Lighting Direction Estimation of a Shaded Image
by a Surface-input Regression Network

Chi Kin CHOW and Shiu Yin YUEN

Abstract–**In augmented reality (AR), the lighting direction plays an important role to the quality of the augmented scene. The corresponding lighting direction estimation is a challenging problem as it depends on an extra unknown variable – reflectance of the material. In this article, we propose to estimate the lighting direction by a neural network (NN) which is trained by a sample set. Since the empirical reflectance of a captured scene is in form of scattered points, we unify the representation of reflectance as a two dimensional polynomials. Moreover, a novel neural network model is presented to construct the mapping from reflectance to lighting direction. Contrary to the existing NNs, the proposed model accepts surface input pattern in which the drawbacks of feature vector are overcome. Experimental results of 2000 lighting estimations with unknown reflectances are presented to demonstrate the performance of the proposed algorithm.**

I. INTRODUCTION

Augmented reality (AR) is a field of computer research that deals with the combination of real world and computer generated data. At present, most AR researches are concerned with the use of live video imagery which is digitally processed and "augmented" by the addition of computer generated graphics. The visual quality of a AR scene is highly sensitive to the lighting consistency between the real and the virtual scenes. Recently, the lighting direction is recovered by a simple camera calibration under the controlled environment. In more complicated cases, the direction is assigned manually (by trial and error) and adjusted until the visual quality of the rendering sence is satisfactory to a human judge. In computer vision, the lighting direction is guessed though clues like shading and shadows. However, since the rendering model is different from image to image, the estimation process is complicated and time consuming.

A useful discussion of the ambiguities involved in lighting direction estimation can be found in [1]. Several of researchers have proposed methods for the estimation of the light source direction. Horn and Brooks [2] propose an iterative method that updates both the shape and the

illuminant direction at every iteration. To avoid local minima, a good initial state is necessary, and furthermore, the requirement for a light source vector of unit length is not enforced. Lee and Rosenfeld [3] used a Gaussian sphere model for the surface normal distribution and local spherical patches, but did not take shadowing effects into account. Zheng and Chellappa [4] present a more sophisticated image based method, considering shadowing effects and using a uniform distribution of the tilt and slant angles of surface normals. They still assume local spherical patches and their algorithm suffers on surfaces that deviate significantly from this assumption. Leclerc and Bobick [5] derive accurate light source information from surfaces reconstructed using stereo data.

Dimitris et al. in [6] proposed an iterative algorithm to estimate the surface and lighting direction of a shaded image. They employed the method of Zheng and Chellappa [4] to initialize the lighting direction. The initialized direction is then used to estimate the model parameters (surface). Afterward, the lighting direction is re-estimated by the feedback of surface obtained in the pervious step. The iteration is repeated until the estimated lighting direction and the model parameters mutually agreed. The accuracy of the estimated lighting direction highly depends on the initiate. Furthermore, the convergence proof of the algorithm is absent that a converged solution is not guaranteed.

Cho and Chow in [7] reported a neural computation approach to recover the reflectance model and 3D shape of a shaded image. Similar to [6], the model of Cho and Chow is an iterative approach; the estimated reflectance model and 3D shape are mutually updated by each another until the proposed stopping criterion is reached. This work suffers from the same convergence problem above. In addition, the network topology of the correct reflectance model is difficult to determine and is a challenging problem.

Reflectance $r(.)$ in computer graphics represents the mapping from a set of rendering parameters (i.e. lighting direction **L**, surface unit normal $\mathbf{N} = [N_x, N_y, N_z]$ and material **K**) of a point to the corresponding intensity I, i.e. $I = r(\mathbf{L}, \mathbf{N}, \mathbf{K})$. A constant reflectance refers to an reflectance with constant **L** and **K** at any point of a captured scene and the shading equation is simplified as $I = r(\mathbf{N})$. In the field of augmented reality, the constant reflectance can be recovered by a depth model and the corresponding shaded image. We denote this reflectance as *empirical reflectance* since it is obtained by reversing the rendering process. It is

Chi Kin CHOW and Shiu Yin YUEN are with the department of electronic engineering, City University of Hong Kong, Hong Kong, PRC (tel: 852-27887717; e-mail: {chowchi, kelviny.ee}@cityu.edu.hk).

pointed out in [7] that reflectance $r(.)$ somehow reflects the lighting direction of a captured scene Ω. Thus, we propose to estimate the lighting direction of a scene by a NN. We treat reflectance as input patterns while the corresponding lighting directions are the output patterns. Notice that the empirical reflectance is represented as a point set $\{\mathbf{N}_i \mid I_i\}$ which is an invalid input pattern of the existing NNs, we propose to model the empirical reflectance as a polynomial, namely *polynomial reflectance*, in order to unify the representation of reflectance.

