Principal Components Analysis and Neural Network Implementation of Photometric Stereo

Yuji Iwahori
Faculty of Engineering
Nagoya Institute of Technology
Nagoya 466, Japan
iwahori@center.nitech.ac.jp

Robert J. Woodham
Dept. of Computer Science
University of British Columbia
Vancouver, BC, Canada
woodham@cs.ubc.ca

Ardeshir Bagheri
Dept. of Computer Science
Langara College
Vancouver, BC, Canada
bagheri@cs.ubc.ca

Abstract

An implementation of photometric stereo is described in which all directions of illumination are close to the viewing direction. This has practical importance but creates a numerical problem that is ill-conditioned. Ill-conditioning is dealt with in two ways. First, many more than the theoretical minimum number of required images are acquired. Second, principal components analysis (PCA) is used as a linear preprocessing technique to extract a reduced dimensionality subspace to use as input. Overall, the approach is empirical. The ability of a radial basis function (RBF) neural network to do non-parametric functional approximation is exploited. One network maps image irradiance to surface normal. A second network maps surface normal to image irradiance. The two networks are trained using samples from a calibration sphere. Comparison between the actual input and the inversely predicted input is used as a confidence estimate. Results on real data are demonstrated.

1 Introduction

Shape-from-shading, originally formulated by Horn [1, 2], is the problem of determining surface shape from the smooth shading present in a single image. Photometric stereo was introduced by Woodham [3] as a method to determine surface orientation locally using multiple images of an object surface acquired from a single viewpoint under different conditions of illumination. These early approaches, and others that followed, exploit principles of optics as a source of radiometric constraint.

In recent years, theoretical aspects of shape-from-shading and photometric stereo have dominated the literature arguably at the expense of practical aspects. Much of the theoretical work reported considers only the case of Lambertian reflectance. Unfortunately, this has served to convince potential implementors that the approach is of little practical significance. Within the "physics-based vision" community, serious attention now is paid to reflectance models [4, 5]. Clearly, this work is fundamental. But, it has not yet proven decisive in leading to practical applications.

Key factors limiting practical application of photometric stereo include shadows and interreflection and the reality of measurement situations in which the range of available illumination directions is restricted. One trade-off is clear. The greater the difference in the directions of illumination, the better conditioned is the estimation of surface orientation. On the other hand, the greater the difference in the directions of illumination, the greater is the number of visible surface points that fail to be commonly illuminated and hence fail to support any estimation of surface orientation. Further, for non-convex surfaces, directions of illumination different from the viewing direction exacerbate the problem caused by shadows cast onto surface points that would otherwise be commonly illuminated. Shadows cause image measurements to be darker than expected. Interreflection causes image measurements to be brighter than expected. Photometric stereo is subject to error in the presence of cast shadows and interreflection. No purely local technique can succeed since these phenomena are inherently non-local. Nevertheless, in most cases, one can locally detect the presence of cast shadows and interreflection by exploiting the data redundancy in photometric stereo [6]. Finally, in tasks ranging from ocular surgery, mining, sub-sea exploration and mobile robotics, there often are restrictions on the range of available illumination directions as well as a requirement to keep the entire sensing system as physically compact as possible. This suggests the need to develop techniques to deal with illumination directions nearly collinear with the viewing direction. If this can be done, it also has the advantage of minimizing regions affected by shadows.

Woodham has developed implementations of photometric stereo in which the reflectance function is determined empirically using a calibration object of known shape, typically a sphere. No explicit assumptions need to be made either about light source directions or about the functional form of surface reflectance. It is sufficient that the calibration sphere and subsequent test objects be viewed under the same conditions of illumination and be made of the same material (i.e., have the same reflectance properties). In this way, a material with any reflectance characteristic can be handled, provided that the necessary calibration can be done. Empirical calibration has the added benefit of automatically compensating for the transfer characteristics of the sensor. In [6], the non-

0-8186-7021-5/95 $4.00 © 1995 IEEE

linear mapping between image irradiance and surface orientation was represented explicitly in a lookup table (LUT). This allowed near real-time implementation of three light source photometric stereo on full frame video data at near video frame rates (i.e., 15Hz). The implementation used commercially available hardware. Of course, should a good physical reflectance model be available for the given application, it can be retrofit to the method for implementation purposes. Calibration may still be required, however, to estimate the unknown parameters of the reflectance model [7], including the direction(s) to the light source(s).

Iwahori has pursued neural network implementations of photometric stereo. In [8], a particular functional model of specular reflectance was assumed and the neural network was used to estimate the unknown parameters in what overall remained a parametric method. In [9], a neural network implementation was used for direct comparison with the lookup table (LUT) implementation described in [6]. The comparison was favourable indicating that a neural network implementation is a viable alternative to an explicit LUT.

The idea of moving light source photometric stereo has previously been explored for a variety of light source configurations under the assumption of Lambertian and other reflectance models [10, 11, 12, 13]. Not surprisingly, it was observed that, with only small movements of the light source, standard photometric stereo becomes ill-conditioned. For this reason, it seemed difficult to realize implementations based on a single moving light source. Nevertheless, the potential advantage was clear and we were motivated to explore further.

The novel idea in this paper is to use principal components analysis (PCA) in conjunction with a neural network to successfully implement photometric stereo for the case of a moving, nearby light source. It is shown that PCA can overcome the fact that with only a limited range of light source directions standard photometric stereo is ill-conditioned. The work also exploits the ability of neural networks to do non-parametric functional approximation and to "learn" the required mapping between input space and the corresponding surface orientation. Data obtained from a calibration sphere are used to train the network. As with both previous non-parametric, empirical implementations [6, 9], no explicit assumptions need to be made either about light source directions or about the functional form of surface reflectance.

