TR-IIS-95-001

Wavelet-Based Shape from Shading

Jun-Wei Hsieh, Hong-Yuan Mark Liao
Ming-Tat Ko, and Kuo-Chin Fan

Received January 12, 1995
Imprinted July 24, 1995

INSTITUTE OF INFORMATION SCIENCE
ACADEMIA SINICA

Wavelet–Based Shape from Shading

Jun–Wei Hsieh†, Hong–Yuan Mark Liao‡*, Ming–Tat Ko‡, and Kuo–Chin Fan†

† Institute of Computer Science and Electronic Engineering

National Central University, Chung–Li, Taiwan

‡Institute of Information Science, Academia Sinica, Taiwan

Keywords: Shape from shading, Wavelet transform, Constrained optimization.

** A preliminary version of this technical report was presented at the **IEEE International Conference on Image Processing**, Austin, Texas, Nov. 13–16, 1994.

* This work is supported by the NSC under Grants NSC 84–2213–E–001–008.

Abstract

This paper proposes a wavelet–based approach to solving the shape from shading (SFS) problem. The proposed method takes advantage of the nature of wavelet theory, which can be applied to efficiently and accurately represent "things," to develop a faster algorithm for reconstructing better surfaces. To derive the algorithm, the formulation of Horn [15], which combines several constraints into an objective function, is adopted. In order to improve the robustness of the algorithm, two new constraints are introduced into the objective function to strengthen the relation between an estimated surface and its counterpart in the original image. Thus , solving the SFS problem becomes a constrained optimization process. In the first stage of the process, the set of function variables to be solved is represented by a wavelet format. Due to this format, the set of differential operators of different orders which is involved in the whole process can be approximated with connection coefficients of Daubechies bases. In each iteration of the optimization process, an appropriate step size which will result in maximum decrease of the objective function is determined. After finding correct iterative schemes, the solution of the SFS problem will finally be decided. Compared with conventional algorithms, the proposed scheme is a great improvement in the accuracy as well as the convergence speed of the SFS problem. Experimental results, using both synthetic and real images, prove that the proposed method is indeed better than traditional methods.

List of Symbols

Symbol	Explanation
E	the observed image
R	the reflectance map function
Z	the height of the desired surface
$\vec{N}$	the normal of Z
$\hat{n}$	the unit normal of Z
$\vec{L}$	the light direction
σ_L	the slant angle of $\vec{L}$
τ_L	the tilt angle of $\vec{L}$
λ	the surface albedo
ϱ	the incident light flux
ξ	the bias brightness
p and q	the partial derivatives of Z with respect to image coordinates, i.e., $p = \frac{\partial Z}{\partial x}$ and $q = \frac{\partial Z}{\partial y}$
R_p and R_q	the partial derivatives of R with respect to p and q, i.e., $R_p = \frac{\partial R}{\partial p}$ and $R_q = \frac{\partial R}{\partial q}$
$\phi_{j,n}$	an orthonomal basis generated by a scaling function $\phi(x)$
$\psi_{j,n}$	an orthonomal basis generated by a wavelet function $\psi(x)$
$z_{i,j}$, $p_{i,j}$, and $q_{i,j}$	the weighting coefficients
W	an objective function
M^2	the size of an image
V_j	the function space spanned by $\{\phi_{j,n}(x)\}_{n\in Z}$
W_j	the function space spanned by $\{\psi_{j,n}(x)\}_{n\in Z}$
$S_j f(x)$	the projection of a continuous function, $f(x)$, on V_j
$W_j f(x)$	the projection of a continuous function, $f(x)$, on W_j
$D(n)$	the Delta function
N	the number of vanishing moments
$\Gamma_{n,0}$ and $\Gamma^a_{n,k}$	the connection coefficients
$\Gamma^{(1)}(k)$, $\Gamma^{(2)}(k)$, $\Gamma^{(3)}(k)$ and $\Gamma^{(4)}(k)$	the 1–D connection coefficients
$\Gamma^{(1)}_x(m,n)$, $\Gamma^{(1)}_y(m,n)$, $\Gamma^{(2)}_x(m,n)$, *etc.*	the 2–D connection coefficients

1. Introduction

The procedure for recovering three–dimensional surfaces of unknown objects is an important task in computer vision research. A robust procedure which can correctly reconstruct surfaces of an object is important in various applications such as visual inspection, autonomous land vehicle navigation, surveillance and robot control, etc. In the past decade, there have been extensive studies on this topic [1]–[35]; for example, shape from defocusing [2]–[3], shape from stereopsis [4], shape from motion [5]–[6], shape from texture [7]–[9] and shape from shading [10]–[35]. One of these methods, i.e., the shape from shading (**SFS**) technique, recovers the surface shape of an object from a 2–D image given that the imaging geometry is known in advance. In general, the SFS problem is formulated as a first–order partial differential equation (PDE). Since the formulated first–order PDE is nonlinear and ill–posed, it cannot be solved by a general method. There have been extensive researches carried out to solve the SFS problem; for example, Pong and Haralick [10], Ikeuchi and Horn [14], Brooks and Horn [15]–[16], Horn [17], Chellappa [11]–[13], Pentland [20]–[25], Lee and Rosenfeld [26], Lee [27], Oliensis [28]–[31], Lee [32]–[33], etc. The above mentioned methods can be classified into two categories. Among them, one group tried to find a closed–form solution while the other group considered the SFS problem as a constrained optimization problem. As to the first approach, Pentland [20]–[23] developed a technique which assumes that the surface is locally spherical at each point. The results obtained were based on second–order partial derivatives of image intensities which are sensitive to noise. Lee and Rosenfeld [26] later improved this technique and claimed that they only required the first–derivatives of image intensity. In [25], Pentland developed another technique to solve the problem. His basic idea was to introduce a linear approximation to the true reflectance function. This technique tried to derive an efficient closed–form solution for the surface shape. In fact, the above techniques which tried to find a closed–form solution are simple but unstable and constrained. Usually, a closed–form solution is not certain to be found in this kind of problem. Therefore, some researchers tried to find a more stable and efficient approach for the problem.

In contrast to the closed–form approach, another group of researchers formulated the SFS problem as a kind of Euler equation and used the constrained optimization approach to solve it. In order to make the ill–posed equation become well–posed, a regularization technique [14]–[17] was usually applied. By introducing the assumption that a surface should be smooth, Horn and his colleagues [14]–[17] first characterized the problem as an objective function of discrete nodal variables. This kind of formulation made it possible to solve the problem by using numerical techniques. However, the existence and uniqueness of a

smoothed surface is not guaranteed in their approach. Therefore, a new technique which dealt with the integrability problem was developed later (Brooks and Horn [15]; Frankot and Chellappa [11]). In fact, most SFS related solutions did not take into account the issue of integrability. Frankot and Chellappa [11] argued that if the integrability constraint is not enforced, the solution to the SFS problem still remains ambiguous. In their work, an orthogonal basis set of functions was used to describe surfaces. The surfaces were then expressed as a linear combination of orthogonal basis functions. By using a standard iterative and least–square scheme, the closest set of coefficients was found. In [10], Haralick and his colleagues proposed a new method to solve the SFS problem. They used a facet model as a basis set of functions to describe and recover surfaces from single or multiple images. The facet model assumes that the three–dimensional object surface is locally fit by a quadratic surface. Under the assumption that this quadratic object surface has Lambertian reflectance, an intensity image surface is then modeled. Having estimated the free parameters of the Lambertian intensity image, the object surface is then recovered. Inspired by the works of Haralick [10] *et al.,* and others [11], [13], [15], we propose a wavelet–based approach to solve the SFS problem in this paper.

Wavelet theory has proved to be a useful tool in various applications such as numerical analysis [62], pattern recognition [63], image interpolation [64]–[65], image compression [43], solutions to differential equations [50], etc. Basically, wavelet theory is a mathematical tool used for describing "things" more efficiently and precisely. Since a wavelet set has superior representation capability, in this paper we shall use this characteristic to efficiently and accurately solve the SFS problem. In order to obtain a good solution to this ill–posed problem, an objective function which should be minimized to yield an optimal solution is introduced. In this paper, the objective function defined by Horn [15], which combines a brightness constraint, an integrability constraint, and a smoothness constraint, is adopted. In order to improve the robustness of the algorithm, two new constraints are introduced into the objective function to strengthen the relation between an estimated surface and its counterpart in the original image. Thus, solving the SFS problem becomes a constrained optimization process. In the first stage of the process, the set of function variables to be solved is represented by a wavelet format. Due to this format, the set of differential operators of different orders which is involved in the whole process can be approximated with connection coefficients of a Daubechies basis. In each iteration of the optimization process, an appropriate step size which will result in a maximum decrease of the objective function is determined. After finding correct iterative schemes, the solution of the SFS problem will finally be decided. In comparison with conventional meth-

ods, there are several advantages when wavelet theory is applied for solving the SFS problem. First, since wavelets can be used to represent both the function variables and differential operators more accurately, the error term in the computation process will drop to a satisfactory level in a very short period. Second , since the representation is very accurate, the quality of the reconstructed surface is better than that from other approaches. The rest of the paper is organized as follows. In the next section, the basic formulation of the SFS problem and its relation to wavelets will be briefly addressed. Then, some preliminary issues including the basic concepts of the wavelet transform and its relation to differential operators will be introduced in Section 3. In Section 4, the proposed solution of the SFS problem will be clearly described. The experimental results will be reported in Section 5. Finally, a conclusion and discussion will be presented in Section 6.