The rest of this article is organized as follows: Section II discusses the reflectance map and its polynomial estimation. In Section III, we propose a surface input neural network module is that acts as the lighting direction estimator. Section IV demonstrates the performance of the proposed algorithm on give exact number thousands of lighting direction estimations. A conclusion is drawn at Section V.

II. REFLECTANCE MAP ESTIMATION

In computer graphics and computer vision, the reflectance map is a shading model that assigns shades to surface points based on the lighting direction(s), surface normal and material. Recently, several generalized reflectance models are proposed: Examples include the Phong reflection model [8], Lambertian model [9], Blinn Phong model [10], Neumann model [11] and Ashikhmin model [12]. These models are represented as linear combination of ambient, specular and diffuse reflectance. Though their formulations are empirical, these models are extensively employed in scene rendering.

In augmented reality, the constant reflectance of a given scene is estimated for the addition of computer generated graphics. In order to reduce the input dimension of constant reflectance $I = r(\mathbf{N})$, $\mathbf{N}$ is converted to $\mathbf{D} = (\alpha, \beta)$ in polar corordinate system such that $\mathbf{N} = [\cos\alpha\cos\beta, \sin\alpha\cos\beta, \sin\beta]$. Suppose $Z(u, v)$ and $I(u, v)$ are the depth map and image of a captured scene Ω, the estimated (empirical) reflectance of Ω is obtain by the procedure summarized in the following: Firstly, we compute the surface gradient $\mathbf{N}(u, v) = [-Z_u(u, v), Z_v(u, v), 1]$ at every (u, v) where

$$Z_u(u,v) = \frac{\partial Z(u,v)}{\partial u} \quad \text{and} \quad Z_v(u,v) = \frac{\partial Z(u,v)}{\partial v}$$

Afterward, $\mathbf{N}(u, v)$ is converted to a polar coordinate $\mathbf{D}(u, v) = [Z_\alpha(u, v), Z_\beta(u, v)]$ where

$$Z_\alpha(u,v) = \frac{2}{\pi}\tan^{-1}\left(\frac{1}{\sqrt{Z_u(u,v)^2 + Z_v(u,v)^2}}\right) \in [0,1] \qquad (1)$$

and

$$Z_\beta(u,v) = \frac{1}{2\pi}\tan^{-1}\left(\frac{Z_v(u,v)}{Z_u(u,v)}\right) \in [0,1] \qquad (2)$$

As a result, the empirical reflectance of Ω is described by a set of scattered reflectance samples $\mathbf{S} = \{[Z_\alpha(u, v), Z_\beta(u, v), I(u, v)]\}$. Fig. 1 shows the empirical reflectances of the first four depth models under the same shading equation. Notice

that the reflectance sample sets of the four models lie on a same surface (the actual reflectance). It can be concluded that the empirical reflectance is invariant to depth model shape.

In this article, we represent the empirical reflectance $r(\alpha, \beta)$ as a two dimensional polynomial surface; namely *polynomial reflectance*:

$$r^{(t)}(\alpha, \beta) = \sum_i w_i \alpha^{p_i} \beta^{q_i}$$

where w_i is the coefficient of the ith term; p_i and q_i are the integer powers of α and β of the ith term respectively. The superscript t denotes the maximum order of a polynomial reflectance, i.e. $t = \max_i p_i + q_i$, the power set $\mathbf{P} = \{p_i, q_i\}$ of $r^{(t)}(\alpha, \beta)$ is defined as:

$$\{p_i, q_i\} = \{x, 0 \le y \le t - x\} \text{ for } x \in [0, t] \qquad (3)$$

For example, the polynomial reflectances with the maximum orders 2 and 3 are defined as:

$$w_1\alpha^2 + w_2\beta^2 + w_3\alpha\beta + w_4\alpha + w_5\beta + w_6$$

and

$$w_1\alpha^3 + w_2\beta^3 + w_3\alpha^2\beta + w_4\alpha\beta^2 + w_5\alpha^2 + w_6\beta^2 + w_7\alpha\beta + w_8\alpha + w_9\beta + w_{10}$$