The particular neural network approach described uses a radial basis function (RBF) neural network and the orthogonal least squares (OLS) learning method described in [14]. Further, we continue to exploit the redundancy inherent in photometric stereo to determine a local confidence estimate. This is achieved by training a two distinct neural networks. The first predicts the intended output (surface normal) from the three most significant inputs determined by PCA. The second inversely predicts the three inputs from the estimated output (surface normal). The difference between the actual input and the inversely predicted input is used as the confidence estimate. This helps to detect local regions where photometric stereo fails due to cast shadow or significant interreflection. Experiments on real data are described.

2 Photometric Stereo

A standard geometry for shape from shading is assumed. That is, let the object surface be given explicitly by $z = f(x, y)$ in a right-handed Euclidean 3D scene coordinate system, with the positive Z direction pointing to the viewer. Image projection is assumed to be orthographic with the 2D image XY axes coincident with the 3D scene XY axes. Let a (unit) surface normal vector at any surface point be $[n_1, n_2, n_3]$. Then an image irradiance equation can be written as

$$E(x,y) = R(n_1, n_2, n_3) \tag{1}$$

where $E(x, y)$ is the image irradiance and $R(n_1, n_2, n_3)$ is the reflectance map, defined using the surface normal, $[n_1, n_2, n_3]$, to represent surface orientation. A reflectance map combines information about surface material, scene illumination and viewing geometry into a single representation that determines image irradiance as a function of surface orientation. When using the surface normal, $[n_1, n_2, n_3]$, to represent surface orientation, it may appear that there are three unknowns to determine. But, there are only two since we have the *a priori* constraint that the length of the normal vector is one. This constraint is not used immediately but does play a role in the neural network implementation.

Photometric stereo uses multiple images obtained under the identical geometry but with different conditions of illumination. With p light source directions, p images, and hence p equations, are obtained

$$\begin{aligned} E_1(x,y) &= R_1(n_1,n_2,n_3) \\ E_2(x,y) &= R_2(n_1,n_2,n_3) \\ &\vdots \\ E_p(x,y) &= R_p(n_1,n_2,n_3) \end{aligned} \tag{2}$$

With $p > 2$, the p image irradiance measurements, $E_1, E_2, \ldots, E_p$, generally overdetermine the local surface orientation at each point, (x, y), since surface orientation only has two degrees of freedom. But, if all directions of illumination are nearly collinear, as will be the case here, the problem can be ill-conditioned.

Ill-conditioning is dealt with by choosing p large and by using PCA of the resulting p image irradiance measurements as a preprocessing technique to extract a reduced dimensionality subspace to use as input for photometric stereo.

For the case of Lambertian reflectance, (2) is a linear set of equations that can be written as

$$\begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_p \end{bmatrix} = \mathbf{L} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \tag{3}$$

where $\mathbf{L}$ is a $p \times 3$ matrix whose ith row is a vector that points at light source i and that has magnitude proportional to the (relative) strength of light source i, $i = 1, 2, \ldots, p$. One could use a variety of techniques to "solve" (3) for $[n_1, n_2, n_3]$. For example, singular value decomposition (SVD) can be used to determine the effective rank of $\mathbf{L}$, as well as the standard least squares best solution to (3). The SVD of $\mathbf{L}$ can be interpreted as the determination of linear combinations of light source directions that approximate (3) as a reduced dimensionality linear problem of equivalent signal to noise ratio.

Via (2), we accept the fact that the photometric stereo problem is, in general, non-linear. This non-linearity is represented using a neural network. At the same time, there is a similarity in that PCA is used as a linear data preprocessing technique, both to decorrelate the image irradiance measurements and to reduce the dimensionality of the non-linear problem to be solved to one of hopefully equivalent signal to noise ratio.

It should be noted that some formulations of photometric stereo, including the original [3], allow the material's bidirectional reflectance factor (i.e., albedo) also to vary spatially. With varying albedo, there are three unknowns, two for surface orientation and one for albedo. Solutions to (2) or (3) need not explicitly enforce the constraint that the normal vector, $[n_1, n_2, n_3]$, is of unit length. Under appropriate assumptions, the length of the solution to (2) or (3) can be related to albedo. Here, constant albedo is assumed so that the problem, as defined, remains overdetermined locally. We can then exploit the redundancy inherent in the image irradiance data to determine a local confidence estimate. This is key to detecting regions of cast shadow and interreflection.

3 Principal Components Analysis

Principal components analysis (PCA) is a classical technique in multivariate statistical data analysis. The basic idea is to describe the dispersion of an array of n points in a p-dimensional space via a new set of orthogonal linear coordinates, called the principal components, determined so that the principal components are mutually uncorrelated and so that the sample variances of the n points are ranked in decreasing order of magnitude with respect to these new coordinates. PCA is an invertible coordinate transformation that captures no more (or no less) information than was originally present.