2. Shape from Shading and Wavelets

In this section, we shall introduce the basic formulation of the SFS problem. Furthermore, the key concept of how to incorporate wavelet theory in the SFS problem will also be discussed. In general, in order to solve the SFS problem, there are two major concerns. First, a mathematical model which correctly describes the relations between a recovered surface and its corresponding region in the original image has to be found. Second, there must exist an efficient and robust method to determine the parameters of this model. In the SFS problem, an image is usually modeled as follows:

$$E = R(p, q),$$

where E is the observed image, R is the reflectance map function, $p = \frac{\partial Z}{\partial x}$, $q = \frac{\partial Z}{\partial y}$ and $Z(x,y)$ is the height of the desired surface. The reflectance map function, R, is a function of the angle between the surface normal, $\vec{N}$, and the light direction, $\vec{L}$. Thus, R at each pixel (x,y) can be expressed as follows [13], [15]:

$$R(p, q) = \lambda \varrho \vec{N} \cdot \vec{L} + \xi, \tag{1}$$

where λ corresponds to the surface albedo, ϱ the incident light flux and ξ the bias brightness. The normal of surface Z at pixel (x, y) can be written as $(-p, -q, 1)^t$. By normalization, the unit normal $\hat{n}$ of surface Z at pixel (x,y) can be written as follows:

$$\hat{n} = \frac{1}{\sqrt{1 + p^2 + q^2}}(-p, -q, 1)^t. \tag{2}$$

The illumination direction $\vec{L}$ is defined by $(\sin\sigma_L \cos\tau_L, \sin\sigma_L \sin\tau_L, \cos\sigma_L)$, where σ_L and τ_L are the slant and tilt angles, respectively. From Equations (1) and (2), R can be rewritten as follows:

$$R(p,q) = \lambda\varrho \frac{-\sin\sigma_L \cos\tau_L p - \sin\sigma_L \sin\tau_L q + \cos\sigma_L}{\sqrt{1+p^2+q^2}} + \xi. \tag{3}$$

In this model, λ, ϱ, ξ, τ_L and σ_L can all be estimated from the observed image *E(x,y)* [13], [22], [26]. In the SFS problem, the ultimate goal is to reconstruct the unknown surface *Z(x,y)*. However, there is no existent direct solution for *Z(x,y)*. Therefore, two aforementioned variables, p and q, were introduced to indirectly solve the problem. In Equation (3), p and q are a pair of unknowns to be solved. Basically, Equation (3) is a first–order PDE. Since there is only one equation while two variables need to be solved, in the present form the problem is, therefore, an ill–posed problem. It is well–known that a nonlinear and ill–posed equation has no unique solution. Hence, in order to derive the solution of the SFS problem, this problem is reformulated in the constrained optimization style. A typical cost function of the SFS problem is usually defined as follows [11]–[17]:

$$W = \int\int \Big\{ [E(x,y) - R(p,q)]^2 + \mu_1 \ (p_x^2 + p_y^2 + q_x^2 + q_y^2) + \mu_2 [(Z_x - p)^2 + (Z_y - q)^2] \Big\} \ dx \ dy. \tag{4}$$

The first term on the right hand side of Equation (4) denotes the "brightness error." The second term is a measure of "departure from smoothness." The third term enforces the integrability of the constrained optimization problem [11], [17]. μ_1 and μ_2 are the Lagrange multipliers. In Equation (4), the objective is to solve the set of unknowns p, q and Z subject to an optimization process which minimizes the cost function W. It is well–known that a constrained optimization problem needs an extraordinary amount of computation. Therefore, an efficient method which requires less computation is preferable. In this paper, we apply wavelet theory to propose an efficient method for solving the SFS problem. Since a wavelet set can efficiently and accurately represent things, we thus select a wavelet–based scheme to represent the set of function variables p, q, and Z. We expect from this good beginning that the solutions of the SFS problem can be more accurately solved in an efficient manner. By properly defining an M^2 admissible function space spanned by a set of finitely supported wavelet bases, the set of function variables p, q and Z which are required to be solved can be represented by the following wavelet format:

$$Z(x,y) = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} z_{i,j} \phi_{i,j}(x,y), \tag{5}$$

$$p(x,y) = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} p_{i,j} \phi_{i,j}(x,y),$$

$$\text{and } q(x,y) = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} q_{i,j}\phi_{i,j}(x,y),$$

where $\phi_{i,j}$ is the scaling function basis of the space V= **span** $\{\phi_{0,0}, \phi_{0,1}, \cdots, \phi_{0,M-1}, \phi_{1,0}, \cdots, \phi_{M-1,M-1}\}$ and $z_{i,j}$, $p_{i,j}$ and $q_{i,j}$ are the weighting coefficients. If the set of weighting coefficients $z_{i,j}$, $p_{i,j}$ and $q_{i,j}$ that minimizes the cost function W can be determined, then the SFS problem can be solved. Therefore, the SFS problem can be considered a process which requires minimizing an objective function W of $3M^2$ nodal variables $z_{i,j}$, $p_{i,j}$ and $q_{i,j}$. Basically, there exist several types of scaling basis functions which can be adopted to solve this problem. In this paper, we select the tensor product of the third–order Daubechies' scaling function [44]–[48] to span the solution space. The details about how to apply wavelet theory to the SFS problem will be further discussed in Section 4. In what follows, we shall address some preliminary issues as preparation for solving the SFS problem.

3. Relation between Wavelet Basis and Differential Operators

Since wavelet theory will be applied as the basis for deriving the solution of the SFS problem, in what follows we shall introduce some key concepts and properties of this mathematical tool. Then, due to the fact that the SFS problem normally requires the calculation of a number of differential equations, the relationship between wavelets and differential operators of different orders will also be discussed. Basically, wavelet theory is a noble mathematical tool which can be used to represent a "thing" more efficiently and precisely. The thing could be a signal, system, process or some physical phenomenon approximated by a set of "special elements." These special elements, which must be oscillatory and must quickly decay to zero, are called "wavelets." Each wavelet in the wavelet set comes from a single function, i.e., the so–called "mother wavelet." In general, a signal function $f(x)$ can be broken down into many little wavelets which are generated by a mother wavelet. This breaking process is called a wavelet transform. By putting these little wavelets back together, one can reconstruct the signal function $f(x)$. This process is recognized as the inverse wavelet transform.

Basically, the process of wavelet transform represents a continuous function, $f(x)$, with a limited number of successive approximations, each of which is basically a smoothed version of $f(x)$ [59], [65]. In this paper, we will employ the Daubechies scaling function [44]–[48] in wavelet theory to represent continuous functions. Denote the Daubechies scaling function by $\phi(x)$ and its dilation and translation functions

$2^{j/2}\phi(2^j x - n)$ by $\phi_{j,n}(x)$ for $j,\ n \in Z$. Let V_j be the function space spanned by $\{\phi_{j,n}(x)\}_{n\in Z}$. In fact, $\{\phi_{j,n}(x)\}_{n\in Z}$ is an orthonomal basis of V_j. The function spaces $V_j,\ j \in Z$ have the following properties:

1) $V_j \subseteq V_{j+1}$ for all $j \in Z$ and

2) $\bigcup_{j=-\infty}^{\infty} V_j = L^2(R)$.

Let $S_j f(x)$ be the projection of a continuous function, $f(x)$, on V_j, i.e.

$$S_j f = \sum_n c_{j,n}\phi_{j,n}(x), n \in Z,$$

where $c_{j,n} = \int f(x)\phi_{j,n}(x)dx$. By property (2), $\{S_j f(x)\}_{n\in Z}$ is an approximation scheme of $f(x)$ in which $S_{j+1} f(x)$ is a better approximation than $S_j f(x)$ for all $j \in Z$.

The difference between two successive approximations, $S_j f(x)$ and $S_{j+1} f(x)$, can be expanded by another set of orthonomal basis $\psi_{j,n}(x)$ which is generated by dilation and translation from another prototype function $\psi(x)$, called the Daubechies wavelet function. Let $W_j f(x) = S_{j+1} f(x) - S_j f(x)$. We have

$$W_j f(x) = \sum_n d_{j,n}\psi_{j,n}(x), n \in Z,$$

where $\psi_{j,n}(x) = 2^{j/2}\psi(2^j x - n)$ and $d_{j,n} = \int f(x)\psi_{j,n}(x)dx$. Let W_j denote the function space spanned by $\{\psi_{j,n}(x)\}_{n\in Z}$. We have

$$L^2(R) = (\bigoplus_{k=j}^{\infty} W_j) \oplus V_j \text{ for all } j \in Z.$$

In other words, any continuous function $f(x) \in L^2(R)$ can be approximated by

$$f(x) = S_j f(x) + \sum_{k=j}^{\infty}\sum_{n\in Z} d_{k,n}\psi_{k,n}(x). \tag{6}$$

In what follows, we shall try to determine the relation between the wavelet transform and differential operators. Basically, this relation is a key issue for efficiently and accurately solving the SFS problem. In order to obtain the solution of the SFS problem, a number of differential operators of different orders

which are involved in the optimization process have to be handled. In the SFS problem, the set of function variables to be solved is p, q, and Z. In order to represent these variables efficiently and accurately, in Section 2, we have proposed using linear combinations of wavelet bases to represent them. Here, we shall discuss how to represent differential operators of different orders by using linear combinations of wavelet bases so that the optimization process can be executed more efficiently and accurately. To start, we represent the original signal $f(x)$ as follows [54]:

$$f(x) = \sum_n c_n \phi_n(x), \tag{7}$$

where $c_n = \int f(x)\phi(x-n)dx$. Since $\int \phi(x-n)dx = 1$ and $\phi_n(x)$ is compactly supported, we can think of $\phi(x-n)$ as a function which is similar to a delta function. Therefore, we have

$$f(n) \approx \int f(x)\phi(x-n)dx. \tag{8}$$

Substituting Equation (8) into Equation (7) and differentiating it, we have

$$f'(x) = \sum_n f(n)\phi'_n(x). \tag{9}$$

If we expand $\phi'_n(x)$ based on the scaling function and wavelet function, then the following equation is obtained:

$$\phi_n'(x) = \sum_k \Gamma_{n,k}\phi_k(x) + \sum_{j\geq 0,k} \Gamma^j_{n,k}\psi_{j,k}(x), \tag{10}$$

where

$$\Gamma_{n,k} = \int \phi'(x-n)\phi(x-k)dx, \tag{11}$$

and

$$\Gamma^j_{n,k} = \int \phi'(x-n)\psi_j(x-k)dx. \tag{12}$$

The above formulation is the so–called wavelet–Galerkin method [54]. $\Gamma_{n,k}$ and $\Gamma^j_{n,k}$ are called connection coefficients [59]–[60]. These coefficients can be explicitly calculated for specific families of wavelets. For example, the Daubechies wavelet system [45] with N=2 vanishing moments can be calculated and expressed as follows:

$$\Gamma_{n,0} = (\frac{1}{12}, -\frac{8}{12}, 0, \frac{8}{12}, \frac{-1}{12}), \quad n = -2, -1, 0, 1, 2.$$

Other examples of connection coefficients can be found in [59]–[60]. If we substitute Equation (10) into Equation (9), we have

$$f'(x) = \sum_{n,k} f(n)\Gamma_{n,k}\phi_k(x) + \sum_{j\geq 0,n,k} f(n)\Gamma^j_{n,k}\psi_{j,k}(x). \tag{13}$$

By multiplying both sides of Equation (13) with $\phi_k(x)$ and integrating, we find that

$$f'(k) \approx \int f'(x)\phi(x-k)dx = \sum_n f(n)\Gamma_{n,k} = \sum_n \Gamma_{n,0} f(k+n).$$

If the scaling function $\phi(x)$ has N vanishing moments, the above equation can be further simplified as follows [54], [57]:

$$f'(k) \approx \sum_{n=-2N+2}^{2N-2} \Gamma_{n,0} f(k+n). \tag{14}$$

As to the second–order case, a similar derivation process can be applied accordingly. A function variable *f(k)* which is operated on by a second–order derivation can be represented as follows:

$$f''(k) \approx \sum_{n=-2N+2}^{2N-2} \Gamma^a_{n,0} f(k+n), \tag{15}$$

where

$$\Gamma^a_{n,k} = \int \phi''(x-n)\phi(x-k)dx.$$

Basically, the first–order and second–order differential operators in Equations (14) and (15) are calculated and absorbed in $\Gamma_{n,0}$ and $\Gamma^a_{n,k}$, respectively. From Equations (14) and (15), we can find that both differentiated function variables (f' and f'') can be approximated by linear combinations of the translations of their original function variable (f). It is obvious that the connection coefficients which absorb the calculation of differential operators play an important role in bridging the gap between wavelets and differential operators. In this paper, all connection coefficients are calculated based on a specific family of wavelets, i.e., the Daubechies wavelet system[45].