respectively. The number of terms of a polynomial reflectance $r^{(t)}(\alpha, \beta)$ is $0.5(t + 1)(t + 2)$. In addition, we define a polynomial reflectance set $\mathbf{R} = \{r^{(t)}(\alpha, \beta) = \sum_i w_i \alpha^p \beta^q\}$ where $w_i \in \Re$; $t \in Z^+$; $\{p, q\}$ satisfy eq. (3) for $t \in Z^+$. Given a set of reflectance samples $\mathbf{S} = \{\mathbf{Q}_j = [\alpha_j, \beta_j] \mid I_j\}$, the coefficients $\{w_i\}$ of $r^{(t)}(\alpha, \beta)$ are computed by the pseudo inverse method.

III. SURFACE INPUT NETWORK

Reflectance is a shape invariant shading information which relates to the lighting direction of a scene. We propose to achieve the inverse relation from reflectance to lighting direction by means of neural network (NN). Owing to the simplification of the training process, input patterns of the existing NN modules are commonly represented as vectors after feature extraction. In the view of lighting direction estimation, the coefficient set $\{w_i\}$ of a polynomial reflectance is possible to be a feature vector of NNs. However, as the influence of coefficient to the reflectance is different from one to another, i.e. $\partial r^{(t)}(\alpha, \beta)/\partial w_a \gg \partial r^{(t)}(\alpha, \beta)/\partial w_b$ if $c + d \gg e + f$ where w_a and w_b are the coefficients of the terms $\alpha^c\beta^d$ and $\alpha^e\beta^f$ respectively, the corresponding feature domain is sensitive to noise. Driven by the drawbacks of vectoral input pattern, we propose a novel neural network module called Surface Input Network (SIN) which constructs a mapping from surface input (polynomial reflectance) to vector output (lighting direction). This feature leads to a more realistic description of the relation between reflectance and lighting direction, through which more accurate estimation can be expected.

A. Reflectance Similarity

Similarity plays an important role in training and prediction of neural NN. For example, the self-organizing map uses the Euclidean distance between neuron and input pattern to conclude the winning node. Support vector machine uses the Euclidean distance among kernels to formulate the energy function. In this section, a measure of similarity between reflectances, namely *reflectance distance*, is introduced.

Suppose $r_1^{(t)}(\alpha, \beta)$ and $r_2^{(u)}(\alpha, \beta)$ are two polynomial reflectances where

$$r_1^{(t)}(\alpha, \beta) = \sum\nolimits_{i=1}^{n_1} w_{1,i} \alpha^{p_{1,i}} \beta^{q_{1,i}}$$

and

$$r_2^{(u)}(\alpha, \beta) = \sum\nolimits_{i=1}^{n_2} w_{2,i} \alpha^{p_{2,i}} \beta^{q_{2,i}}$$

we denote by a notation of the reflectance distance between $r_1^{(t)}(\alpha, \beta)$ and $r_2^{(u)}(\alpha, \beta)$ as $A(r_1^{(t)}(\alpha, \beta), r_2^{(u)}(\alpha, \beta))$:

$$A(r_1^{(t)}(\alpha,\beta), r_2^{(u)}(\alpha,\beta)) \stackrel{def}{=} \int_0^1\int_0^1 \left(r_1^{(t)}(\alpha,\beta) - r_2^{(u)}(\alpha,\beta) \right)^2 d\alpha\, d\beta$$

$$(4)$$

which is an integration of the squared differences between $r_1^{(t)}(\alpha, \beta)$ and $r_2^{(u)}(\alpha, \beta)$. As we represent any reflectance as a two dimensional polynomial, $A(r_1^{(t)}(\alpha, \beta), r_2^{(u)}(\alpha, \beta))$ can be further rewritten as:

$$= \int_0^1\int_0^1 \left(\sum\nolimits_{i=1}^{n_1} w_{1,i} \alpha^{p_{1,i}} \beta^{q_{1,i}} - \sum\nolimits_{j=1}^{n_2} w_{2,j} \alpha^{p_{2,j}} \beta^{q_{2,j}} \right)^2 d\alpha\, d\beta$$