Algebraically, PCA involves finding the eigenvalues and eigenvectors of the sample covariance matrix. (A variant of standard PCA uses the sample correlation matrix instead). A covariance matrix is symmetric and positive definite so that its eigenvectors form an orthogonal set. The (normalized) eigenvectors define the new set of coordinates axes (i.e., the principal component axes). Since PCA does remove correlation from data coordinates (in a p-dimensional space), it does have application to dimensionality reduction, given sample data with significant correlation. Dimensionality reduction typically is achieved by simple elimination of lower order principal components. Additional data reduction is possible via re-quantization since, in a fixed point, finite precision, implementation, it is possible to allocate the number of bits-per-component differentially to contribute most to overall problem accuracy. The analogy is to FM stereo radio and to NTSC colour video. In FM stereo, left and right audio channels are linearly transformed into sum and difference components. Similarly, in NTSC colour video, three RGB colour video channels are linearly transformed into the standard YIQ components. In both cases, owing to the high degree of correlation in the original representations, differential bandwidth can be allocated to the transformed components resulting in significant data compression. With p directions of illumination that are nearly collinear, one expects a high degree of correlation between images. It is because of this that PCA is helpful.

PCA is implemented as follows. First, calculate the $p \times p$ covariance matrix, $\mathbf{K}$, using sampled object points commonly illuminated in each of the p input images. Second, determine the normalized eigenvectors, ϕ_k, and associated eigenvalues, λ_k, of the covariance matrix, $\mathbf{K}$, $k = 1, 2, \ldots, p$. The normalized eigenvectors define a new, orthonormal basis for p-space. Let $\mathbf{\Phi}$ be the $p \times p$ matrix whose columns are ϕ_k, $k = 1, 2, \ldots, p$. Let $[E_1, E_2, E_3, \ldots, E_p]$ be the p-tuple of image irradiance measurements from a given object point. PCA defines a new p-tuple, denoted here by $[A_1, A_2, A_3, \ldots, A_p]$, where

$$\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_p \end{bmatrix} = \mathbf{\Phi}^T \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_3 \end{bmatrix} \tag{4}$$

(^{T} denotes matrix transpose). That is, the matrix $\mathbf{\Phi}$ determines the coefficients required to represent the given $[E_1, E_2, E_3, \ldots, E_p]$ in terms of the new basis, ϕ_k, $k = 1, 2, \ldots, p$. For each of n sample p-tuples, one obtains a corresponding $[A_1, A_2, A_3, \ldots, A_p]$. The kth principal component, viewed as a data set, is the set of all n values of the kth coordinate of $[A_1, A_2, A_3, \ldots, A_p]$. The variance of the kth principal component is equal to the corresponding eigenvalue, λ_k. The total variance is given by the trace of the covariance matrix, $\mathbf{K}$, (or equivalently by the sum of the p eigenvalues). Thus, the proportion of the total variance accounted for by principal component k is

$$\frac{\lambda_k}{\sum_{i=1}^{p} \lambda_i} \tag{5}$$

Given an estimate of the signal to noise ratio, one could predict how much of the total variation is due to signal and how much is due to noise. This would provide some guidance as to how many principal components are significant. In the absence of knowledge about measurement noise, the analysis still reveals how much of the total sample variance is being accounted for. Suppose, as is done here, one chooses to retain the three most significant principal components for subsequent analysis. Let $\hat{\mathbf{\Phi}}$ be the $p \times 3$ matrix whose columns are ϕ_k, $k = 1, 2, 3$. The 3-dimensional subspace selected is

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} = \hat{\mathbf{\Phi}}^T \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_p \end{bmatrix} \qquad (6)$$

The proportion of the total variance accounted for is

$$\frac{\sum_{i=1}^{3} \lambda_i}{\sum_{i=1}^{p} \lambda_i}$$

4 NN Functional Approximation

4.1 RBF Networks and OLS Learning

Neural networks are attractive for non-parametric functional approximation. A radial basis function (RBF) neural network is one choice suitable for many applications. In particular, it has been widely used for strict interpolation in multidimensional spaces [15, 16]. It is argued that RBF neural networks often can be designed in a fraction of the time it takes to train standard feed-forward networks. They are claimed to work well when many training vectors are available.

RBF networks represent non-linearity via the choice of basis functions. A Gaussian isn't the only choice of radial basis function for RBF networks but it is the choice widely used and the one used here. One common learning algorithm for RBF networks is based on first randomly choosing data points as RBF centers and then solving for the optimal weights of the network. Performance, however, critically depends on the chosen centers. In practice the centers often are chosen to be an arbitrary selected subset of the data points. This selection mechanism typically is unsatisfactory. The resulting network may perform poorly, because the centers do not suitably sample the input data, or it may have excessive size, if a very large number of centers is used.

An alternative learning procedure is based on an orthogonal least squares (OLS) method [14, 16]. Of course, the performance of an RBF network still critically depends on the chosen centers. Because a fixed center corresponds to a given regressor in a given regression model, the selection of RBF centers can be regarded as a problem of subset model selection. The OLS method can be employed as a forward regression procedure to select a suitable set of centers (regressors) from a large set of candidates. At each step of the regression, the increment to the explained variance of the desired output is maximized.

The learning procedure adopted here is based on the above OLS learning method. It chooses radial basis function centers one by one in a systematic way until an adequate network has been constructed. The algorithm has the property that each selected center maximizes the increment to the explained variance of the desired output while remaining numerically well-conditioned.

With this learning procedure, two RBF networks are trained using input/output data from a calibration sphere. Here, we use reduced dimensionality data preprocessed by PCA as described in section 3. Many training vectors are available since data from the calibration sphere are dense and include all possible visible surface normals, $[n_1, n_2, n_3]$. Each RBF network used consists of two layers, (i.e., a hidden layer of P neurons and an output layer of 3 neurons), as shown in Figure 1.

