$ is generated as follows:

1. Compute the gradient vector $\nabla_{\mathbf{x}}\phi$ of each boundary point in $\mathbf{B}$ after the SRNN is trained using $\mathbf{B}$ and $\mathbf{O}$ data sets.

2. Starting from each boundary point, sample along the direction of gradient in a constant step until the other boundary point is met or exceeding a pre-specified limiting coordinate value.

3. Repeat (2) by sampling along the opposite gradient direction.

Figure 3(a) shows the reconstructed shape of the 2-D airplane based on the proposed regularized cost function (see Eq. (5)) using the same 500 uniformly sampled boundary points in Figure 2(b). The reconstructed constant slope profile of the whole 2-D image given by the trained SRNN is shown in Figure 3(b).

3 Alignment of Distorted Objects Before Matching

After trained by the (reference) exemplar object with regularizations, the SRNN establishes an implicit parametric representation of the curves or surfaces (and the whole image space). We are now ready to perform the matching between the represented exemplar objects and any

Figure 4: A trained SRNN for representing the exemplar object is coordinated with the similarity transform module for gradual (mental) object alignment.

distorted testing object $\{\mathbf{y}^{(i)}\}$ without point correspondence if the latter one can be pre-aligned (i.e., appropriately translated, rotated, and scaled). Our design of alignment of the distorted object is based on the studies by experimental psychology that the task of matching rotated and scaled shapes is done by mentally rotating and scaling gradually one of the shapes into the orientation and size of the other one and then tested for a match [8, 19].

Figure 4 shows the gradual (iterative) similarity transform module in coordination with the SRNN. The basic idea is to allow the *similarity transform module* gradually update the similarity transform parameters $(\theta, \mathbf{t}, s)$ through the guidance of the trained SRNN for exemplar object (by keeping constant the weights of the SRNN). The testing boundary points $\{\mathbf{y}\}$ can thus be best transformed (via rotation $\mathbf{R}(\theta)$ of angle vector θ, scaling constant s, and translation vector $\mathbf{t}$) to the same orientation of the exemplar boundary points $\{\mathbf{x}\}$. Note that, $\hat{\mathbf{x}} = s\mathbf{R}(\theta)\mathbf{y} - \mathbf{t}$. Therefore, the goal of the similarity transform module is to have the transformed (aligned) data $\{\hat{\mathbf{x}}\}$ best match that of the surface reconstructed representation $\phi(\mathbf{x})$ of the exemplar object.

Only by using this kind of cascading configuration, we can efficiently provide the necessary mismatch information in the representation domain from the master SRNN to the slave similarity transform module in the training of the latter one. This idea is similar in concept to several neural network controller designs [17]. The updating of the similarity transform parameters can be implemented by an iterative optimization search. We again used the Gauss Newton search in our implementation.

$$\theta_k \Longleftarrow \theta_k - \eta \frac{\partial E}{\partial \theta_k} = \theta_k - \eta \sum_j^m \frac{\partial E}{\partial \hat{x}_j} \frac{\partial \hat{x}_j}{\partial \theta_k} \tag{6}$$

where $\frac{\partial E}{\partial \hat{x}_j}$ is calculated in a network inversion procedure [13], which is similar to the derivative calculation of weights $\frac{\partial E}{\partial w}$ in the back-propagation learning. The mathematical expression of the partial derivatives $\frac{\partial \hat{x}_j}{\partial \theta_k}$ can be pre-calculated since the function (similarity transform equation) is well defined.

The initial guess of translation parameter was obtained via the vector difference of the centroid of the exemplar object and that of the similarity transformed object. Additional care has to be taken in the case of 3-D recognition tasks where only partial viewing data are acquired. Similarly, the initial guess of the scale is obtained by the ratio of the effective volume $V_{\mathbf{x}}$ of the exemplar object to that of $V_{\mathbf{y}}$ the distorted object, where

$$V_{\mathbf{x}} = \frac{1}{||\mathbf{B}||} \sum_{\mathbf{x} \in \mathbf{B}} ||\mathbf{x} - \bar{\mathbf{x}}|| \tag{7}$$

$$V_{\mathbf{y}} = \frac{1}{||\mathbf{B}'||} \sum_{\mathbf{y} \in \mathbf{B}'} ||\mathbf{y} - \bar{\mathbf{y}}|| \tag{8}$$

where $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ denotes the sample average of data sets $\mathbf{B}$ and $\mathbf{B}'$ respectively.