$$= \sum\nolimits_{i=1}^{n_1}\sum\nolimits_{j=1}^{n_1} \frac{w_{1,i} w_{1,j}}{(p_{1,i} + p_{1,j} + 1)(q_{1,i} + q_{1,j} + 1)} -$$

$$\sum\nolimits_{i=1}^{n_1}\sum\nolimits_{j=1}^{n_2} \frac{2 w_{1,i} w_{2,j}}{(p_{1,i} + p_{2,j} + 1)(q_{1,i} + q_{2,j} + 1)} +$$

$$\sum\nolimits_{i=1}^{n_2}\sum\nolimits_{j=1}^{n_2} \frac{w_{2,i} w_{2,j}}{(p_{2,i} + p_{2,j} + 1)(q_{2,i} + q_{2,j} + 1)}$$

$$(5)$$

B. Structure of SIN

The proposed estimator $F(\cdot)$ describes the mapping from a polynomial reflectance $r^{(t)}(\alpha, \beta) \in \mathbf{R}$ to the corresponding lighting direction $\mathbf{L} = [\theta_1, \theta_2]$, i.e. $\mathbf{L} = F(r^{(t)}(\alpha, \beta))$ by means of the Gaussian mixture model (GMM):

$$\mathbf{L} = \sum\nolimits_k \mathbf{m_k} \exp\left(-\frac{A(r^{(t)}(\alpha,\beta), \mu_k(\alpha,\beta))}{2\sigma_k^2} \right) + \mathbf{B}$$

where $\mathbf{B}$ is the bias vector; $\mathbf{m_k} = [m_{k,1}, m_{k,2}]$, $\mu_k(\alpha, \beta) \in \mathbf{R}$ and σ_k^2 are the weight vector, reflectance mean and variance of the kth kernel respectively. Initially, n Gaussian kernels are assigned for a SIN with a training set $\mathbf{T} = \{r_k^{t(k)}(\alpha, \beta) \mid \mathbf{L_k} = [\theta_{1,k}, \theta_{2,k}]\}_{k \in [1,n]}$ consisting of n samples. The number of kernels is pruned iteratively under the training error reaches the given threshold ε_T.

C. Training algorithm of SIN

The proposed training algorithm is an iterative method. The size of training set, namely contributed training set $\mathbf{T'} = \{r_j^{t(i)}(\alpha, \beta)' \in \mathbf{R} \mid \mathbf{L_j'} = [\theta_{1,j}, \theta_{2,j}]\} \subset \mathbf{T}$, is reduced by one after each iteration. The remaining training pattern in $\mathbf{T'}$ are regarded as the most contributed samples for the network. During the training phase, five types of parameters of SIN: 1) the number of kernels, 2) means, 3) variances, 4) bias vector and 5) weight vectors have to be determined.

1. The number of kernel is equivalent to the size of $\mathbf{T'}$.

2. Mean $\mu_j(\alpha, \beta)$: Each training input pattern in $\mathbf{T'}$ is regarded as the reflectance mean of the SIN, i.e. $\mu_j(\alpha, \beta) \leftarrow r_j^{t(i)}(\alpha, \beta)'$.

3. Variance σ_j^2: It is a measure of the averaged reflectance distance of an input reflectance from a reflectance mean. The value of σ_j^2 is suggested as:

$$\sigma_j^2 = \frac{4\log 0.8}{\min\limits_{i<j} A(r_j^{t(k)}(\alpha,\beta)', r_i^{t(i)}(\alpha,\beta)')}$$

4. Bias vector $\mathbf{B} = [b_1, b_2]$: It denotes as the averaged lighting direction, i.e. $\mathbf{B} = \sum_k \mathbf{L_k} / n$.

5. Weight vectors $\{\mathbf{m_j}\}$: They are computed by minimizing the least square error of the estimated $\mathbf{L}$ with respect to $\{\mathbf{m_j}\}$. This process can be done by a pseudo inversion calculation method.

The training process is terminated at the tth iteration if the least square error in this iteration reaches ε_T. Otherwise, $\mathbf{T'}$ is updated by the following procedures:

1. The contribution C_j of the kernel $[\mu_j(\alpha, \beta), \sigma_j^2, \mathbf{m_j}]$ is defined as the maximum of its hyper-volumes in θ_1 and θ_2 domains, i.e. $C_j = max(m_{j,1}\sigma_j(2\pi)^{0.5}, m_{j,2}\sigma_j(2\pi)^{0.5})$.