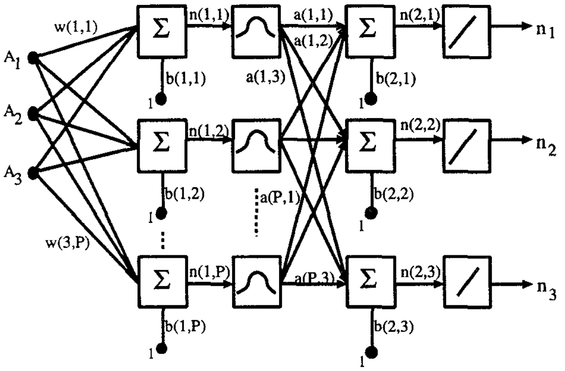


Figure 1: Radial Basis Function Neural Network

The learning procedure builds an RBF neural network one neuron at a time. Neurons are added to the network until the sum-squared error falls beneath an error goal (or a maximum number of neurons has been used). In learning, it is important that the so-called spread constant of the radial basis function be large enough that the neurons respond to overlapping regions of the input space, but not so large that all the neurons respond in essentially the same manner. Once learning is complete, that which has been learned is represented by the weights connecting each RBF neural network unit. The resulting network generalizes in that it predicts a surface normal, $[n_1, n_2, n_3]$, given any triple of input values, $[A_1, A_2, A_3]$. The resulting network trained using the calibration sphere can then be used to estimate the surface orientation of other test objects.

4.2 A 2nd Network Estimates Confidence

The RBF neural network trained as described in section 4.1 necessarily predicts a surface normal, $[n_1, n_2, n_3]$, for any input triple, $[A_1, A_2, A_3]$, whether or not that triple derives from a point on the calibration sphere (or other test object with similar reflectance properties). Thus, it is valuable to have some measure to help segment input values for which the estimated surface normal can be believed with confidence from input values for which the estimated surface normal is suspect. Nothing in the training of the network in section 4.1 enforces the constraint that the surface normal is a unit surface normal. One idea is to use the length of the surface normal, $[n_1, n_2, n_3]$, to define a confidence measure. We have found a better idea to be the simultaneous training of a second network, during calibration, to inversely predict the input, $[A_1, A_2, A_3]$ from the estimated output, $[n_1, n_2, n_3]$. Comparison between the actual input and the inversely predicted input then serves as a suitable confidence estimate.

Confidence estimation is required to separate object from background. It also is required to detect regions of cast shadow and significant interreflection. Cast shadows and interreflection do not arise with a calibration sphere or with any other single convex object. But, they do arise with multiple objects and with objects that are not convex.

The architecture for the two step RBF network is shown in Figure 2. As mentioned, both component networks are trained during calibration. During training, each normal vector input to the inverse network necessarily is a unit normal. If the 3D subspace from the calibration sphere is input to this two step network, the output of the second step, which we call the resynthesized input subspace, should be very similar to the original input. However, if an impossible triple (i.e., one that could not have arisen from the calibration sphere) is input, we expect the resynthesised input subspace to be quite different since the resynthesized values necessarily correspond to points on the calibration sphere.

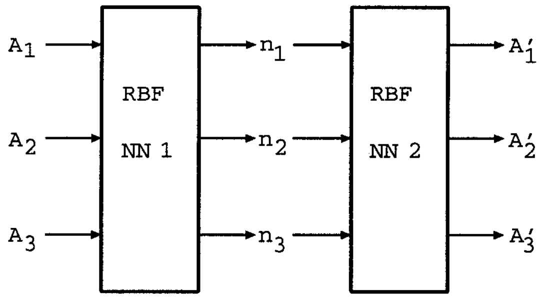


Figure 2: Two-Step RBF Neural Network

For each point on a test object, the triple $[A_1, A_2, A_3]$ is input to the first network. The length of the estimated output, $[n_1, n_2, n_3]$, is normalized to one prior to using it as input to the second network. Let $[A'_1, A'_2, A'_3]$ be the resynthesized input subspace estimated by the second network. Let

$$d = \sqrt{(A_1 - A'_1)^2 + (A_2 - A'_2)^2 + (A_3 - A'_3)^2} \quad (7)$$

The larger the value of d, the larger is the deviation of the test point from a point that could have arisen on the calibration sphere.

5 Experiments

5.1 Experimental Setting

A calibrated imaging facility (CIF) has been developed at the UBC Laboratory for Computational Intelligence (LCI) to control both scene parameters and conditions of imaging. It is based on an optical bench with mounting hardware for controlled positioning and motion of cameras, light sources and test objects. The current work uses multiple images of a scene acquired sequentially by moving a single light source. Positioning of the light source is controlled via motion stages on the optical bench.

Two objects are used in the experiments reported. One is a pottery sphere, used for calibration purposes, and the other is a pottery boy face. In this case, both objects are made of the same material with the same reflectance properties. No particular assumptions for the surface reflectance or light source directions are used or needed for the experiments. The pottery boy face was mounted on a rotational motion stage. Different test images were acquired from the same viewpoint and illumination but different rotations of the boy face. The configuration of the moving light source used in this experiment is shown in Figure 3. Seven light source directions were used, each differing from its neighbours by about 2 degrees, as shown in Figure 3.