4. Wavelet–based Solution to the Shape from Shading Problem

In this paper, we propose a wavelet–based method, which is more robust, efficient, and accurate than traditional methods, to solve the SFS problem. In order to improve the robustness, we propose to modify the cost function. It is known that in Zheng and Chellappa's paper [13], they required that the gradients of the reconstructed intensity must equal that of the input image, i.e.,

$$R_x(p,q) = E_x(x,y) \quad \text{and} \quad R_y(p,q) = E_y(x,y).$$

However, if these nonlinear constraints are introduced into the cost function, the computation load will be significantly increased. In order to alleviate the above difficulties while still maintaining the relations among Z, p and q , we propose to introduce two new constraints as follows:

$$Z_{xx} = p_x \quad \text{and} \quad Z_{yy} = q_y.$$

Having the new constraints, the cost function can be redefined as follows:

$$W = \int\int \Big\{[E(x,y) - R(p,q)]^2 + (p_x^2 + p_y^2 + q_x^2 + q_y^2) + (Z_{xx} - p_x)^2 + (Z_{yy} - q_y)^2 + [(Z_x - p)^2 + (Z_y - q)^2]\Big\} dxdy. \tag{16}$$

It is noticeable that the new constraints not only enforce integrability but also introduce a smoothness constraint in an implicit manner. In what follows, we are ready to detail an efficient algorithm for solving the SFS problem based on wavelet theory.

First of all, we assume that the surface $Z(x,y)$ to be reconstructed is defined at the fine resolution 0, i.e., $Z(x,y) = S_0 Z(x,y)$. As was described in Section 2, $Z(x,y)$ can be represented by a wavelet format as follows:

$$Z(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{M-1} z_{m,n}\phi(x-m, y-n), \tag{17.a}$$

where $z_{m,n}$ are the weighting coefficients, and $\phi(x-m, y-n)$'s are the wavelet basis of a certain subspace at the fine resolution 0. The size of surface $Z(x,y)$ is $M \times M$. Similarly, $p(x,y)$ and $q(x,y)$ can be defined accordingly as follows:

$$p(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{M-1} p_{m,n}\phi(x-m, y-n), \tag{17.b}$$

$$\text{and } q(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{M-1} q_{m,n}\phi(x-m, y-n), \tag{17.c}$$

where $p_{m,n}$ and $q_{m,n}$ are the weighting coefficients. Plugging Z, p and q defined in Equation (17) into Equation (16), we have

$$W = \int\int \Big\{[E(x,y) - R(\sum_{m,n=0}^{M-1} p_{m,n}\phi_{m,n}(x,y), \sum_{m,n=0}^{M-1} q_{m,n}\phi_{m,n}(x,y))]^2 + [(\sum_{m,n=0}^{M-1} p_{m,n}\phi_{m,n}^{(x)}(x,y))^2 + (\sum_{m,n=0}^{M-1} p_{m,n}\phi_{m,n}^{(y)}(x,y))^2 + (\sum_{m,n=0}^{M-1} q_{m,n}\phi_{m,n}^{(x)}(x,y))^2 + (\sum_{m,n=0}^{M-1} q_{m,n}\phi_{m,n}^{(y)}(x,y))^2]$$

$$+ [(\sum_{m,n=0}^{M-1} z_{m,n}\phi_{m,n}^{(xx)}(x,y) - \sum_{m,n=0}^{M-1} p_{m,n}\phi_{m,n}^{(x)}(x,y))^2 + (\sum_{m,n=0}^{M-1} z_{m,n}\phi_{m,n}^{(yy)}(x,y) - \sum_{m,n=0}^{M-1} q_{m,n}\phi_{m,n}^{(y)}(x,y))^2]$$

$$+ [(\sum_{m,n=0}^{M-1} z_{m,n}\phi_{m,n}^{(x)}(x,y) - \sum_{m,n=0}^{M-1} p_{m,n}\phi_{m,n}(x,y))^2 + (\sum_{m,n=0}^{M-1} z_{m,n}\phi_{m,n}^{(x)}(x,y) - \sum_{m,n=0}^{M-1} q_{m,n}\phi_{m,n}(x,y))^2]\Big\} dxdy, \quad (18)$$

where $\phi_{m,n}(x,y) = \phi(x-m, y-n)$, $\phi_{m,n}^{(x)} = \frac{\partial}{\partial x}\phi_{m,n}(x,y)$ and $\phi_{m,n}^{(xx)} = \frac{\partial^2}{\partial x^2}\phi_{m,n}(x,y)$. Similarly, $\phi_{m,n}^{(y)} = \frac{\partial}{\partial y}\phi_{m,n}(x,y)$ and $\phi_{m,n}^{(yy)} = \frac{\partial^2}{\partial y^2}\phi_{m,n}(x,y)$. If the $3M^2$ nodal variables $p_{m,n}, q_{m,n}$ and $z_{m,n}$ that minimize the objective function W can be found, then the surface $Z(i,j)$ can be recovered. Before deriving the iterative schemes for p, q and Z, we have to define a set of connection coefficients [59]–[60] in advance. Basically, we can follow the similar derivation process described in Section 3 and define the following connection coefficients:

$$\Gamma_x^{(4)}(m,n) = \int\int \phi^{(xx)}(x,y)\phi_{m,n}^{(xx)}(x,y)dxdy,\ \Gamma_y^{(4)}(m,n) = \int\int \phi^{(yy)}(x,y)\phi_{m,n}^{(yy)}(x,y)dxdy,$$

$$\Gamma_{xy}^{(4)}(m,n) = \int\int \phi^{(xy)}(x,y)\phi_{m,n}^{(xy)}(x,y)dxdy,\ \Gamma_{yx}^{(4)}(m,n) = \int\int \phi^{(yx)}(x,y)\phi_{m,n}^{(yx)}(x,y)dxdy,$$

$$\Gamma_x^{(3)}(m,n) = \int\int \phi^{(xx)}(x,y)\phi_{m,n}^{(x)}(x,y)dxdy,\ \Gamma_y^{(3)}(m,n) = \int\int \phi^{(yy)}(x,y)\phi_{m,n}^{(y)}(x,y)dxdy,$$

$$\Gamma_x^{(2)}(m,n) = \int\int \phi^{(x)}(x,y)\phi_{m,n}^{(x)}(x,y)dxdy,\ \Gamma_y^{(2)}(m,n) = \int\int \phi^{(y)}\phi_{m,n}^{(y)}(x,y)dxdy,$$

$$\Gamma_x^{(1)}(m,n) = \int\int \phi^{(x)}(x,y)\phi_{m,n}(x,y)dxdy \text{ and } \Gamma_y^{(1)}(m,n) = \int\int \phi^{(y)}(x,y)\phi_{m,n}(x,y)dxdy.$$

If we decompose the cost function W as

$$W_1 = \int\int [E(x,y) - R(p(x,y), q(x,y))]^2 dxdy,$$

$$W_{21} = \int\int (\sum_{m,n=0}^{M-1} p_{m,n}\phi_{m,n}^{(x)}(x,y))^2 dxdy,$$

$$W_{22} = \int\int (\sum_{m,n=0}^{M-1} p_{m,n}\phi_{m,n}^{(y)}(x,y))^2 dxdy,$$

$$W_{23} = \int\int (\sum_{m,n=0}^{M-1} q_{m,n}\phi_{m,n}^{(x)}(x,y))^2 dxdy,$$

$$W_{24} = \int\int (\sum_{m,n=0}^{M-1} q_{m,n}\phi_{m,n}^{(y)}(x,y))^2 dxdy,$$

$$W_{31} = \int\int (\sum_{m,n=0}^{M-1} z_{m,n}\phi_{m,n}^{(xx)}(x,y) - \sum_{m,n=0}^{M-1} p_{m,n}\phi_{m,n}^{(x)}(x,y))^2 dxdy,$$

$$W_{32} = \int\int (\sum_{m,n=0}^{M-1} z_{m,n}\phi^{(yy)}_{m,n}(x,y) - \sum_{m,n=0}^{M-1} q_{m,n}\phi^{(y)}_{m,n}(x,y))^2 dxdy,$$

$$W_{41} = \int\int (\sum_{m,n=0}^{M-1} z_{m,n}\phi^{(x)}_{m,n}(x,y) - \sum_{m,n=0}^{M-1} p_{m,n}\phi_{m,n}(x,y))^2 dxdy,$$

$$\text{and } W_{42} = \int\int (\sum_{m,n=0}^{M-1} z_{m,n}\phi^{(y)}_{m,n}(x,y) - \sum_{m,n=0}^{M-1} q_{m,n}\phi_{m,n}(x,y))^2 dxdy,$$

then we have

$$W = W_1 + W_{21} + W_{22} + W_{23} + W_{24} + W_{31} + W_{32} + W_{41} + W_{42}. \tag{19}$$

Let $\delta p_{i,j}$, $\delta q_{i,j}$ and $\delta z_{i,j}$ represent the updating amounts of $p_{i,j}$, $q_{i,j}$, and $z_{i,j}$ in the iterative equations, respectively. If $p'_{i,j}$, $q'_{i,j}$ and $z'_{i,j}$ represent the values after update, then

$$p'_{i,j} = p_{i,j} + \delta p_{i,j}, \quad q'_{i,j} = q_{i,j} + \delta q_{i,j} \text{ and } z'_{i,j} = z_{i,j} + \delta z_{i,j}. \tag{20}$$

If we plug $p'_{i,j}$, $q'_{i,j}$ and $z'_{i,j}$ into W_1, W_1 will be updated by an amount δW_1. By Taylor's expansion, W_1 can be rewritten as follows:

$$W'_1 = W_1 + \delta W_1$$
$$= W_1 + 2(R-E)R_{p_{i,j}}\delta p_{i,j} + 2(R-E)R_{q_{i,j}}\delta q_{i,j} + 2R_{p_{i,j}}R_{q_{i,j}}\delta p_{i,j}\delta q_{i,j} + R^2_{p_{i,j}}\delta p^2_{i,j} + R^2_{q_{i,j}}\delta q^2_{i,j},$$

where the higher order derivatives R_{pp}, R_{qq}, and R_{pq} are all zero due to the linear approximation of R around (p,q). Similarly, if $p'_{i,j}$, $q'_{i,j}$ and $z'_{i,j}$ are plugged into W_{21}, the value of W_{21} will be changed by δW_{21}, that is:

$$W'_{21} = W_{21} + \delta W_{21} = \int\int (\sum_{m,n=0}^{M-1} p_{m,n}\phi^{(x)}(x-m,y-n) + \delta p_{i,j}\phi^{(x)}(x-i,y-j))^2 dxdy$$
$$= W_{21} + 2\delta p_{i,j}\sum_{m,n=0}^{M-1} p_{m,n}\int\int \phi^{(x)}_{m,n}(x,y)\phi^{(x)}_{i,j}(x,y)dxdy$$
$$+ \delta p^2_{i,j}\int\int \phi^{(x)}_{i,j}(x,y)\phi^{(x)}_{i,j}(x,y)dxdy$$
$$= W_{21} + 2\delta p_{i,j}\sum_{m,n=0}^{M-1} p_{m,n}\Gamma^{(2)}_x(i-m,j-n) + \delta p^2_{i,j}\Gamma^{(2)}_x(0,0).$$

The same derivation can be applied to W_{22}, W_{23} and W_{24}, where

$$W'_{22} = W_{22} + \delta W_{22} = W_{22} + 2\delta p_{i,j}\sum_{m,n=0}^{M-1} p_{m,n}\Gamma^{(2)}_y(i-m,j-n) + \delta p^2_{i,j}\Gamma^{(2)}_y(0,0),$$

$$W'_{23} = W_{23} + \delta W_{23} = W_{23} + 2\delta q_{i,j}\sum_{m,n=0}^{M-1} q_{m,n}\Gamma^{(2)}_x(i-m,j-n) + \delta q^2_{i,j}\Gamma^{(2)}_x(0,0),$$

and $W'_{24} = W_{24} + \delta W_{24} = W_{24} + 2\delta q_{i,j} \sum_{m,n=0}^{M-1} q_{m,n}\Gamma_y^{(2)}(i-m, j-n) + \delta q_{i,j}^2 \Gamma_y^{(2)}(0,0)$.