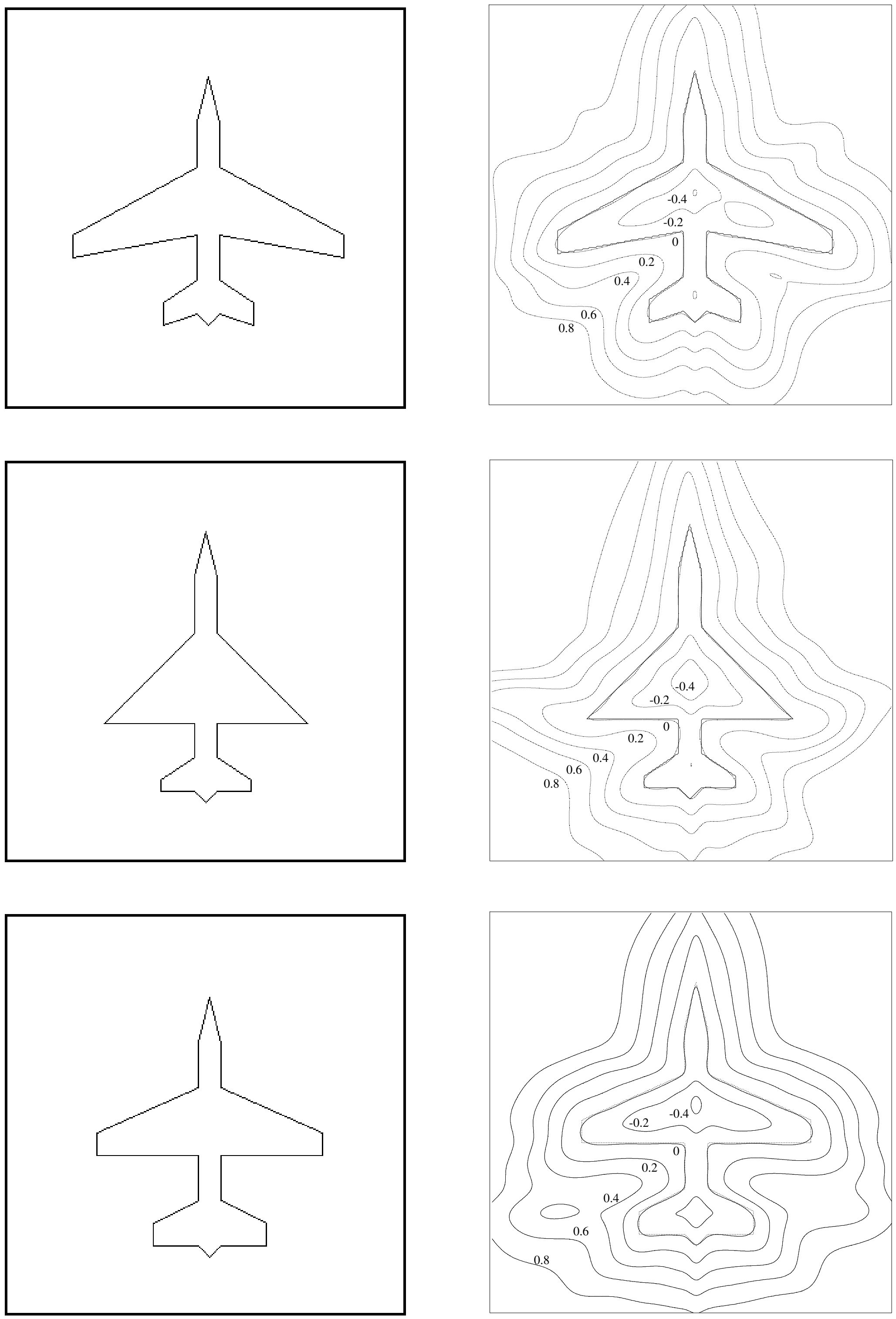
The initial guess of the rotation angle(s) is obtained by starting with several rotation angle combinations. The one gives the smallest matching error from one SRNN is used to compete with other SRNN results. For example, in the 2-D invariant pattern recognition application to be discussed in Section 4, we tried 8 angles with 45° apart (see Section 4). While in the 3-D motion estimation application, 32 combinations out of three angles are tested (see Section 5).

The determination of the best similarity transform alignment goes through a couple of iterative sweeps. In each sweep, two steps of updating are consecutively performed. First, only the rotation and the translation parameters are updated (keeping the scaling parameter fixed). Second, after the rotation and the translation parameters converge, the scaling parameter is updated (keeping rotation and translation parameters fixed). This completed the updating of one sweep. The updating process continues until no more changes through all the three steps in one sweep.

4 Application to 2-D/3-D Invariant Pattern Recognition

For invariant pattern recognition, one is required to represent each distinct 2-D/3-D exemplar object with a separate SRNN. Each SRNN can be trained with a quite complete set of (usually noise free) boundary data of the corresponding exemplar object. In our experiments, three classes of airplanes were recognized (see Figure 5(a)). Each class was represented by a 2-layer SRNN (with 24 hidden units) trained by 500 noise-free boundary points uniformly sampled from the corresponding exemplar object within a square box of 500×500. The SRNN reconstructed boundary shapes and the constant slope 2-D image profiles are shown in Figure 5(b).