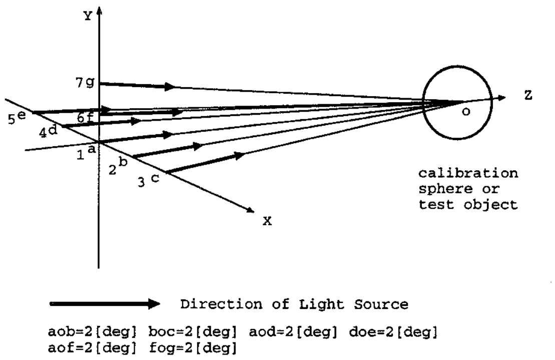

Y
7g
5e
6f
4d
1a
2b
3c
o
Z
X
calibration sphere or test object
Direction of Light Source
aob=2[deg] boc=2[deg] aod=2[deg] doe=2[deg]
aof=2[deg] fog=2[deg]

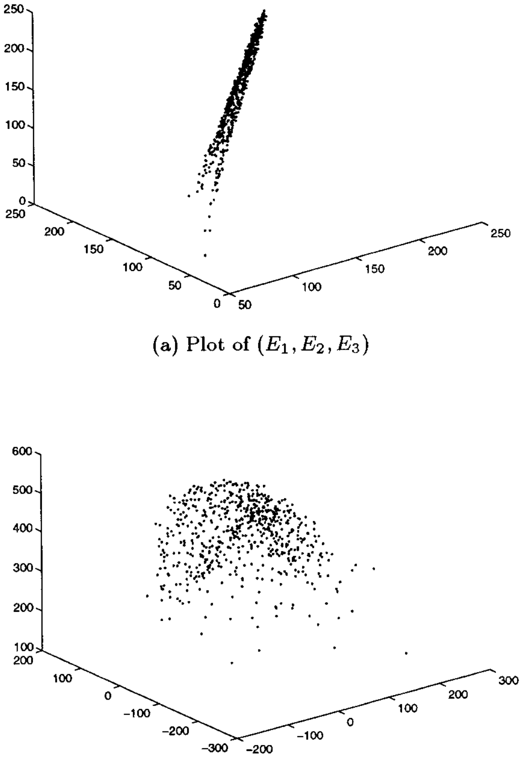
Figure 3: Configuration of Moving Light Source

5.2 Calibration

Calibration measures reflectance data using an object of known shape. A sphere is a good choice because it is convex, thus eliminating interreflection, because it is easy to dead reckon local surface orientation geometrically from the object silhouette and because it spans all possible visible surface normal vectors. Details of the calibration procedure used are described elsewhere [6, 9].

5.3 Results

For PCA, the seven 512×512 input images were subsampled every 4 pixels. In one example, this resulted in about 3800 objects points. The covariance matrix, $\mathbf{K}$, its eigenvector matrix, $\mathbf{\Phi}$, and eigenvalues, λ_k, are shown in Table 1. One can verify that the sum of the diagonal elements of $\mathbf{K}$ and $\mathbf{\Phi}$ are equal (to within round-off error). The most significant principal component accounts for about 98% of the total variance. Inspection of ϕ_1, the first column of $\mathbf{\Phi}$, shows that the corresponding eigenvector has strictly positive coefficients that are approximately equal in value. Thus, the first principal component is (approximately) the average of all the seven input images. This is a kind of signal averaging that contributes to noise reduction.



(a) Plot of (E_1, E_2, E_3)

(b) Plot of (A_1, A_2, A_3)

Figure 4: 3D plots of (E_1, E_2, E_3) and (A_1, A_2, A_3)

The second most significant principal component accounts for about 1% of the total variance. Inspection of ϕ_2, the second column of $\mathbf{\Phi}$, shows that the corresponding eigenvector has coefficients that sum approximately to zero. Thus, the second principal component is (approximately) a difference image of (weighted combinations of) the seven input images. It is hard to argue that the third (or any larger) principal component accounts for any significant proportion of the total variance. This is not unexpected, given that the overall problem has only two degrees of freedom. Nevertheless, the third principal component is retained in the subsequent analysis. Arguably, it plays the role of a kind of checksum, essential to distinguish legitimate inputs $[E_1, E_2, \ldots, E_p]$ from those either not part of the test object or affected by cast shadows or interreflection.

To further illustrate, a 3D plot of (E_1, E_2, E_3) for the calibration sphere is shown in Figure 4(a). In this figure, points are concentrated about a line demonstrating the high degree of correlation. Plots of other sub-selected triples, (E_i, E_j, E_k), give similar results. The 3D plot of the corresponding (A_1, A_2, A_3) is shown in Figure 4(b). Significant decorrelation has been achieved.

Values from the calibration sphere, (A_1, A_2, A_3), and their corresponding, dead reckoned surface normal coordinates, (n_1, n_2, n_3), were used as the inputs and outputs, respectively, for training the first neural network. The role of input and output is reversed when training the second neural network. Thus, the number of input and output units for each network is three. Overall, the training set used 600 points on the calibration sphere. Matlab RBF neural network software was used, as described in section 6 of the Matlab neural network toolbox user's guide [17]. In general, both the mapping from (A_1, A_2, A_3) to (n_1, n_2, n_3) and the inverse mapping from (n_1, n_2, n_3) to (A_1, A_2, A_3) are non-linear. For the first network, iterative learning proceeded for 50 epochs (i.e., until the number of hidden units became 50), while for the second network, iterative learning proceeded for 100 epochs. The reason for the difference relates to generalization. We are more willing for the first network to generalize to estimate a surface normal. On the other hand, given that the second network is being used as a validity check, we are less willing to have it generalize.

One needs also to consider the dynamic range of neural network inputs and outputs. For visible surface points, $-1 \leq n_i \leq 1$, $i = 1, 2$, and $0 \leq n_3 \leq 1$. With 8 bit-per-pixel input images, $0 \leq E_i \leq 255$, $i = 1, 2, 3, \ldots, 7$. The dynamic range of the corresponding A_i will, of course, be different. As mentioned above, one can differentially consider the number of bits to allocate per principal component. This was not done here. In RBF neural network learning, the spread constant of the radial basis function should be much larger than the minimum distance and much smaller than the maximum distance between input vectors. Given the expected range of values for A_i and n_i, $i = 1, 2, 3$, the spread constant for the first network was set to 150 and the spread constant for the second network was set to 1. The learning states for the two networks are shown in Figures 5 and 6, respectively. Learning was effectively achieved using the 600 points on the calibration sphere.