As to the case of W_{31} and W_{32}, we can derive accordingly and obtain the following:

$$
\begin{aligned}
W'_{31} &= W_{31} + \delta W_{31} \\
&= W_{31} - 2\delta p_{i,j}[\sum_{m,n=0}^{M-1} z_{m,n}\Gamma_x^{(3)}(i-m, j-n) - \sum_{m,n=0}^{M-1} p_{m,n}\Gamma_x^{(2)}(i-m, j-n)] + 2\delta p_{i,j}\delta z_{i,j}\Gamma_x^{(3)}(0,0) + \delta p_{i,j}^2\Gamma_x^{(2)}(0,0) \\
&\quad + \delta z_{i,j}^2\Gamma_x^{(4)}(0,0) + 2\delta z_{i,j}\sum_{m,n=0}^{M-1} z_{m,n}\Gamma_x^{(4)}(i-m, j-n) - 2\delta z_{i,j}\sum_{m,n=0}^{M-1} p_{m,n}\Gamma_x^{(3)}(m-i, n-j).
\end{aligned}
$$

$$
\begin{aligned}
W'_{32} &= W_{32} + \delta W_{32} \\
&= W_{32} - 2\delta q_{i,j}[\sum_{m,n=0}^{M-1} z_{m,n}\Gamma_y^{(3)}(i-m, j-n) - \sum_{m,n=0}^{M-1} q_{m,n}\Gamma_y^{(2)}(i-m, j-n)] + 2\delta q_{i,j}\delta z_{i,j}\Gamma_y^{(3)}(0,0) + \delta q_{i,j}^2\Gamma_y^{(2)}(0,0) \\
&\quad + \delta z_{i,j}^2\Gamma_y^{(4)}(0,0) + 2\delta z_{i,j}\sum_{m,n=0}^{M-1} z_{m,n}\Gamma_y^{(4)}(i-m, j-n) - 2\delta z_{i,j}\sum_{m,n=0}^{M-1} q_{m,n}\Gamma_y^{(3)}(m-i, n-j).
\end{aligned}
$$

For the case of W_{41} and W_{42}, we have

$$
\begin{aligned}
W'_{41} &= W_{41} + \delta W_{41} \\
&= W_{41} - 2\delta p_{i,j}[\sum_{m,n=0}^{M-1} z_{m,n}\Gamma_x^{(1)}(i-m, j-n) - p_{i,j}] + 2\delta p_{i,j}\delta z_{i,j}\Gamma_x^{(1)}(0,0) + \delta p_{i,j}^2 + \delta z_{i,j}^2\Gamma_x^{(2)}(0,0) \\
&\quad + 2\delta z_{i,j}\sum_{m,n=0}^{M-1} z_{m,n}\Gamma_x^{(2)}(i-m, j-n) - 2\delta z_{i,j}\sum_{m,n=0}^{M-1} p_{m,n}\Gamma_x^{(1)}(m-i, n-j).
\end{aligned}
$$

$$
\begin{aligned}
W'_{42} &= W_{42} - 2\delta q_{i,j}[\sum_{m,n=0}^{M-1} z_{m,n}\Gamma_y^{(1)}(i-m, j-n) - q_{i,j}] + 2\delta q_{i,j}\delta z_{i,j}\Gamma_y^{(1)}(0,0) + \delta q_{i,j}^2 + \delta z_{i,j}^2\Gamma_y^{(2)}(0,0) \\
&\quad + 2\delta z_{i,j}\sum_{m,n=0}^{M-1} z_{m,n}\Gamma_y^{(2)}(i-m, j-n) - 2\delta z_{i,j}\sum_{m,n} q_{m,n}\Gamma_y^{(1)}(m-i, n-j).
\end{aligned}
$$

Since $\Gamma_x^{(k)}(0,0) = 0$ and $\Gamma_y^{(k)}(0,0) = 0$ when k is an odd positive integer, we can thus reorganize the above terms and rewrite W' as follows:

$$
\begin{aligned}
W' &= W + \delta W = W + \delta W_1 + \delta W_{21} + \delta W_{22} + \delta W_{23} + \delta W_{24} + \delta W_{31} + \delta W_{32} + \delta W_{41} + \delta W_{42} \\
&= W + 2(R-E)R_p\delta p_{i,j} + 2(R-E)R_q\delta q_{i,j} + 2R_pR_q\delta p_{i,j}\delta q_{i,j} + R_p^2\delta p_{i,j}^2 + R_q^2\delta q_{i,j}^2 \\
&\quad + 2\delta p_{i,j}\sum_{m,n=0}^{M-1} p_{m,n}[2\Gamma_x^{(2)}(i-m, j-n) + \Gamma_y^{(2)}(i-m, j-n)] - 2\delta p_{i,j}\sum_{m,n=0}^{M-1} z_{m,n}\Gamma_x^{(3)}(i-m, j-n) \\
&\quad - 2\delta p_{i,j}\sum_{m,n=0}^{M-1} z_{m,n}\Gamma_x^{(1)}(i-m, j-n) + 2\delta p_{i,j}p_{i,j} + \delta p_{i,j}^2[2\Gamma_x^{(2)}(0,0) + \Gamma_y^{(2)}(0,0) + 1]
\end{aligned}
$$

$$+ 2\delta q_{i,j} \sum_{m,n=0}^{M-1} q_{m,n}[\Gamma_x^{(2)}(i-m,j-n) + 2\Gamma_y^{(2)}(i-m,j-n)] - 2\delta q_{i,j} \sum_{m,n=0}^{M-1} z_{m,n}\Gamma_y^{(3)}(i-m,j-n)$$

$$- 2\delta q_{i,j} \sum_{m,n=0}^{M-1} z_{m,n}\Gamma_y^{(1)}(i-m,j-n) + 2\delta q_{i,j}q_{i,j} + \delta q_{i,j}^2[\Gamma_x^{(2)}(0,0) + 2\Gamma_y^{(2)}(0,0) + 1]$$

$$+ 2\delta z_{i,j} \sum_{m,n=0}^{M-1} z_{m,n}[\Gamma_x^{(4)}(i-m,j-n) + \Gamma_y^{(4)}(i-m,j-n) + \Gamma_x^{(2)}(i-m,j-n) + \Gamma_y^{(2)}(i-m,j-n)]$$

$$- 2\delta z_{i,j} \sum_{m,n=0}^{M-1} p_{m,n}[\Gamma_x^{(3)}(m-i,n-j) + \Gamma_x^{(1)}(m-i,n-j)]$$

$$- 2\delta z_{i,j} \sum_{m,n=0}^{M-1} q_{m,n}[\Gamma_y^{(3)}(m-i,n-j) + \Gamma_y^{(1)}(m-i,n-j)]$$

$$+ \delta z_{i,j}^2[\Gamma_x^{(4)}(0,0) + \Gamma_y^{(4)}(0,0) + \Gamma_x^{(2)}(0,0) + \Gamma_y^{(2)}(0,0)]. \tag{21}$$

In order to insure that the terms in Equation (21) are in accordance with the format of the tensor product method [44]–[48], [61], we set $\phi(x,y) = \phi(x)\phi(y)$. In this form, the 2–D connection coefficients in Equation (21) can be converted into 1–D form. For example, the 2–D connection coefficient $\Gamma_x^{(4)}(m,n)$ can be converted into 1–D form by the following derivation:

$$\Gamma_x^{(4)}(m,n) = \int\int \phi^{(xx)}(x,y)\phi^{(xx)}(x-m,y-n)dxdy$$

$$= \int \phi^{(xx)}(x)\phi^{(xx)}(x-m)dx \int \phi(y)\phi(y-n)dy = \Gamma^{(4)}(m)D(n),$$

where $D(n) = \begin{cases} 1, & n = 0 \\ 0, & otherwise \end{cases}$ and $\Gamma^{(4)}(m) = \int \phi^{(xx)}(x)\phi^{(xx)}(x-m)dx$. Let $\Gamma^{(2)}(m) = \int \phi^{(x)}(x)\phi^{(x)}(x-m)dx$, $\Gamma^{(3)}(m) = \int \phi^{(xx)}(x)\phi^{(x)}(x-m)dx$ and $\Gamma^{(1)}(m) = \int \phi^{(x)}(x)\phi(x-m)dx$. The other connection coefficients can be derived accordingly as follows:

$$\Gamma_y^{(4)}(m,n) = D(m)\Gamma^{(4)}(n), \quad \Gamma_{xy}^{(4)}(m,n) = \Gamma^{(2)}(m)\Gamma^{(2)}(n), \quad \Gamma_{yx}^{(4)}(m,n) = \Gamma^{(2)}(m)\Gamma^{(2)}(n),$$

$$\Gamma_x^{(3)}(m,n) = D(n)\Gamma^{(3)}(m), \quad \Gamma_y^{(3)}(m,n) = D(m)\Gamma^{(3)}(n), \quad \Gamma_x^{(2)}(m,n) = D(n)\Gamma^{(2)}(m),$$

$$\Gamma_y^{(2)}(m,n) = D(m)\Gamma^{(2)}(n), \quad \Gamma_x^{(1)}(m,n) = D(n)\Gamma^{(1)}(m), \quad \text{and } \Gamma_y^{(1)}(m,n) = D(m)\Gamma^{(1)}(n).$$