Each testing distorted object was specified by 100 randomly selected boundary points. The amounts of similarity transforms for each testing object were: the translation was random within $[-50,\ 50]$ for each coordinate, scaling was random within $[0.2,\ 2]$, and rotation was also random within $[-180°,\ 180°]$. A valid combination of the similarity transform ensures all the 100 sample points are confined within the square box domain. For each distorted testing object, we further create two additional testing objects by corrupting with: small Gaussian noise $N(\mu, \sigma) = N(0, 0.05V_{\mathbf{y}})$ to each coordinate of the boundary points, and large noise large Gaussian noise $N(\mu, \sigma) = N(0, 0.1V_{\mathbf{y}})$ to each coordinate of the boundary points, where $V_{\mathbf{y}}$ is the effective volume defined in Eq. (8).



(a) (b)

Figure 5: (a) Three exemplar airplanes to be recognized. (b) The SRNN reconstructed boundary shapes and 2-D image profiles.

Exemplar Objects $\sigma = 0.00V_{\mathbf{y}}$, No Noise	Airplane **1**	Airplane **2**	Airplane **3**
1 (100 Tests)	100 %	0 %	0 %
2 (100 Tests)	0 %	100 %	0 %
3 (100 Tests)	0 %	0 %	100 %
Average Accuracy		100 %	

Exemplar Objects $\sigma = 0.05V_{\mathbf{y}}$, Small Noise	Airplane **1**	Airplane **2**	Airplane **3**
1 (100 Tests)	100 %	0 %	0 %
2 (100 Tests)	1 %	99 %	0 %
3 (100 Tests)	1 %	0 %	99 %
Average Accuracy		99.3 %	

Exemplar Objects $\sigma = 0.10V_{\mathbf{y}}$, Large Noise	Airplane **1**	Airplane **2**	Airplane **3**
1 (100 Tests)	99 %	1 %	0 %
2 (100 Tests)	20 %	80 %	0 %
3 (100 Tests)	24 %	0 %	76 %
Average Accuracy		85.0 %	

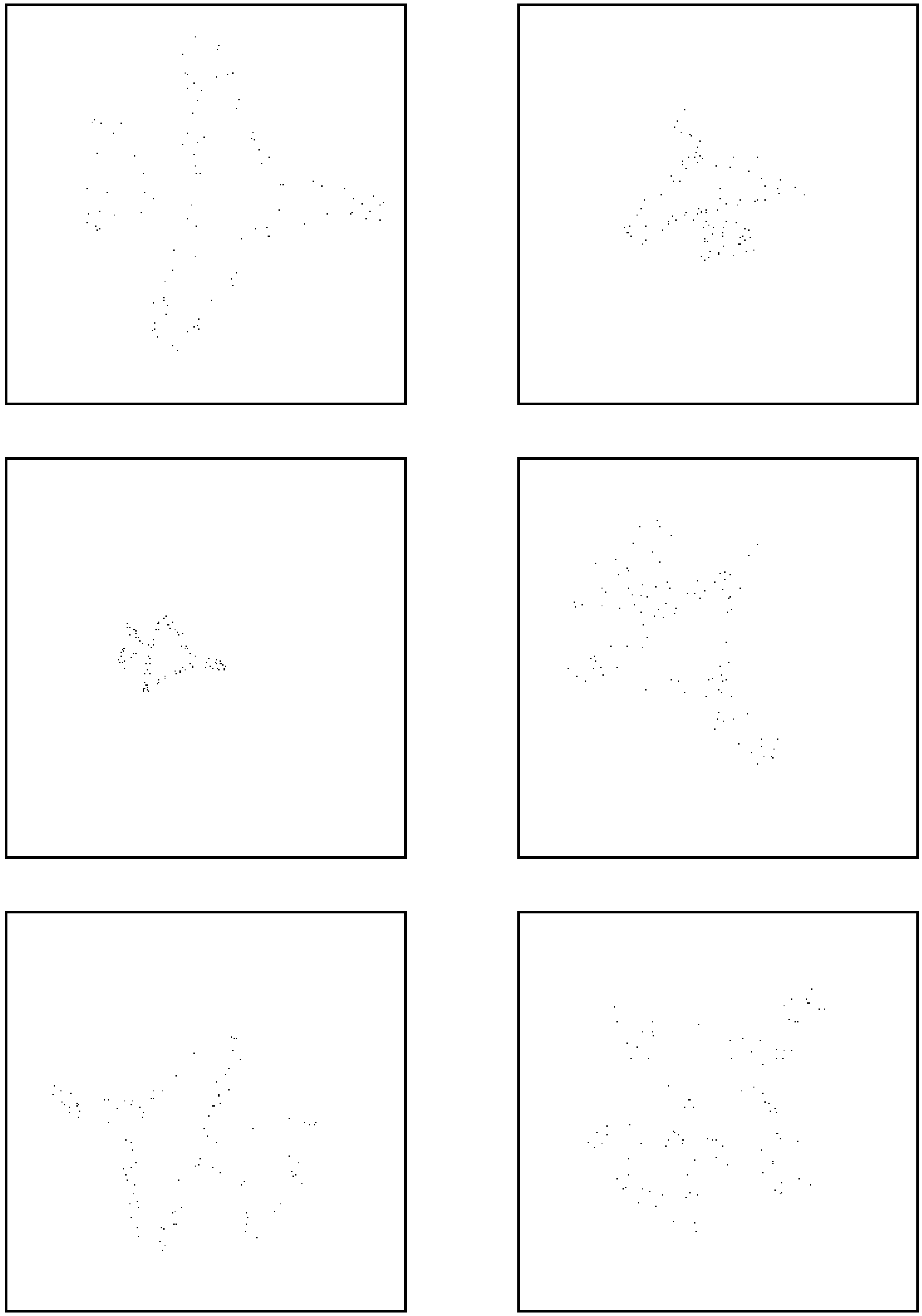
Table 1: The performances of three simulation environments.