Once neural network learning is complete, the resulting networks can be applied to the test object images. Two examples are shown, called boy-1 and boy-2. For boy-1, the face is oriented directly towards the viewer. Figure 7-(a) shows one of the seven input images. Figures 7-(b) and 7-(c) show the estimated surface orientation, represented as slope and aspect. Figure 7-(b) linearly encodes the slope angle, e, (i.e., the angle between the surface normal and the vector pointing to the viewer) as a gray value in the range black ($e = 0$) to white ($e = \pi/2$). Figure 7-(c) plots the aspect angle (i.e., the projection of the surface normal onto the XY plane) as a short line segment. (To avoid clutter, the aspect angle is plotted for every fourth point in x and y.) Figure 7-(d) shows the value of the confidence estimate d, encoded as a gray level. Bright values corresponds to points in the image where confidence in the estimated surface orientation is lowest. As can be seen, bright points correspond to areas, near occluding boundaries, affected by shadow and to locally concave areas affected by interreflection. Figures 8 shows results for the boy-2 example. Boy-2 is the same object rotated with respect to the boy-1 example.

No quantitative analysis of the results has been

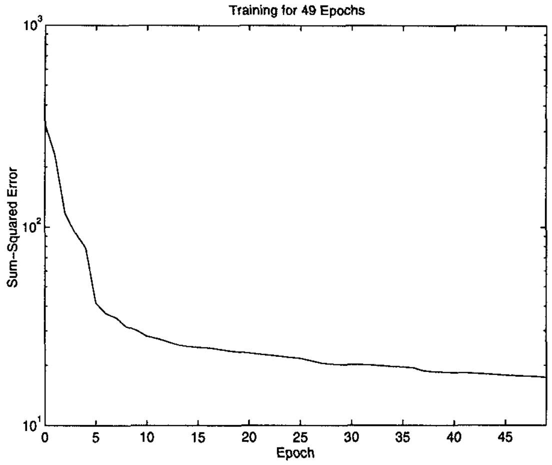


Figure 5: Learning 1: (A_1, A_2, A_3) to (n_1, n_2, n_3)

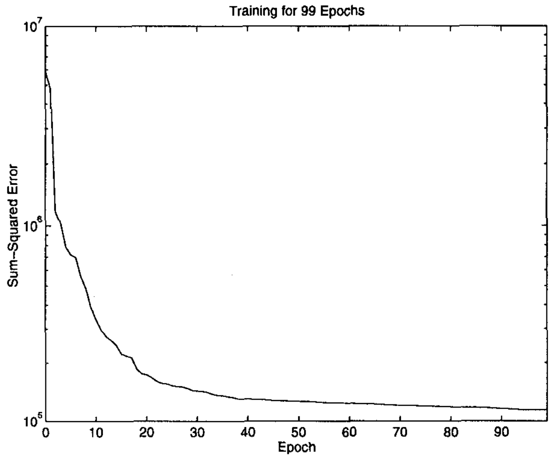


Figure 6: Learning 2 for (n_1, n_2, n_3) to (A_1, A_2, A_3)

performed since the true shape of the boy face is not known. Both the boy-1 and boy-2 results are qualitatively correct, except in areas where the value of the corresponding d is large. The results are of comparable accuracy, in areas of common coverage, to other implementations of photometric stereo. One advantage of the current approach is clear. Because the illumination directions all are nearly collinear with the viewing direction, there are relatively few areas where the estimation of surface normal is affected by shadow. Interreflection and shadows near occluding boundaries do lead to large values of d. In these cases, the first neural network cannot be expected to be correct because it is encountering input values well outside the space of inputs covered in training. The second network thus provides an essential check on the confidence to attach to the estimated surface normal. Conversely, when d is small, there is reason to be confident in the estimate of surface orientation.

It might be argued that PCA would not needed if the neural network was given all seven image irradiances, $(E_1, E_2, \ldots, E_7)$, as input. It would then be the network's job to learn, via training, whatever signal averaging and decorrelation is appropriate. In another experiment, direct learning of the mapping of $(E_1, E_2, \ldots, E_7)$ to (n_1, n_2, n_3) was explored, again using data from the calibration sphere. Experimentation was not comprehensive but did confirm expectations. The results, in terms of complexity of the network, time to learn and accuracy of the result, all were significantly poorer. It does seem that PCA is an effective preprocessing step prior to neural network training.

6 Conclusion

A new approach to implementation of photometric stereo has been described. Multiple images are acquired using several, nearby positions of a single illuminant. In the configuration developed, all directions of illumination are close to the viewing direction and thus are nearly collinear. This has the potential advantage of extending the domain of practical application of photometric stereo, especially to tasks where the entire sensing system, including light sources, needs to be as physically compact as possible,

The usefulness of more than the theoretical minimum number of images has been demonstrated. Given a light source moving in a (periodic) trajectory, it would be possible to exploit many more than $p = 7$ images. Principal components analysis (PCA) is an effective linear preprocessing technique to extract a reduced dimensionality subspace prior to non-linear functional approximation via a neural network. The particular implementation described uses a radial basis function (RBF) neural network and the orthogonal least squares (OLS) learning method. This was sufficient both to learn and to represent the mapping between input subspace and surface normal.