2. The training pattern $[r_u^{t(u)}(\alpha, \beta)' \mid \mathbf{L_u'}] \in \mathbf{T'}$ is removed from $\mathbf{T'}$ where $u = arg\ min\{C_j\}$.

3. Retrain the SIN with the updated $\mathbf{T'}$.

IV. EXPERIMENTAL RESULTS

In this experiment, we apply SIN to estimate the lighting direction $\mathbf{L} = [\theta_1, \theta_2]$ of a captured scene Ω based on the shaded image $I(u, v)$ and the depth map $Z(u, v)$ of Ω. SIN is trained by a set of synthetic reflectance samples for which the shading equations are known. In the actual implementation, the shaded images are captured from scenes instead of by synthesis.

Six depth maps: Mozart, 4 Mountains, Vase, Sphere, Skull and Heart shown in Fig. 2 are used to train and evaluate the proposed estimator. The first four models had appeared in [14] and the remaining two models can be found in [15]. The shading equations in this experiment are based on the Phong reflection model [8].

$$\text{Phong model}: P(\mathbf{N}, \mathbf{L}) = k_a + k_d\,(\mathbf{N} \cdot \mathbf{L}) + k_s\,(\mathbf{V} \cdot \mathbf{F})^{n_s}$$

where $\mathbf{V} = [0, 0, 1]$ and $\mathbf{F} = 2\mathbf{N}(\mathbf{N} \cdot \mathbf{L}) - \mathbf{L}$. The values of $k_a \in [0, 0.05]$, $k_d \in [0.05, 0.5]$, $k_s \in [0.05, 0.5]$ and $n_s \in [2, 6]$ are chosen randomly under the uniform distribution. Totally 40 patterns are used in training phase. In order to illustrate the generalization of the estimator, different sets of depth models are used in training and evaluation phases. The first three models (Mozart, 4 Mountains and Vase) form the depth map set $\{Z^t_i(u, v)\}_{i=1,2,3}$ for the construction of training set $\mathbf{T} = \{r_k^{t(k)}(\alpha, \beta) \mid \mathbf{L_k}\}$ while the last three models (Sphere, Skull and Heart) form the depth map set $\{Z^e_i(u, v)\}_{i=1,2,3}$ for the evaluation. Although the reflectance depends only on the surface normal, the distribution of $[Z_\alpha(u, v), Z_\beta(u, v)]$ of $\mathbf{S}$ is varied from depth map to depth map such that the corresponding polynomial reflectance piece-wisely agrees with the actual one. This is equivalent to introduce noise to training set of SIN. The training error threshold ε_T is chosen as 0.01.

The performance of the estimator is compared with two methods:

1) Support vector machine (SVM) [13] – a state of the art vector-input NN: The coefficients $\{w_i\}$ of a polynomial reflectance $r^{(t)}(\alpha, \beta)$ form the corresponding feature vector $\mathbf{f}$, i.e. $\mathbf{f} = [w_1, w_2, \ldots, w_{0.5(t+1)(t+2)}]$.

2) Nearest neighbor scheme (NNS): Given the sample set $\mathbf{T}$, the output of an reflectance $r^{(t)}(\alpha, \beta)$ under the nearest neighbor scheme is estimated as $\mathbf{L_c}$ where the subscript c is defined as:

$$c = \arg \min_k R\left(r^{(t)}(\alpha, \beta), r_k^{t(k)}(\alpha, \beta)\right)$$

and $r_k^{t(k)}(\alpha, \beta) \in \mathbf{R}$.

The number of kernels of trained SIN and SVM are 20 and 36 respectively. Empirically, the computational load of SIN is much lower (only 55%) than that of SVM as it involves less reflectance distance calculation. By comparing with the NNS, the reduced computational load is more significant since more than 30% of training patterns can be ignored in the network construction.

In the evaluation phase, the lighting directions of 2000 testing patterns are estimated by the proposed estimator, SVM and the nearest neighbor scheme. The generation procedures of the testing patterns follow those of the training patterns, except that the depth map set $\{Z^t_i(u, v)\}$ is replaced with $\{Z^e_i(u, v)\}$. Fig. 3(a) and Fig. 3(b) show the sorted errors of the estimated θ_1 and θ_2 respectively. Fig. 4 shows the angles θ_e between the actual lighting directions $\mathbf{L_a}$ and the estimated lighting directions $\mathbf{L_e}$, i.e. $\theta_e = \cos^{-1}(\mathbf{L_a} \cdot \mathbf{L_e} / |\mathbf{L_a}| \cdot |\mathbf{L_e}|)$. The solid lines represent the results of SIN. The dotted lines represent the results of SVM and the dashed lines represent the results of the nearest neighbor scheme. Table I and Table II list the statistics of the absolute errors of θ_1 and θ_2. Table 3 lists the statistic of θ_e.