For each exemplar airplane, there were 300 distorted testing airplanes (100 noiseless, 100 small noise, and 100 large noise) created, and the standard squared error

$$L_2 = \sum_{\mathbf{y} \in \mathbf{B}'} \phi^2(\mathbf{y}) \tag{9}$$

was used as an indicator of the goodness of matching. Figures 6(a) and (b) show the typical distorted 100-point testing boundary data randomly sampled from three testing objects with small and large noises.

To be more robust in the rotation alignment, there are 8 initial guesses of rotation angles were tried, i.e., $\theta = 0°, 45°, ..., 315°$. For each SRNN, the smallest L_2 value out of the eight initial rotation angles was chosen to be the matching score for this class. Then the smallest L_2 value out of all three exemplar SRNNs determined the winner of the classification. The classification performance for three test environments is presented as confusion matrices of accuracy given in Table 1.



(a) (b)

Figure 6: Typical 100-point testing boundary data randomly sampled from three testing objects. (a) With small noise. (b) With large noise.

5 Application to 3-D Motion Estimation using Partial Viewing Data from A Single Aspect View

The proposed SRNN can be naturally applied to 3-D invariant pattern recognition applications where the testing data might represent partial portion of the complete object due to a limited number of viewing aspect angles. The is demonstrated by the following 3-D motion estimation example.

Given a known 3-D moving object already represented by an SRNN, we are to estimate its motion via estimating the amount of translation and rotation based on a sequence of sets of 3-D sparse range data $\mathbf{B}'_k = \mathrm{y},\ k = 0, 1, \ldots,$ taken from one (varying) aspect of the object. Any similarity transformed data $\mathrm{y} = (y_1, y_2, y_3)^T$ on the moving object is obtained by randomly sampling the surface points from the rotated and translated version of the exemplar object. To minimize the complexity involved, the size scaling effect is ignored in this application (i.e., $s = 1$). By using the similarity transform module cascaded between the input and the trained SRNN (see Figure 4), the motion estimation can be done in a similar manner as invariant pattern recognition. Note that we are aligning the distorted object through the similarity transform module to the orientation of the exemplar object. Therefore, the motion parameters estimated are defined with respect to the distorted object, which is different from the usual definition of motion parameters with respect to the exemplar object, this concern can be easily answered. As given in Figure 4 (also assume $s = 1$)

$$\hat{\mathrm{x}} = \mathbf{R}(\theta)\mathrm{y} - \mathbf{t} \tag{10}$$

$$\mathbf{R} = \begin{bmatrix} \cos\alpha\ \cos\beta & \cos\alpha\ \sin\beta\ \sin\gamma - \sin\alpha\ \cos\gamma & \cos\alpha\ \sin\beta\ \cos\gamma + \sin\alpha\ \sin\gamma \\ \sin\alpha\ \cos\beta & \sin\alpha\ \sin\beta\ \sin\gamma + \cos\alpha\ \cos\gamma & \sin\alpha\ \sin\beta\ \cos\gamma - \cos\alpha\ \sin\gamma \\ -\sin\beta & \cos\beta\ \sin\gamma & \cos\beta\ \cos\gamma \end{bmatrix} \tag{11}$$

where α, β, γ, and $\mathbf{t} = (t_1, t_2, t_3)^T$ represent roll, pitch, yaw rotation angles and translation vector respectively with respect to the distorted object. Eq. (10) can be rewritten as

$$\mathrm{y} = \mathbf{R}^T\hat{\mathrm{x}} + \mathbf{R}^T\mathbf{t} \tag{12}$$

Therefore, after the similarity transform module estimates the $\mathbf{R}$ and $\mathbf{t}$ motion parameters with respect to the distorted object, the desired rotation matrix with respect to the exemplar object can be easily obtained as $\mathbf{R}^T$ and the desired translation vector with resepct to the exemplar object is $\mathbf{R}^T\mathbf{t}$.

By taking advantage of the continuity assumption in tracking the motion, the initial guess of angles (or locations) at each time instance is set to be the estimated angles (or locations) at previous time instance except the initial guesses in the very beginning, i.e., the 0^{th} time instance.