The implementation continues to exploit the ability of photometric stereo to overdetermine the solution locally. This is achieved by training two neural networks. The first predicts the intended output (surface normal) from the given input. The second inversely predicts the input from the estimated output (surface normal). Comparison between the real input and the inversely predicted input is used as the confidence estimate. The local confidence estimate helps to detect regions of cast shadow and significant interreflection. At the moment, the possibility of spatially varying albedo is not considered. There is no reason, in principle, not to include spatially varying albedo in the problem formulation. This creates new problems for calibration. On the other hand, if some assumption is made about the form of the object's bidirectional reflectance distribution function (BRDF), for example that it is spectrally separable, then the possibility exists to synthesize the effect of spatially varying albedo within the current method of calibration.

Finally, it is valuable to stress that the entire approach is empirical in that no explicit assumptions are made about light source directions or surface reflectance. It is sufficient that the calibration sphere used in training and the subsequent test objects be viewed under the same pattern of illumination and be made of the same material (i.e., have the same reflectance properties). Both principal components analysis and neural network functional approximation are used here as non-parametric, empirical techniques.

Acknowledgement

The research described in this paper was performed while Iwahori was visiting UBC with the support of Japanese Ministry of Education. Major support for Woodham's research is provided by the Institute for Robotics and Intelligent Systems (IRIS), one of the Canadian Networks of Centres of Excellence, by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Canadian Institute for Advanced Research (CIAR). The authors thank members of the UBC Laboratory for Computational Intelligence, Dr. H. Tanaka of the Toyota College of Technology and Prof. N. Ishii of the Nagoya Institute of Technology for many useful discussions.

References

[1] B. K. P. Horn, "Obtaining shape from shading information," in *The Psychology of Computer Vision* (P. H. Winston, ed.), pp. 115–155, McGraw-Hill, 1975.

[2] B. K. P. Horn, "Understanding image intensities," *Artificial Intelligence*, vol. 8, pp. 201–231, 1977.

[3] R. J. Woodham, "Photometric method for determining surface orientation from multiple images," *Optical Engineering*, vol. 19, pp. 139–144, 1980.

[4] G. Healey, S. Shafer, and L. Wolff, eds., *Physics-Based Vision: Principles and Practice (Vol. 1 Radiometry, Vol. 2 Color and Vol. 3 Shape Recovery)*. Boston, MA: Jones and Bartlett Publishers, Inc., 1992.

[5] G. Healey and R. Jain, "Physics-based machine vision," *Journal of the Optical Society of America, A*, vol. 11, p. 2922, 1994. (Introduction to special issue).

[6] R. J. Woodham, "Gradient and curvature from the photometric stereo method, including local confidence estimation," *Journal of the Optical Society of America, A*, vol. 11, pp. 3050–3068, 1994.

[7] Y. Iwahori, N. Hiratsuka, H. Kamei, and S. Yamaguchi, "Extended photometric stereo for an object with unknown reflectance property," *Trans. IEICE*, vol. J71-D, no. 1, pp. 110–117, 1988. (in Japanese).

[8] Y. Iwahori, H. Tanaka, R. J. Woodham, and N. Ishii, "Photometric stereo for specular surface shape based on neural network," *IEICE Transactions on Information and Systems*, vol. E77-D, no. 4, pp. 498–506, 1994. (Special issue on neurocomputing).

[9] Y. Iwahori, A. Bagheri, and R. Woodham, "Neural network implementation of photometric stereo," in *Vision Interface 95*, (Quebec City, Canada), 1995. (in press).

[10] L. B. Wolff, "Shape understanding from Lambertian photometric flow fields," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1989*, pp. 46–52, June 1989.

[11] H. Tanaka, Y. Iwahori, and N. Ishii, "Uni-directional photometric flow field under illuminating directions with slightly varied azimuth angles," *Trans. IEICE*, vol. J74-D-II, no. 11, pp. 1538–1545, 1991. (in Japanese).

[12] Y. Iwahori, R. J. Woodham, H. Tanaka, and N. Ishii, "Shape from shading with a nearby moving point light source," in *Proc. 2nd International Conference on Automation, Robotics and Computer Vision*, (Singapore), pp. CV–5.5.1–CV–5.5.5, 1992.

[13] H. Tanaka, Y. Iwahori, and N. Ishii, "Determining surface orientation from directional derivatives of shading images by slightly varied illuminating direction," *Trans. IEICE*, vol. J76-D-II, no. 5, pp. 1076–1079, 1993. (in Japanese).

[14] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.

[15] M. J. D. Powell, "Radial basis functions for multivariable interpolation: a review," in *Algorithms for Approximation* (J. C. Mason and M. G. Cox, eds.), pp. 143–167, Oxford University Press, 1987.