Moreover, if the scaling function $\phi(x)$ has N vanishing moments [59][60], then for $m \notin [-2N+2, 2N-2]$, $\Gamma^{(4)}(m) = \Gamma^{(3)}(m) = \Gamma^{(2)}(m) = \Gamma^{(1)}(m) = 0$. Table 1 illustrates the 1–D connection coefficients for Daubechies's wavelet with N=3 vanishing moments. In order to make the optimization process efficient, δW has to be maximized. Therefore, from Equation (21), we can obtain the following equations:

$$\frac{\partial \delta W}{\partial \delta p_{i,j}} = 0, \quad \frac{\partial \delta W}{\partial \delta q_{i,j}} = 0 \quad \text{and} \quad \frac{\partial \delta W}{\partial \delta z_{i,j}} = 0.$$

From $\frac{\partial \delta W}{\partial \delta p_{i,j}} = 0$, we obtain

$$[R_p^2 + 3\Gamma^{(2)}(0) + 1]\delta p_{i,j} + R_p R_q \delta q_{i,j} = (E - R)R_p - p_{i,j} + \sum_{k=-2N+2}^{2N-2} z_{i-k,j}[\Gamma^{(3)}(k) + \Gamma^{(1)}(k)]$$
$$- \sum_{k=-2N+2}^{2N-2} [2p_{i-k,j} + p_{i,j-k}]\Gamma^{(2)}(k). \tag{22.a}$$

Similarly, from $\frac{\partial \delta W}{\partial \delta q_{i,j}} = 0$ and $\frac{\partial \delta W}{\partial \delta z_{i,j}} = 0$, we obtain

$$R_p R_q \delta p_{i,j} + [R_q^2 + 3\Gamma^{(2)}(0) + 1]\delta q_{i,j} = (E - R)R_q - q_{i,j} + \sum_{k=-2N+2}^{2N-2} z_{i,j-k}[\Gamma^{(3)}(k) + \Gamma^{(1)}(k)]$$
$$- \sum_{k=-2N+2}^{2N-2} [q_{i-k,j} + 2q_{i,j-k}]\Gamma^{(2)}(k), \tag{22.b}$$

and

$$2[\Gamma^{(4)}(0) + \Gamma^{(2)}(0)]\delta z_{i,j}$$
$$= - \sum_{k=-2N+2}^{2N-2} p_{i-k,j}[\Gamma^{(3)}(k) + \Gamma^{(1)}(k)] - \sum_{k=-2N+2}^{2N-2} q_{i,j-k}[\Gamma^{(3)}(k) + \Gamma^{(1)}(k)]$$
$$- \sum_{k=-2N+2}^{2N-2} z_{i-k,j}[\Gamma^{(4)}(k) + \Gamma^{(2)}(k)] - \sum_{k=-2N+2}^{2N-2} z_{i,j-k}[\Gamma^{(4)}(k) + \Gamma^{(2)}(k)]. \tag{22.c}$$

Equation (22) can be further simplified as follows:

$$D_{11}\delta p_{i,j} + R_{p_{i,j}} R_{q_{i,j}} \delta q_{i,j} = C_1, \tag{23.a}$$
$$R_{p_{i,j}} R_{q_{i,j}} \delta p_{i,j} + D_{22}\delta q_{i,j} = C_2 \tag{23.b}$$
$$\text{and } D_{33}\delta z_{i,j} = C_3, \tag{23.c}$$

where C_1 equals the right hand side of Equation (22.a), C_2 represents the right hand side of Equation (22.b), and C_3 is the right hand side of Equation (22.c). As to the values of D_{11}, D_{22} and D_{33}, we have $D_{11} = R_{p_{i,j}}^2 + 3\Gamma^{(2)}(0) + 1$, $D_{22} = R_{q_{i,j}}^2 + 3\Gamma^{(2)}(0) + 1$, and $D_{33} = 2\Gamma^{(2)}(0) + 2\Gamma^{(4)}(0)$. By solving Equation (23), we have

$$\delta p_{i,j} = (C_1 D_{22} - C_2 R_{p_{i,j}} R_{q_{i,j}})/D,$$
$$\delta q_{i,j} = (C_2 D_{11} - C_1 R_{p_{i,j}} R_{q_{i,j}})/D,$$
$$\text{and } \delta z_{i,j} = C_3/D_{33},$$

where $D = D_{11}D_{22} - R_{p_{i,j}}^2 R_{q_{i,j}}^2$. Summarizing the above results, the required iterative equations can be represented as follows:

$$p_{i,j}^{m+1} = p_{i,j}^{m} + \delta p_{i,j}, \quad q_{i,j}^{m+1} = q_{i,j}^{m} + \delta q_{i,j} \text{ and } z_{i,j}^{m+1} = z_{i,j}^{m} + \delta z_{i,j}. \tag{24}$$

Since the iterative equations shown in Equation (24) will be applied to all elements of an image, some boundary conditions have to be introduced to avoid those calculations that involve some pixels outside the legal image. Here, a pixel $z_{i,j}$ is classified as a boundary pixel if $i < 0$ or $j < 0$ or $i > M - 1$ or $j > M - 1$. Each boundary pixel should satisfy a set of boundary conditions as follows:

1. $z_{i,j} = z_{0,j}$ if $i < 0$ for each j.
2. $z_{i,j} = z_{i,0}$ if $j < 0$ for each i.
3. $z_{i,j} = z_{M-1,j}$ if $i > M - 1$ for each j.
4. $z_{i,j} = z_{i,M-1}$ if $j > M - 1$ for each i.

The set of boundary conditions also applies to $p_{i,j}$ and $q_{i,j}$. Based on Equation (24), we can iteratively solve the SFS problem. By using wavelet–based representations in the set of iterative equations, the desired set of solutions can converge in a more efficient way. The complexity of the proposed algorithm is $O(M^2)$. The fact that fewer iterations are required is the major advantage of the proposed method.

5. Experimental Results

In order to prove the efficiency and accuracy of our algorithm, we used two synthetic images and two real images as test images. Among them, two synthetic images were used to verify whether the proposed theory is accurate. On the other hand, two real images were used to examine how well this algorithm works. All test images were of size 256×256. In the experiments, two synthetic images were generated with the parameters $\lambda\varrho = 250$, $\sigma_L = \pi/4$, $\tau_L = \pi/4$ and $\xi = 0$. For each synthetic image, the surface heights around the border of each object were assumed to be zero. In addition, the estimated parameters $\lambda, \varrho, \sigma_L, \tau_L$ and ξ for two real images were derived by applying Zheng and Chellappa's method[13]. At each iteration of the dynamic equations, the Gauss–Seidel method was performed to update variables.

Figure 1 shows the experimental results of a synthetic portrait image. Figure 1(a) is a true range surface used to generate a 2–D synthetic image. Figure 1(b) shows the 3–D plot of (a). Figure 1(c) shows the test intensity image generated from Figure 1(a). The surface recovered by our algorithm is shown in Figure 1(d). The 3–D plot of the reconstructed surface is shown in Figure 1(e). From the above results, we can find that the reconstructed surface is close to the original surface. Since there are no sufficient boundary conditions available, the errors between the reconstructed surface and the original surface are indispensable. Figure 1(f) shows the image generated by projecting the reconstructed surface onto a 2–D surface. The set of parameters $\lambda, \varrho, \sigma_L, \tau_L$ and ξ adopted is the same as that used to generate Figure 1(c).

Figure 2 illustrates another example of a synthetic Mozart image. Figure 2(a) shows a true range surface which was used to generate a 2–D synthetic image. Figure 2(b) is the 3–D plot of 3(a). Figure 2(c) shows the test intensity image generated from (a) with parameters $\sigma_L = \pi/4$, $\tau_L = \pi/4$, $\varepsilon = 0$ and $\lambda\varrho = 250$. Notice that since the original depth surface has holes inside the object and discontinuities along the object boundaries, the 2–D synthetic image has noise at the places where those holes are located. The surface recovered under these circumstances is shown in Figure 2(d). The 3–D plot of the reconstructed surface is shown in Figure 2(e). Due to the discontinuities and holes, there exist indispensable errors between the reconstructed surface and the original surface. However, the deviation is tolerable and, therefore, the result is considered satisfactory. Figure 2(f) shows the image generated from the recovered surface by using the same set of parameters $\lambda, \varrho, \sigma_L, \tau_L$ and ξ.

In order to verify the performance of the proposed SFS algorithm, it was necessary to apply our algorithm to natural images. The next example shows the results obtained by applying our approach to the Lena image. Figure 3(a) shows the original Lena image. The reflectance map parameters estimated by Zheng and Chellppa's method[13] are $\lambda\varrho$=192.01, σ_L=59.92°, τ_L=7.74° and ξ=3.0. The depth map recovered by our algorithm is shown in Figure 3(b). It is difficult to evaluate the performance of our algorithm when it is applied to a real image. It can be seen from Figure 3(b) that the shape of the hat, cylinder and shoulder are recovered correctly. Other features such as nose, eyes, lips, etc. are also correctly reconstructed. Figure 3(c) is the 3–D plot of (b). Figure 3(d) shows the image synthesized from the reconstructed *(p,q)* maps with the same reflectance map parameters. Figure 3(e) shows another image synthesized from the reconstructed surface using the same set of reflectance map parameters except $\tau_L = 97.74°$.

Figure 4 is the case of another real image. Figure 4(a) shows the original multi–pepper image. The reflectance map parameters estimated are $\lambda\varrho$=255.96, σ_L=58.20°, τ_L=15.92° and ξ=0.0. The depth map recovered by our algorithm is shown in Figure 4(b). It is noticeable that this image is more complicated because it contains many small peppers, albedo variations, and shadows. For the above reason, some peppers which were originally inside the image could not be reconstructed correctly. However, most of the objects which were originally contained in the image were correctly recovered by our algorithm. Figure 4(c) shows the 3–D plot of (b). Figure 4(d) shows the image synthesized from the reconstructed *(p,q)* maps using the same set of reflectance map parameters. Figure 4(e) shows another image synthesized from the same set of reflectance map parameters except τ_L=105.92°.

In order to compare our algorithm with other methods, two brilliant works in the literature, which were respectively proposed by Horn [17] and Zheng–Chellppa [13], are selected. Here, we made some modifications to Horn's method so that it could converge even faster. In the comparison, we used the Lena image and the multi–pepper image, respectively. For each image, the average magnitudes of the "brightness error" and "integrability constraint" at each pixel were chosen as the indicators to compare their performances. Figure 5 shows the comparison results when the three methods were applied on the Lena image. Figure 5(a) shows how the average magnitude of the brightness error decreases over time for the three different methods. Figure 5(b) shows the results obtained by using the average magnitude of integrability constraint as an error measure. In this experiment, the results indicate that our method performs better than the other two methods. Figure 6 shows other comparison results when the three methods are applied to the multi–pepper image. Figures 6(a) and (b) are the results obtained by using the average brightness error at each pixel as a comparison measure. Since the average "brightness error" magnitude of (b), obtained by our method, is too small, it is therefore represented by a separate figure with different scale ($\times 10^{-5}$). In Figure 6(b), different N represent different compactly supported bases. The results, which show the average magnitude of integrability constraint of the three methods, are shown in Figure 6(c). From the results shown in Figures 5 and 6, it is obvious that the wavelet–based SFS algorithm is indeed an efficient , robust, and accurate way to solve the SFS problem.