Without loss of generality, let's assume the exemplar object is centered at origin. Due to large portion of missing data resulted from a single aspect range data collection, the initial guess of $\mathbf{t}$ in the similarity transform module at the 0^{th} time instance cannot be solely relied on the centroid information of the partial data $\mathbf{B}'_0$. The centroid $\bar{\mathrm{y}} = (\bar{y}_1, \bar{y}_2, \bar{y}_3)^T$ is defined from sample average of the single aspect range data in $\mathbf{B}'_0$. If the range finder is assumed to point to y_3 direction, then the initial guesses of t_1 and t_2 at the 0^{th} time instance can still be set to be

$\bar{y}_1$ and $\bar{y}_2$, while the initial guess of t_3, the depth of the data, at the 0^{th} time instance should be compensated to be

$$t_3 = \bar{y}_3 + \sqrt{V_{\mathbf{x}}^2 - V_{\mathbf{y}}^2}, \tag{13}$$

where $V_{\mathbf{x}}$ and $V_{\mathbf{y}}$ are the effective volumes of exemplar object and the single aspect range finding object (see Eq. (8)).

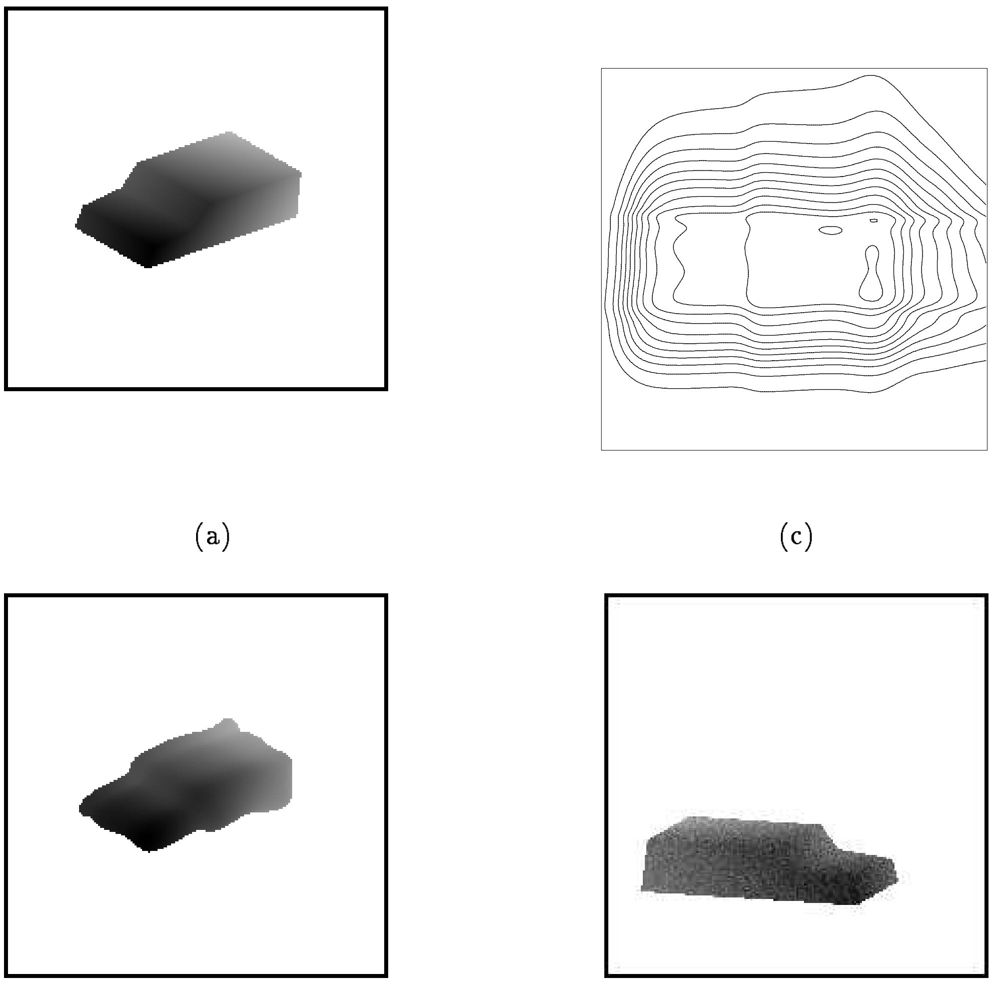
The initial guess of the rotation angles at the 0^{th} time instance is obtained by starting with 32 different possible rotation combinations: $\alpha = \{-135°, -45°, 45°, 135°\}$, $\beta = \{-45°, 45°\}$, and $\gamma = \{-135°, -45°, 45°, 135°\}$. After the presentation of 1st single aspect range data set $\mathbf{B}'_1$, the one produces smallest matching error L_2 is chosen as the initial guess for future motion tracking.

To demonstrate that our motion estimation can work in a wide range of motion parameters, a simulation with trajectory of 101 time instances was conducted. First, a synthetic exemplar object (3-D truck) was generated and one view of the object is shown in Figure 7(a). There are 600 range data uniformly sampled from six views (front, end, left, right, top, and bottom) used for training the SRNN, which consists of one hidden layer of 16 hidden neurons. Figure 7(b) shows one aspect of the reconstructed SRNN boundary representation. One single cross section of the constant slope profile of the 3-D image space representing the 3-D truck is shown in Figure 7(c).