Seen from the table, SIN is able to estimate the mapping from a set of piecewise correct information to θ_1 and θ_2. The estimated θ_1 and θ_2 are more accurate than those obtained by SVM and the nearest neighbor scheme in both mean and median measures. Furthermore, the inaccurate results of the nearest neighbor scheme indicate that the relation between polynomial reflectance and lighting direction is complicated. In addition of the huge estimation errors of SVM, the light estimation by SIN is necessary.

TABLE 1
STATISTICS OF THE ESTIMATION ERRORS OF θ_1

	Mean	Std.	Median
SIN	2.196	2.959	1.279
SVM	38.432	44.189	24.041
NNS	14.027	10.771	11.034

TABLE 2
STATISTICS OF THE ESTIMATION ERRORS OF θ_2

	Mean	Std.	Median
SIN	20.756	22.127	14.311
SVM	29.159	161.81	57.3
NNS	73.536	44.835	78..714

TABLE 3
STATISTICS OF THE θ_E

	Mean	Std.	Median
SIN	19.752	21.746	13.137
SVM	90.089	36.425	90.405
NNS	68.929	37.233	71.402

V. CONCLUSION

Lighting direction is an essential information piece of in computer graphics, especially in augmented reality. A high quality augmented scene is dependent on an accurate lighting direction. The corresponding lighting direction estimation is considered as an information retrieval from the empirical reflectance of a captured scene. In this article, we propose estimating the lighting direction by means of neural network (NN). The promising results support two contributions of this article. Firstly, the empirical reflectance is parameterized as a two dimensional polynomial for the training. Since the empirical reflectance is in form of scattered reflectance samples, in addition to the variable sizes of the sample sets, the empirical reflectance is an invalid input pattern of NN. The proposed polynomial reflectance unifies the representations of reflectances such that the learning based estimation is possible. Secondly, the proposed surface input network (SIN) overcomes the drawback of non-uniform coefficient influences of polynomial reflectance. The SIN considers polynomial reflectances as input patterns, by which a more realistic description of the relation between reflectance and lighting direction is achieved. It is experimentally verified that a more accurate estimation is achieved compared with the support vector machine and the nearest neighbor scheme.

REFERENCES

[1] P. N. Belhumeur and D. J. Kriegman, "What is the Set of Images of an Object under all Possible Lighting Conditions", *IEEE Computer Vision and Pattern Recognition*, pp. 270-277, 1996.

[2] B. K. P. Horn and M. J. Brooks, "Shape and Source from Shading", *Int'l Joint Conf. Artificial Intelligence*, pp. 932-936, 1985.

[3] C. H. Lee and A. Rosenfeld, "Improved Methods of Estimating Shape from Shading Using the Light Source Coordinate System", *Artificial Intelligence*, vol. 26, no. 2, pp. 125-143, May 1985.

[4] Q. Zheng and R. Chellappa, "Estimation of Illuminant Direction, Albedo, and Shape from Shading", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 680-702, July 1991.

[5] Y. G. Leclerc and A. F. Bobick, "The Direct Computation of Height from Shading", *IEEE Computer Vision and Pattern Recognition*, pp. 552-558, 1991.

[6] Dimitris Samaras and Dimitris Metaxas, "Incorporating Illumination Constraints in Deformable Models for Shape from Shading and Light Direction Estimation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 247 – 264, 2003.

[7] Siu-Yueng Cho and T. W. S. Chow, "Neural computation approach for developing a 3D shape reconstruction model", *IEEE Transactions on Neural Networks*, vol. 12, Issue 5, pp. 1204 – 1214, Sept. 2001.