6. Conclusions

In this paper, we have applied wavelet theory to solve the shape from shading problem for 3–D surface reconstruction. To derive this algorithm, we have adopted the formulation of Horn [15]–[17], which combines several constraints into an objective function, which is then minimized. In order to strengthen the relations among Z, p and q, two new constraints have been introduced into the objective function. The process of solving the SFS problem was then converted into a constrained optimization problem. The proposed method uses a wavelet basis to approximate a curved surface so that a new technique which can reconstruct a 3–D surface better and make the process converge faster can be developed. To verify the performance of the proposed method, two synthetic images and two two real images were adopted as test data. Experimental results proved that our method works better than do traditional methods, both in convergence speed and accuracy.

7. References

[1] R. M. Bolle and B. C. Vemuri, "On three dimensional surface reconstruction methods," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–13, no. 1, pp. 1–13, Jan. 1991.

[2] A. P. Pentland, "A new sense for depth of field," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–9, no. 4, pp. 523–531, July 1987.

[3] T. Hwang, J. J. Clark and A. L. Yuille, "A depth recovery algorithm using defocus information," *IEEE Conf. Computer Vision an Pattern Recognition*, pp. 476–481, 1989.

[4] G. Medioni and R. Nevatia, "Segment–based stereo matching," *Comput. Vision Graphics Image Processing*, vol. 31, pp. 2–18, July 1985.

[5] A. M. Waxman and K. Gurumoorthy, "Image flow theory: a framework for 3–D inference from time–varying imagery," in *Advances in Machine Vision,* ed. by C. Brown, Springer–Verlag, pp. 165–224, 1988.

[6] J. Weng, N. Ahuja and T. S. Huang, "Optimal motion and structure estimation," *IEEE Conf. Computer Vision an Pattern Recognition*, pp. 144–152, 1989.

[7] A. P. Witkin, "Recovering surface shape and orientation from texture," *Artificial Intell.,* vol. 17, pp. 17–47, 1981.

[8] J. R. Kender, "Shape from Texture," *Proc. 6th Inter. J. Conf. Artificial Intelligence*, Tokyo, pp. 562–570, 1979.

[9] Y. Choe and R. L. Kashyap, "3–D shape from a shading and textural surface image," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–13, no. 9, pp. 907–919, Sept. 1991.

[10] T. C. Pong, R. M. Haralick and L. G. Shapiro, "Shape from shading using the facet model," *Pattern Recognition*, vol. 22, no. 6, pp. 683–695, 1989.

[11] R. T. Frankot and R. Chellappa, "A method for enforcing integrability in shape from shading algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–10, no. 4, pp. 439–451, July 1988.

[12] T. Simchony and R. Chellappa, "Direct analytical methods for solving Poisson equations in computer vision problems," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–12, no. 5, pp. 435–446, May 1990.

[13] Zheng and R. Chellappa, "Estimation of illuminant direction, albedo, and shape from shading," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–13, no. 7, pp. 680–702, July 1991.

[14] K. Ikeuchi and B. K. P. Horn, "Numerical shape form shading and occluding boundaries," *Artificial Intell.*, vol. 17, pp. 141–184, August 1981.

[15] B. K. P. Horn and M. J. Brooks (Eds.), Shape from shading, Cambridge, MA; MIT Press, 1989.

[16] B. K. P. Horn and M. J. Brooks, "The variational approach to shape from shading," *Comput. Vision Graphics Image Processing*, vol. 33, pp. 174–188, 1986.

[17] B. K. P. Horn, "Height and gradient from shading," *Intern. J. Comput. Vision*, pp. 37–75, 1990.

[18] F. P. Ferrie and M.D. Levine, "Where and why local shading analysis works," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–11, no. 2, pp. 198–206, Feb. 1989.

[19] S. Peleg and G. Ron, "Nonlinear multiresolution: a shape–from–shading example," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–12, pp. 1206–1210, Dec. 1990.

[20] A. Pentland, "Local shading analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–6, pp. 170–178, Mar. 1984.

[21] A. Pentland and P. Kube, "On the imaging of fractal surfaces," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–10, pp. 704–707, Sept. 1988.

[22] A. Pentland, "Finding the illumination direction," *J. Opt. Soc. Amer.*, vol. 72, pp. 448–455, Apr. 1982.

[23] A. Pentland, "Fractal–based description of natural scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–6, pp. 661–674, Nov. 1984.

[24] A. Pentland, "Shading into texture," *Artificial Intell.,* vol. 29, pp. 147–170, 1986.

[25] A. Pentland, "Linear shape from shading," *Intern. J. Comput. Vision*, pp. 153–162, 1990.

[26] C. H. Lee and A. Rosenfeld, "Improved methods of estimating shape from shading using the light source coordinate system," *Artificial Intell.,* vol. 26, pp. 125–143, 1985.

[27] D. Lee, "A provably convergent algorithm for shape from shading," *DARPA Image Understanding Workshop,* Miami Beach, Florida, December, 1985.

[28] J. Oliensis, "Shape from shading as a partially well–constrained problem," *Comput. Vision Graphics Image Processing: Image Understanding*, vol. 54, No. 2, pp. 163–183, Sept. 1991.

[29] J. Oliensis, "Uniqueness in shape from shading," *Intern. J. Comput. Vision*, pp. 75–104, 1991.

[30] P. Dupuia and J. Oliensis, "Direct method for reconstructing shape from shading," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Champaign, Illinois, pp. 453–458, 1992.

[31] J. Oliensis and P. Dupuia, "A global algorithm for shape from shading," *The Fourth Intern. Conference on Computer Vision*, Berlin, Germany, pp. 692–701, May 1993.

[32] M. K. Lee and C. C. J. Kuo, "Shape from shading with a linear triangular element surface model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–15, no. 8, pp. 813–822, 1993.

[33] M. K. Lee and C. C. J. Kuo, "Shape from shading with perspective projection," *Comput. Vision Graphics Image Processing: Image Understanding*, vol. 59, No. 2, pp. 202–212, March. 1994.

[34] E. Rouy, A. Tourin, "A viscosity solutions approach to shape–from–shading," *SIAM J. Numer. Anal.* , vol. 29, No. 3, pp. 867–884, June 1992.

[35] F. P. Ferrie and J. Lagrade, "Curvature consistency improves local shading analysis," *Comput. Vision Graphics Image Processing: Image Understanding*, vol. 55, No. 1, pp. 95–105, Jan. 1992.

[36] J. W. Hsieh, H. Y. Liao, and K. C. Fan, "A new approach for edge detection using wavelet transforms," *Proc. Asian Conf. on Computer Vision*, Osaka, Japan, Nov. 23–25, pp. 520–525, 1993.

[37] J. W. Hsieh, H. Y. Liao, K. C. Fan and M. T. Ko, "A new wavelet based approach for edge detection," submitted to *IEEE Trans. on Image Processing*.

[38] S. Mallat and S. Zhong, "Characterization of signals from multi–scale edges," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–14, no. 7, pp. 710–732, July 1992.

[39] S. Mallat, "Multi–frequency channel decompositions of images and wavelet models," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP–37, no. 12, pp. 2091–2110, Dec. 1989.

[40] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–11, no. 12, pp. 674–693, July, 1989.

[41] S. Mallat, "Multiresolution approximation and wavelet orthonormal bases of $L(R^2)$," *Trans. Amer. Math. Soc.*, vol. 3–15, pp. 69–87, Sept. 1989.

[42] T. Chang and C. C. Jay Kuo, "Texture analysis and classification with tree–structured wavelet transform," *IEEE Trans. Image Processing.*, vol. 2, no. 4, pp. 429–441, October 1993.

[43] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image code using wavelet transform," *IEEE Trans. Image Processing.*, vol. 1, no. 2, pp. 205–220, April 1992.

[44] I. Daubechies, "The wavelet transform, time–frequency localization and signal analysis," *IEEE Trans. Information Theory.*, vol. 36, no. 5, pp. 961–1005, Sept. 1990.

[45] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. Pure Appl. Math.* vol. 41, pp. 909–996, 1988.

[46] I. Daubechies, Wavelets and applications, CBMS–Lecture Notes, to appear.

[47] I. Daubechies, *Ten Lectures on Wavelets*, CBMS–NSFReg. Conf. Ser. Appl. Math. 61, Philadelphia. Soc. Ind. Appl. Math.

[48] I. Daubechies, A. Grossman, and Y. Meyer, "Painless nonorthogonal expansions," *J. Math. Phys.*, vol. 27, pp. 1271–1283, 1986.

[49] A. Grossman, J. Morlet and T. Paul, "Transforms associated to square integrable group representations I: general results," *J. Math. Phys.*, vol. 26, pp. 2473–2479, 1985.

[50] V. Perrier, "Towards a method for solving partial differential equations using wavelet bases," *Wavelets: time–frequency methods and phase space* ed. by J. M. Combes, A. Grossman and Ph. Tchamithian, Sprinber–Verlag, pp. 269–283, 1989

[51] J.–C. Risset, "The computer, music, and sound models," *Wavelets: time–frequency methods and phase space* ed. by J. M. Combes, A. Grossman and Ph. Tchamithian, Sprinber–Verlag, pp. 102–123, 1989

[52] J. S. Lienard and C. d' Alsssandro, "Wavelets and Granular analysis of speech," *Wavelets: time–frequency methods and phase space* ed. by J. M. Combes, A. Grossman and Ph. Tchamithian, Sprinber–Verlag, pp. 158–163, 1989

[53] J. S. Lee, Y. N. Sun, C. H. Chen and C. T. Tsai, "Wavelet based corner detection," *Pattern Recognition*, vol. 26, no. 6, pp. 853–865, 1993.

[54] R. O. Wells and X. Zhou, "Wavelet interpolation and approximate solutions of elliptic partial differential equations," Technical Report, Rice University, 1993, Computational mathematics Laboratory.

[55] W. Hackbusch, "The frequency decomposition multi–grid method, part I: Application to anisotropic equations," *Numer. Math.*, vol. 56, pp. 229–245, 1989.

[56] A. Rieder, R. O. Wells and X. Zhou, "A wavelet approach to robust multilevel solvers for anisotropic elliptic problems," Technical Report, Rice University, 1993, Computational mathematics Laboratory.

[57] A. Rieder and X. Zhou, "On the robustness of the damped V–cycle of the wavelet frequency decomposition multi–grid method," Technical Report, Rice University, 1993, Computational mathematics Laboratory.

[58] D. Terzopoulos, "The computation of visible–surface representations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–10, no. 4, pp. 417–438, July 1988.

[59] G. Beylkin, "On the representation of operators in bases of compactly supported wavelets," *SIAM J. Numerical Analysis*, vol. 29, pp. 1716–1740, Dec. 1992.

[60] A. Latto, H. L. Resnikoff, and E. Tenenbaum, "The evaluation of connect coefficients of compactly supported wavelets," In *Proceedings of the USA–French Workshop on wavelet and Turbulence.* Princeton Univerity, 1991.

[61] Y. Meyer, Ondelettes. Vol I. hermann, Paris. 1990.