This SRNN is then cascaded with the similarity transform module for motion estimation of the 3-D truck. We fixed the camera viewing angle and moved (translate and rotate) the truck. Images with noise-free and Gaussian noise $N(0, 0.2V_{\mathbf{y}})$ were used to test the performance of our method. Figure 7(d) shows the noisy range data collected from a single aspect at the first time instance. There were only 100 range data randomly sampled at each time instance. Figures 8(a) and (b) show the 100 data samples (noise-free and noisy) collected from the range finder pointing to y_3 direction. The corresponding 3-D trucks of the distorted data sampled from are also shown. The estimation results of 6 motion parameters (3 rotation angles and 3 translation coordinates) during the motion tracking in the 101 time instances are shown in Figures 9 and 10.

6 Conclusion

This paper presents a regularized surface reconstruction neural network approach to perform robust invariant 2-D/3-D object recognition and motion estimation. By efficiently embed the whole 2-D/3-D image space into a neural network into a neural network parametric representation, we are able to elegantly duplicate the human's mental image transform and matching capability in performing the rotating and scaling of objects as suggested by the studies of experimental psychology. The preliminary simulations of applying this technique to invariant 2-D target classification and 3-D object motion estimation using sparse range data collected from a single aspect view are encouraging. Further research in using this technique for standard 2-D images (instead of range data) is explored.



(a) (c)

(b) (d)

Figure 7: (a) A synthetic exemplar object (3-D truck). (b) One aspect view of the trained SRNN representation. (c) One cross section of the reconstructed SRNN constant slope representation of the whole 3-D image. (d) The noisy range data collected from a single aspect view for motion estimation.