[8] Bui Tuong Phong, "Illumination for Computer Generated Pictures", *University of Utah Ph.D. dissertation*, 1973

[9] J. Lambert, "Photometria Sive de Mensura et Gradibus Luminus, Colorum et Umbrae" , *Eberhard Klett*, 1760

[10] Jim F., "Models of light reflection for computer synthesized pictures", In *Computer Graphics (Proceedings of ACM SIGGRAPH 77) 11*, 2, ACM, pp. 192 – 198, 1977

[11] Neumann L., Neumann A. and Szirmay-Kalos L., "Compact metallic reflectance models", *Computer Graphics Forum 18*, 13, 1999.

[12] Ashikhmin M. and Shirley P., "An Anisotropic Phong BRDF Model", *Journal of Graphics Tools 5*, 2, pp. 25 - 32, 2000.

[13] V. N. Vapnik, *Statistical Learning Theory*, New York: Wiley, 1998.

[14] S. Y. Yuen, Y. Y. Tsui, Y. W. Leung and M. M. Chen, "Fast marching method for shape from shading under perspective projection", *Proc. 2nd IASTED Int'l Conf. Visualization, Imaging and Image Processing*, pp. 584-588, 2002.
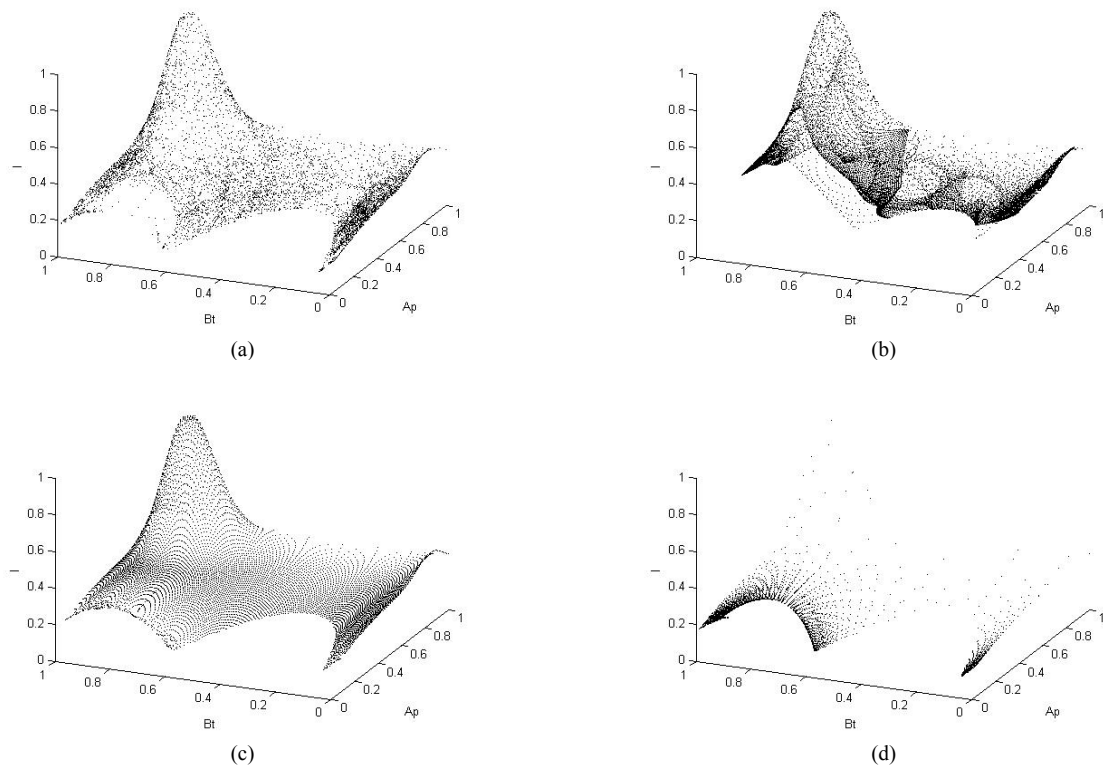
[15] http://www.3dcafe.com