[16] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

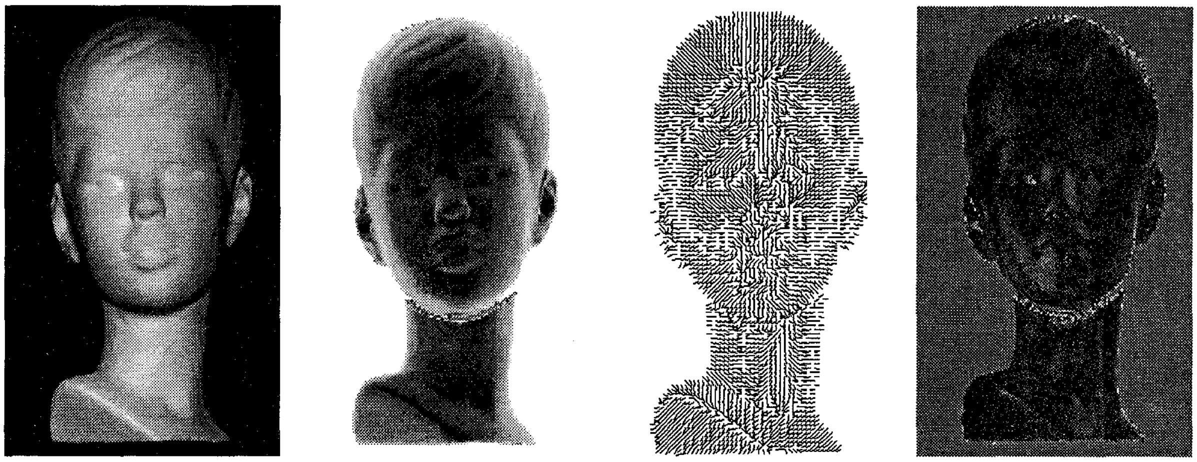
[17] H. Demuth and M. Beale, *Neural Network Toolbox User's Guide*. Natick, MA: The MathWorks Inc., 1994.

$$\mathbf{K} = 10^3 \times \begin{bmatrix} 1.5058 & 1.4394 & 1.4006 & 1.4009 & 1.4003 & 1.4372 & 1.4288 \\ 1.4394 & 1.3937 & 1.3712 & 1.3337 & 1.3224 & 1.3795 & 1.3731 \\ 1.4006 & 1.3712 & 1.3687 & 1.2887 & 1.2664 & 1.3462 & 1.3423 \\ 1.4009 & 1.3337 & 1.2887 & 1.3209 & 1.3303 & 1.3412 & 1.3329 \\ 1.4003 & 1.3224 & 1.2664 & 1.3303 & 1.3531 & 1.3403 & 1.3336 \\ 1.4372 & 1.3795 & 1.3462 & 1.3412 & 1.3403 & 1.3862 & 1.3874 \\ 1.4288 & 1.3731 & 1.3423 & 1.3329 & 1.3336 & 1.3874 & 1.4073 \end{bmatrix}$$

$$\mathbf{\Phi} = \begin{bmatrix} 0.3957 & -0.0392 & 0.2414 & 0.7499 & 0.3628 & -0.2608 & 0.1470 \\ 0.3799 & 0.2899 & 0.2344 & 0.0893 & -0.0331 & 0.6855 & -0.4875 \\ 0.3709 & 0.6472 & 0.2223 & -0.4604 & 0.1362 & -0.3658 & 0.1728 \\ 0.3694 & -0.3248 & 0.2213 & -0.1585 & -0.2822 & 0.3555 & 0.6913 \\ 0.3693 & -0.6235 & 0.1592 & -0.4020 & 0.2612 & -0.2233 & -0.4120 \\ 0.3801 & 0.0056 & -0.2307 & 0.1699 & -0.7745 & -0.3521 & -0.2228 \\ 0.3796 & 0.0355 & -0.8426 & -0.0459 & 0.3179 & 0.1671 & 0.1168 \end{bmatrix}$$

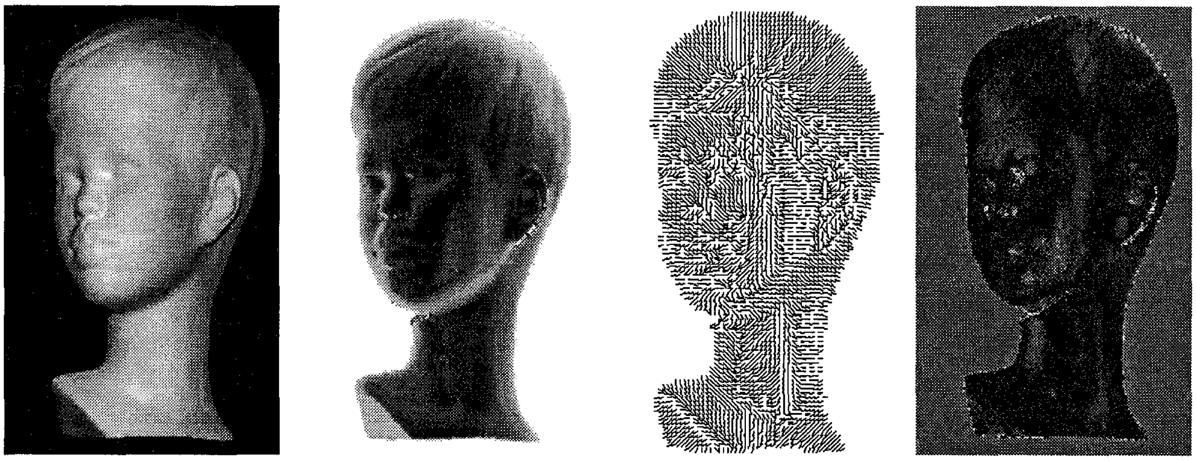
$$[\lambda_1 \ldots \lambda_7] = 10^3 \times [\ 9.5662 \quad 0.1166 \quad 0.0398 \quad 0.0084 \quad 0.0024 \quad 0.0012 \quad 0.0009\ ]$$

Table 1: Example covariance matrix, eigenvector matrix and eigenvalues



(a) Boy-1 sample image (b) Boy-1 slope (c) Boy-1 aspect (d) Boy-1 confidence

Figure 7: Boy-1 example



(a) Boy-2 sample image (b) Boy-2 slope (c) Boy-2 aspect (d) Boy-2 confidence

Figure 8: Boy-2 example