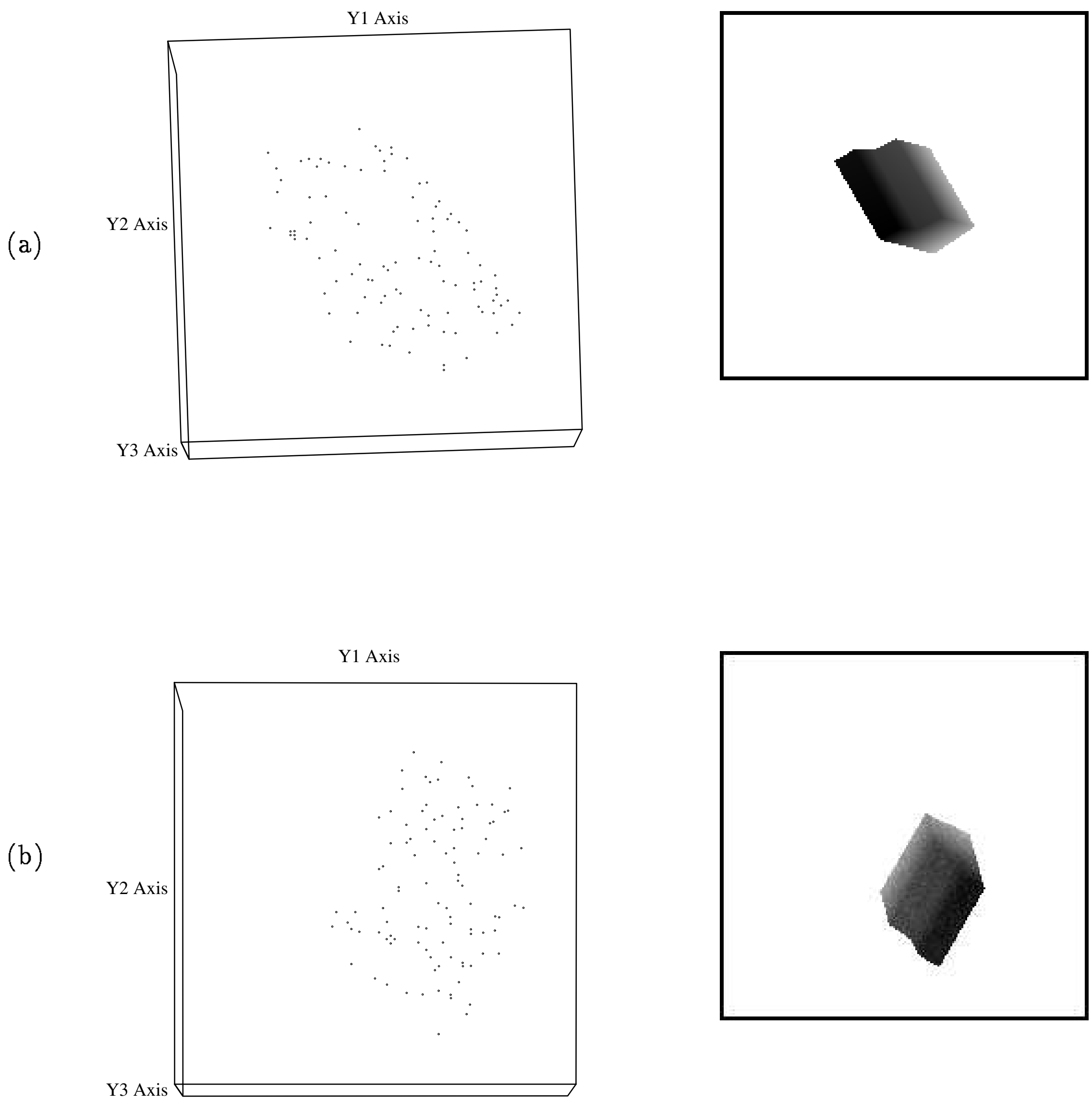


Figure 8: (a) Typical 100 noise free range data of the 3-D truck sampled from a single aspect view. (a) Typical 100 noisy range data of the 3-D truck sampled from a single aspect view.

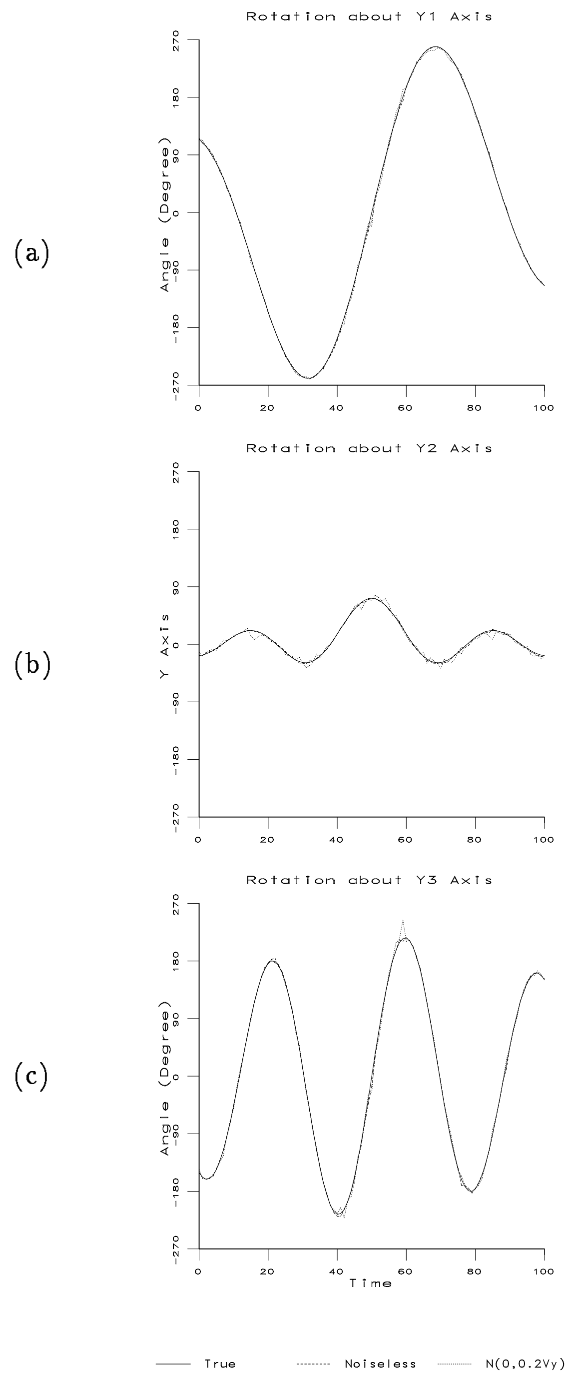


Figure 9: Motion estimation results of 3 rotation angles based on noiseless and noisy sparse range data collected from a single aspect view. (a) Estimation of α. (b) Estimation of β. (c) Estimation of γ.

(a)

(b)

(c)