[62] Y. Meyer, "Wavelets and applications," in *Proceeding of the International Congress of Mathematics*, Kyoto, 1990

[63] Y. Sheng, D. Roberge, Harold Szu, and T. Lu, "Optical wavelet matched filters for shift–invariant pattern recognition," *Optics Letters*, vol. 18, no. 4, pp. 299–301, Feb. 1993.

[64] A. P. Pentland, "Interpolation using wavelet transform," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–16, no. 4, pp. 410–414, April 1994.

[65] M. H. Yoau and W. T. Chang, "Fast surface interpolation using multi–resolution wavelet transform," to be published in *IEEE Trans. Pattern Anal. Machine Intell.*

[66] D. M. Young, *Iterative solution of large linear systems.* Academic Press, New York, 1971.

Caption of Table

Table 1: The connection coefficients for Daubechies's basis with three vanishing moments.

Captions of Figures

Figure 1: The SFS algorithm applied to a portrait image. (a) The true range surface used to generate a synthetic image. (b) The 3–D plot of (a). (c) The input intensity image generated from (a) with parameters $\sigma_L = \pi/4$, $\tau_L = \pi/4$, $\xi = 0$ and $\lambda\varrho = 250$. (d) The reconstructed surface. (e) The 3–D plot of (d). (f) The output intensity image generated from (d) using the same set of parameters as in (c).

Figure 2: The SFS algorithm applied to the Mozart image. (a) The true range surface used to generate a synthetic image. (b) The 3–D plot of (a). (c) The input intensity image generated from (a) with parameters $\sigma_L = \pi/4$, $\tau_L = \pi/4$, $\xi = 0$ and $\lambda\varrho = 250$. (d) The reconstructed surface. (e) The 3–D plot of (d). (f) The output intensity image generated from (d) using the same set of parameters as in (c).

Figure 3: The SFS algorithm applied to the Lena image. (a) The input image. Its estimated reflectance map parameters are $\sigma_L = 59.52^\circ$, $\tau_L = 7.74^\circ$, $\lambda\varrho = 192.01$ and $\xi = 3.0$. (b) The reconstructed surface. (c) The 3–D plot of (b). (d) The image generated from the reconstructed (p, q) maps with the same reflectance map parameters. (e) The image synthesized from the reconstructed (p, q) using the same set of reflectance map parameters except $\tau_L = 97.74^\circ$.

Figure 4: The SFS algorithm applied to a multi–pepper image. (a) The input image. Its estimated reflectance map parameters are $\sigma_L = 58.20^\circ$, $\tau_L = 15.92^\circ$, $\lambda\varrho = 255.96$ and $\xi = 0.0$ (b) The reconstructed surface. (c) The 3–D plot of (b). (d) The image generated from the reconstructed (p, q) maps with the same reflectance map parameters. (e) The image synthesized from the reconstructed (p, q) using the same set of reflectance map parameters except $\tau_L = 105.92^\circ$.

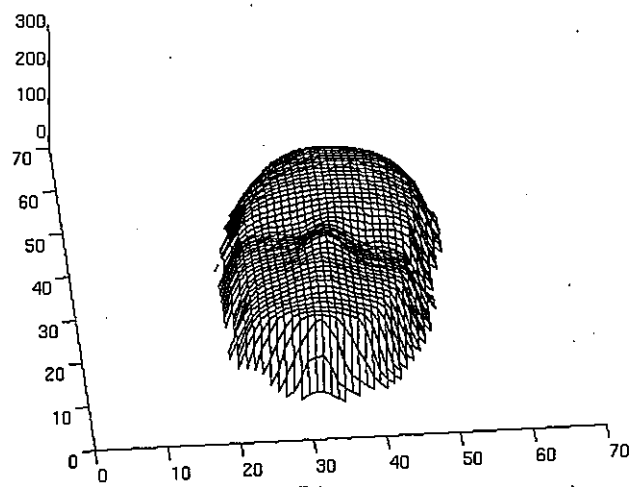
Figure 5: The comparisons of our SFS algorithm with two other algorithms based on the Lena image. (a) The results of adopting the average magnitude of the "brightness error" at each pixel as an error measure. (b) The results of adopting the average "integrability constraint" magnitude as a comparison measure.

Figure 6: The comparisons of our SFS algorithm with two other algorithms based on the multi–pepper image. (a) The average magnitudes of the "brightness error" generated by the other two algorithms. (b) The average "brightness error" of our algorithm. The scale of the vertical axis has been scaled down to 10^{-5}. Two curves in (b) are results obtained by different compactly supported basis (N=3, 4). (c) The results of adopting the average "integrability constraint" magnitude as a comparison measure.

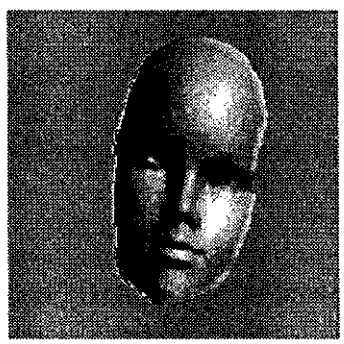
$\Gamma^{(1)}[k]$	$\Gamma^{(2)}[k]$	$\Gamma^{(3)}[k]$	$\Gamma^{(4)}[k]$
$\Gamma^{(1)}[-4] = 0.00034246575342$	$\Gamma^{(2)}[-4] = -0.00535714285714$	$\Gamma^{(3)}[-4] = -0.0075$	$\Gamma^{(4)}[-4] = -0.05625$
$\Gamma^{(1)}[-3] = 0.01461187214612$	$\Gamma^{(2)}[-3] = -0.11428571428572$	$\Gamma^{(3)}[-3] = -0.0800$	$\Gamma^{(4)}[-3] = -0.30000$
$\Gamma^{(1)}[-2] = -0.14520547945206$	$\Gamma^{(2)}[-2] = 0.87619047619047$	$\Gamma^{(3)}[-2] = 0.8950$	$\Gamma^{(4)}[-2] = 1.92500$
$\Gamma^{(1)}[-1] = 0.74520547945206$	$\Gamma^{(2)}[-1] = -3.39047619047619$	$\Gamma^{(3)}[-1] = -1.5200$	$\Gamma^{(4)}[-1] = -4.10000$
$\Gamma^{(1)}[0] = 0.0$	$\Gamma^{(2)}[0] = 5.26785714285715$	$\Gamma^{(3)}[0] = 0.0000$	$\Gamma^{(4)}[0] = 5.06250$
$\Gamma^{(1)}[1] = -0.74520547945206$	$\Gamma^{(2)}[1] = -3.39047619047619$	$\Gamma^{(3)}[1] = 1.5200$	$\Gamma^{(4)}[1] = -4.10000$
$\Gamma^{(1)}[2] = 0.14520547945206$	$\Gamma^{(2)}[2] = 0.87619047619047$	$\Gamma^{(3)}[2] = -0.8950$	$\Gamma^{(4)}[2] = 1.92500$
$\Gamma^{(1)}[3] = -0.01461187214612$	$\Gamma^{(2)}[3] = -0.11428571428572$	$\Gamma^{(3)}[3] = 0.0800$	$\Gamma^{(4)}[3] = -0.30000$
$\Gamma^{(1)}[4] = -0.00034246575342$	$\Gamma^{(2)}[4] = -0.00535714285714$	$\Gamma^{(3)}[4] = 0.0075$	$\Gamma^{(4)}[4] = -0.05625$

Table 1. The connection coefficients for Daubechies's basis with three vanishing moments.

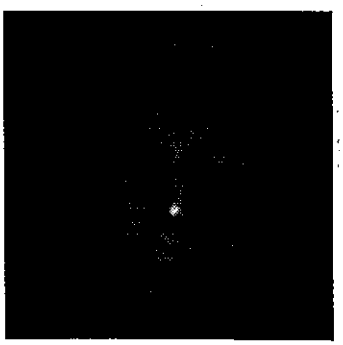
(a)



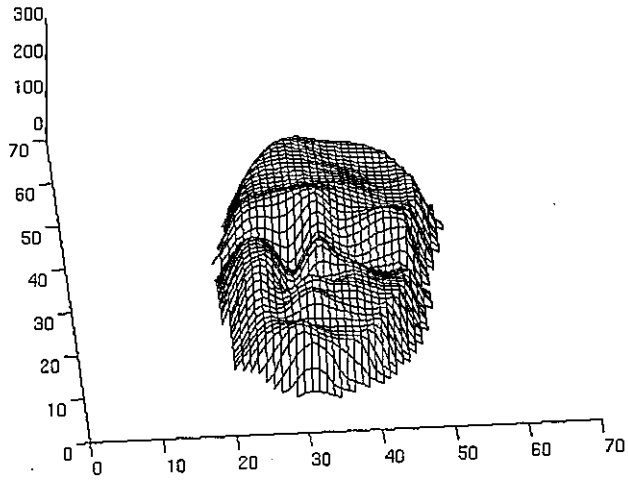
(b)



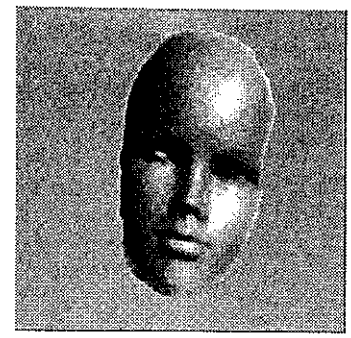
(c)



(d)

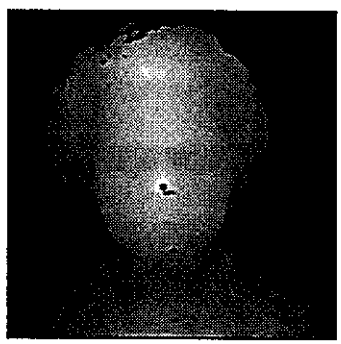


(e)

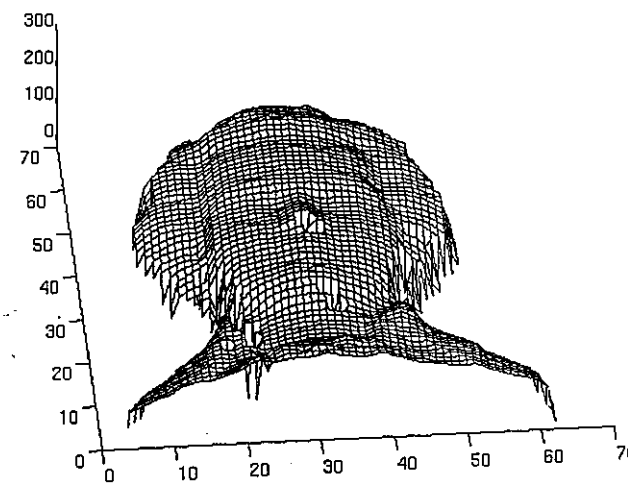


(f)

Figure 1. The SFS algorithm applied to a portrait image. (a) The true range surface used to generate a synthetic image. (b) The 3–D plot of (a). (c) The input intensity image generated from (a) with parameters $\sigma_L = \pi/4$, $\tau_L = \pi/4$, $\xi = 0$ and $\lambda\varrho = 250$. (d) The reconstructed surface. (e) The 3–D plot of (d). (f) The output intensity image generated from (d) using the same set of parameters as in (c).