Fig. 1. The empirical albedos of different depth models under the same shading equation. (a) Mozart, (b) 4 Mountains, (c) Sphere and (d) Vase
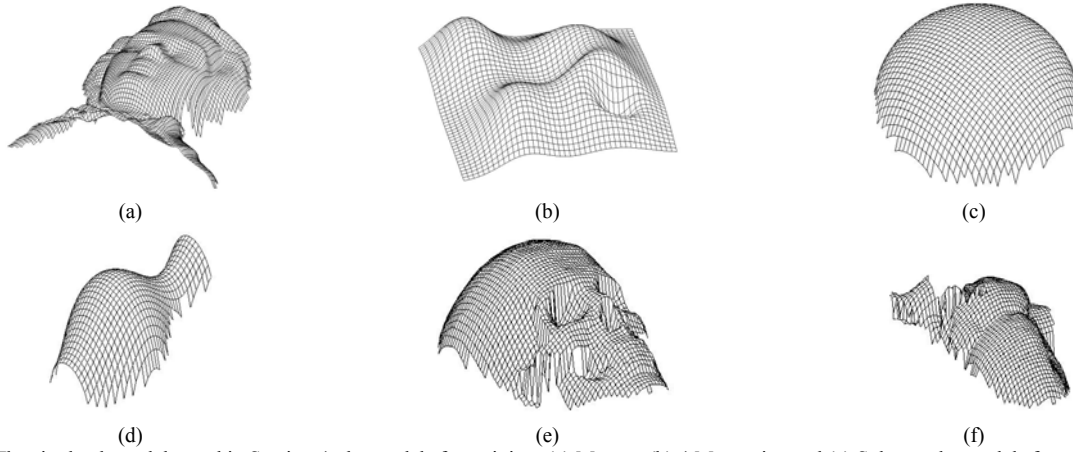
Fig. 2. The six depth models used in Section 4; the models for training: (a) Mozart, (b) 4 Mountains and (c) Sphere; the models for evaluation (d) Vase, (e) Skull and (f) Heart.
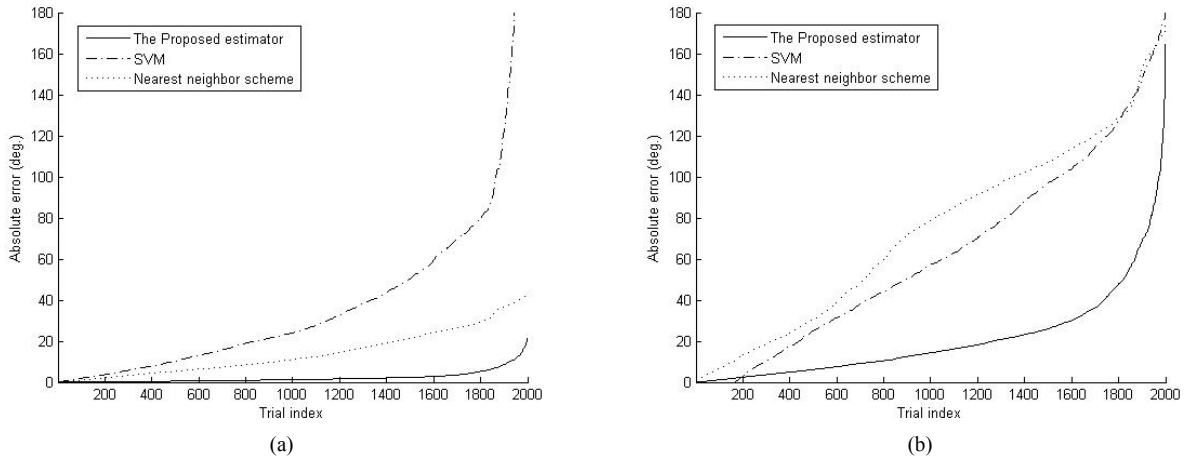

Fig. 3. The absolute angle errors: (a) the estimated θ_1 and (b) the estimated θ_2
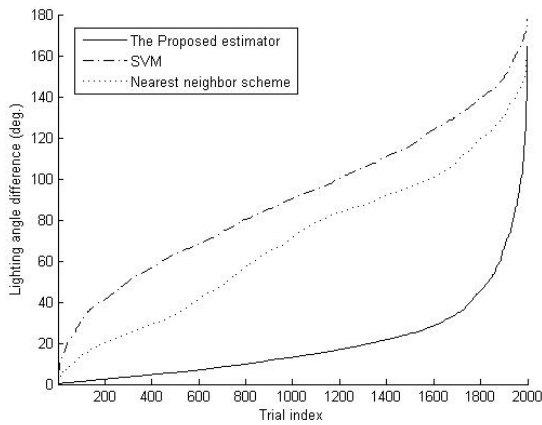

Fig. 4. The angle differences between $\mathbf{L_a}$ and $\mathbf{L_e}$