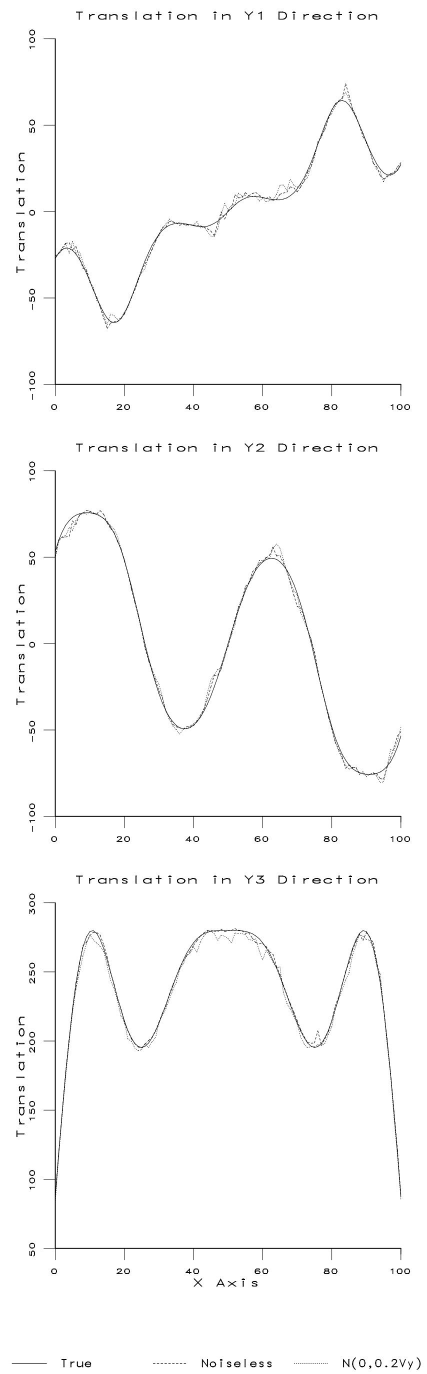


Figure 10: Motion estimation results of 3 translation coordinates based on noiseless and noisy sparse range data collected from a single aspect view. (a) Estimation of t_1. (b) Estimation of t_2. (c) Estimation of t_3.

References

[1] K. S. Arun, T. S. Huang, S. D. Blostein. Least squares fitting of two 3-D point sets. *IEEE Trans. on PAMI*, 9(5):698-700, Sept. 1987.

[2] E. Barnard and D. Casaasent. Invariance and neural nets. IEEE Trans. on Neural Networks, 2(5):498-508, September 1991.

[3] R. H. Bartels, J. C. Beatty, and B. A. Barsky. An Introduction to Splines for use in Computer Graphics and Geometric Modeling. Morgan Kaufmann Publishing Inc., Los Altos, CA 1987.

[4] R.M. Bolle. Information extraction about complex three-dimensional objects from visual data. Ph.D. Dissertation, Brown University, May 1984. (Also Brown Univ. Tech. Rep. LEMS 6.)

[5] R. J. Bolle and B. C. Vermuri. On three-dimensional surface reconstruction methods. *IEEE Trans. on PAMI*, 13(1):1-13, January 1991.

[6] R.M. Bolle and D.B. Cooper. On optimally combining pieces of information, with application to estimating 3-D complex-object position from range data. IEEE Trans. on PAMI, 8(5):619-637, September 1986.

[7] G. Borgefors. Hierarchical chamfer matching: a parametric edge matching algorithm. IEEE Trans. on PAMI, 10(6):849-865, Nov. 1988.

[8] C. Bundesen and A. Larsen. Visual Transformation of Size. Journal of Experimental Psychology: Human Perception and Performance, 1(3):214-220, 1975.

[9] B. K. Horn, H. M. Hilden, S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A*, 5(7):1127-1135, July 1988.

[10] K. Hornik. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[11] T. S. Huang, S. D. Blostein, E. A. Margerum. Least squares estimation of motion parameters from 3-D point correspondences. In Proc. *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 198-201, Miami Beachd, FL, 1986.

[12] J. N. Hwang and P. S. Lewis. From nonlinear optimization to neural network learning. In Proc. *24th Asilomar Conf. on Signals, Systems, & Computers*, pp. 985-989, Pacific Grove, CA, November 1990.

[13] J. N. Hwang, J. J. Choi, S. Oh, and R. J. Marks II. Query based learning applied to partially trained multilayer perceptrons. IEEE Trans. on Neural Networks, Vol. 2, No. 1, pp. 131-136, January 1991.

[14] J.N. Hwang and H. Li. Invariant object recognition via surface reconstruction neural networks. Int'l Joint Conf. on Neural Networks, Vol. 3, pp. 184-189, 1992.

[15] J.N. Hwang and Y. H. Tseng. 3-D motion estimation using single perspective sparse range data via surface reconstruction neural networks. Submitted to Int'l Conf. on Neural Networks, San Francisco, CA, March 1993.

[16] Z.C. Lin, H. Lee, T. S. Huang. A frequency-domain algorithm for determining motion of a rigid body from range data without correspondences. In Proc. *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 194-198, Miami Beach, FL, 1986.

[17] D. Nguyen and B. Widrow. The truck backer-upper: an example of self-learning in neural networks. In *Int'l Joint Conf. on Neural Networks (IJCNN), Washington D.C.*, pages II357–II363, June 1989.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representation by error propagation. in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Volume 1: Foundations*, Chapter 8. MIT Press, Cambridge, MA, 1986.

[19] R. N. Shepard and L. A. Cooper, Mental Images and Their Transformations. MIT Press, Cambridge, Massachusetts, 1982.

[20] L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. IEEE Trans. on PAMI, 14(11):1061-1075, November 1992.

[21] R. Szeliski. Estimating motion from sparse range data without correspondence. Int'l Conf. on Computer Vision, pp. 336-340, 1988.

[22] G. Taubin. Algebraic nonplanar curve and surface estimation in 3-space with applications to position estimation. Technical Report, Brown University, LEMS-43, Feb. 1988.

[23] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. on PAMI*, 13(11):1115-1138, November 1991.

[24] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. on PAMI*, 13(4):376-380, April 1991.

[25] B. Widrow. The rubber mask technique – I and II. *Pattern Recognition*, 5:175-211, 1973.

Contents

List of Figures