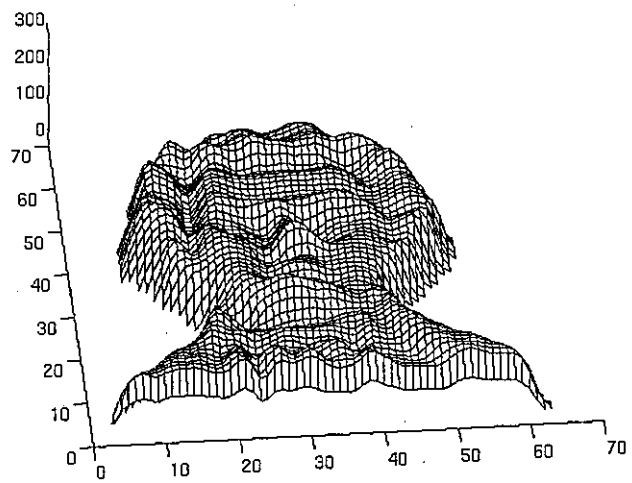
(a)



(b)

(c)

(d)



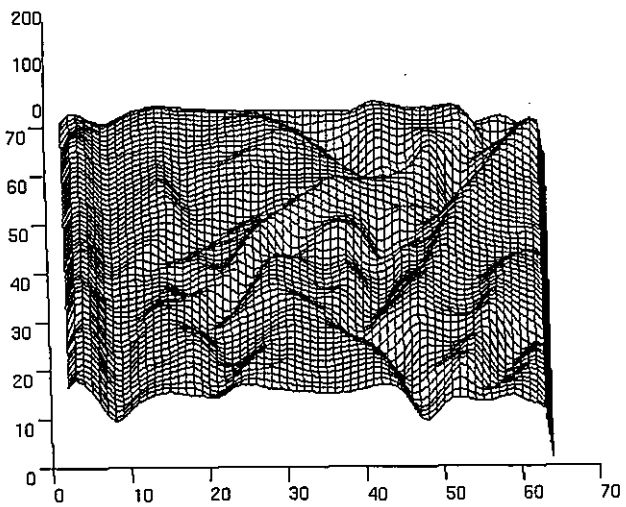
(e)

(f)

Figure 2. The SFS algorithm applied to the Mozart image. (a) The true range surface used to generate a synthetic image. (b) The 3–D plot of (a). (c) The input intensity image generated from (a) with parameters $\sigma_L = \pi/4$, $\tau_L = \pi/4$, $\xi = 0$ and $\lambda\varrho = 250$. (d) The reconstructed surface. (e) The 3–D plot of (d). (f) The output intensity image generated from (d) using the same set of parameters as in (c).

(a)

(b)



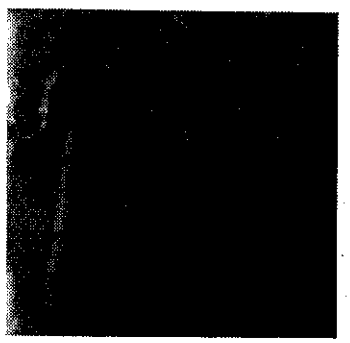
(c)

(d)

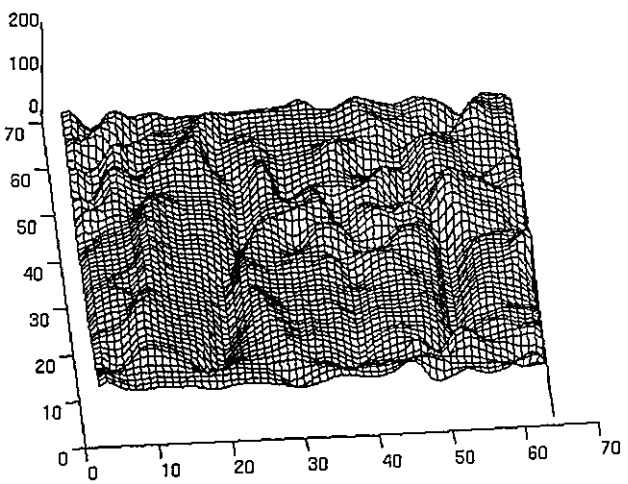
(e)

Figure 3. The SFS algorithm applied to the Lena image. (a) The input image. Its estimated reflectance map parameters are $\sigma_L = 59.52°$, $\tau_L = 7.74°$, $\lambda_Q = 192.01$ and $\xi = 3.0$. (b) The reconstructed surface. (c) The 3–D plot of (b). (d) The image generated from the reconstructed (p, q) maps with the same reflectance map parameters. (e) The image synthesized from the reconstructed (p, q) using the same set of reflectance map parameters except $\tau_L = 97.74°$.

(a)

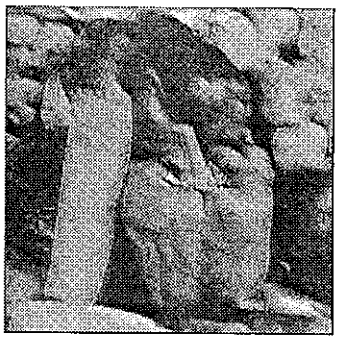


(b)



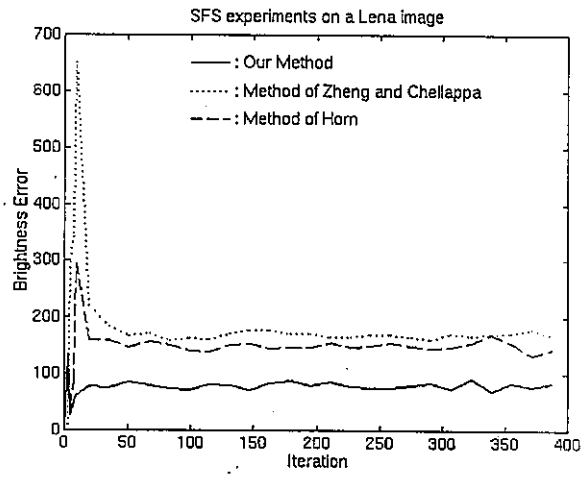
(c)

(d)

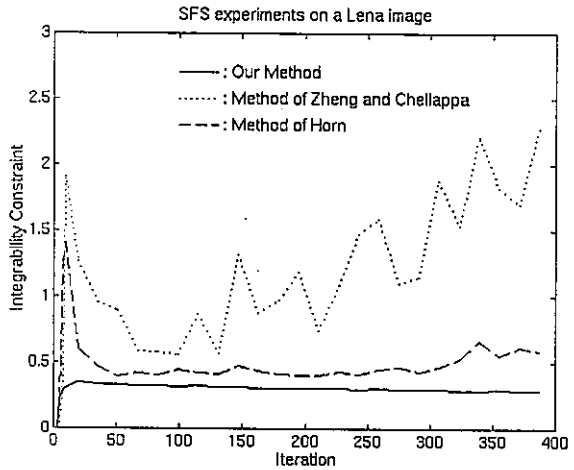


(e)

Figure 4. The SFS algorithm applied to a multi–pepper image. (a) The input image. Its estimated reflectance map parameters are $\sigma_L = 58.20°$, $\tau_L = 15.92°$, $\lambda\varrho = 255.96$ and $\xi = 0.0$ (b) The reconstructed surface. (c) The 3–D plot of (b). (d) The image generated from the reconstructed (p, q) maps with the same reflectance map parameters. (e) The image synthesized from the reconstructed (p, q) using the same set of reflectance map parameters except $\tau_L = 105.92°$.

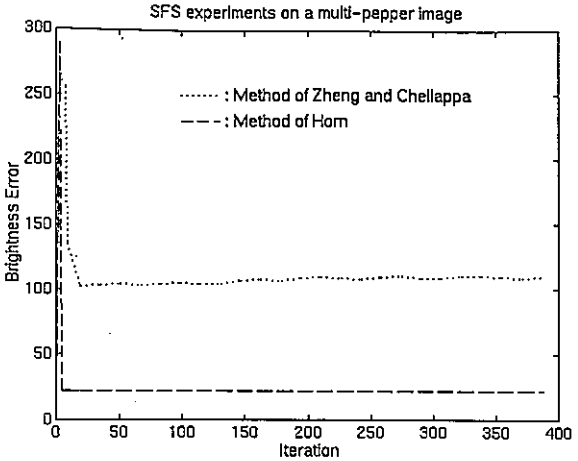


(a)

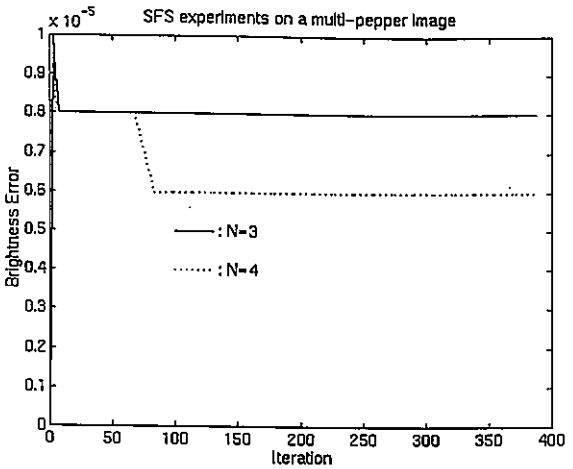


(b)

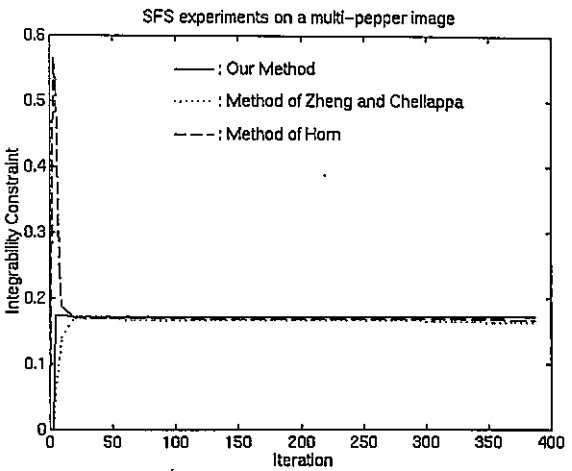
Figure 5. The comparisons of our SFS algorithm with two other algorithms based on the Lena image. (a) The results of adopting the average magnitude of the "brightness error" at each pixel as an error measure. (b) The results of adopting the average "integrability constraint" magnitude as a comparison measure.



(a)



(b)



(c)

Figure 6. The comparisons of our SFS algorithm with two other algorithms based on the multi–pepper image. (a) The average magnitudes of the "brightness error" generated by the other two algorithms. (b) The average "brightness error" of our algorithm. The scale of the vertical axis has been scaled down to 10^{-5}. Two curves in (b) are results obtained by different compactly supported basis (N=3, 4). (c) The results of adopting the average "integrability constraint" magnitude as a comparison measure.