A regularized solution of shape from shadows

MICHAEL HATZITHEODOROU
TOMASZ JACKOWSKI
ANARGYROS PAPAGEORGIOU

Department of Computer Science
Columbia University
New York, NY 10027
CUCS - 416 - 89

Abstract. We present a regularized solution to the shape from shadows problem. In this problem the shadows cast on an unknown surface yield data that can be used for the reconstruction of this surface. In the formulation presented here we assume that the data can now be perturbed by noise. It is shown that the regularized approach produces a solution that can handle noisy information while being very similar to the solution obtained by the approximation theoretic approaches used in earlier work. We provide implementation runs where the performance of the algorithm in recovering unknown surfaces is tested. Furthermore, we study the visual effects of smoothing on the various reconstructions.

1. Introduction

The shape from shadows problem has been addressed in a series of papers [3, 4, 6], where different formulations for its solution have been presented. In this problem, an unknown surface is reconstructed from the data that are derived from the shadows cast on it by a point light source located at a distance from this surface, see Fig. 1.

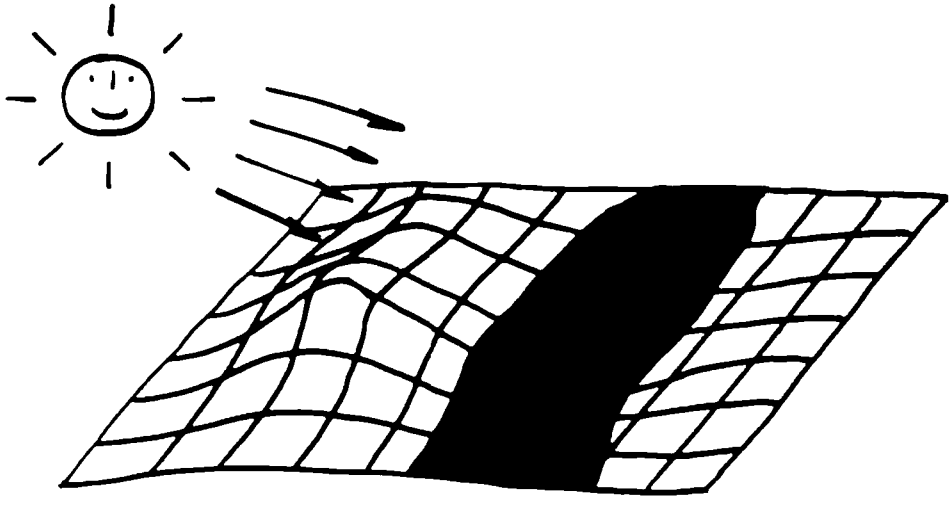

Figure 1.

Typeset by AMS-TEX

Shadows are a robust source of information. The process that uses shadows is not affected by texture or by surface reflectance. Furthermore, the imaging system does not need a grey scale or color capabilities; it is sufficient for it to be able to distinguish between black and white. Noise in the form of bright spots inside a dark area and vice-versa can be filtered out easily. Consequently, thresholding coupled to a simple histograming process will yield clean shadowed areas. It is thus evident that shadows can be very useful tools for the reconstruction process.

The data that can be obtained from the shadows include :

i. The gradient of the function at the shadow boundary closer to the light source (start of the shadow),
ii. The difference between the value of the function at any point in the closest boundary and the value of the function at any point in the remote boundary (end of the shadow).
iii. The fact that the unknown function will be underneath the plane defined by the light source and the shadow boundaries inside the area covered by them.
iv. The actual location of the shadow boundary.

We will analyze how to use the shadow data later.

In the work presented in [3, 4] the problem was solved assuming that the data are exact. Due to certain inadequacies of the imaging system the shadow data, and in particular the location of the shadow boundary, may not be recovered exactly. This type of noise can result in problems in the reconstruction processes and unsatisfactory performance of the employed algorithms.

To overcome these problems the principles of regularization theory will be employed. Regularization theory deals with methods for correctly defining ill-posed problems, and for selecting and imposing constraints in such a manner that the constrained problem becomes well-posed. The first definition of a well-posed problem, and consequently of an ill-posed problem is attributed to Hadamard [1]. Problems can become ill-posed due to the presence of noise, as is often the case in computer vision. Many researchers have studied methods for solving ill-posed problems in general [9, 10], and problems with noisy data in particular. For a survey see [2, 5, 7, 8].

In this paper we will define the problem using a regularization theory methodology. There is a variety of methods for solving regularization problems. We would like our solution to be as simple as possible, and if possible to be similar to the one presented in [3, 4]. We will show that this can be achieved.

The organization of the rest of this paper will be as follows. In the next section we will provide some basics from regularization theory on which we will base our analysis.

In section 3 we will formally define the problem, and in particular the space of functions and the information that we will need to extract from the shadows.

Section 4 contains the derivation of the regularized solution to the shape from shadows problem. This solution which we will call *smoothing spline* always exists and is unique regardless of the existence of noise.

Finally, in section 5, we will discuss the implementation of the proposed algorithm, and provide a series of test runs for simulated and real shadow data. The effects of the noise in a non-regularized solution will be shown and this will be compared to a regularized one.

2. Aspects of regularization theory

Assume that we have a metric space F that contains the unknown functions f that we want to recover. Also assume that we are given data $\vec{y} \in U$, where U is a normed metric space. A problem can be defined as an equation

$$S\, f = \vec{y}, \tag{2-1}$$

for some operator $S : F \longrightarrow U$.

Solving a problem defined in the above manner is equivalent to obtaining a solution to the equation (2-1). This may not always be realizable. The solution may not exist, may not be unique, or if S^{-1} is discontinuous, it may not be stable.

To solve the problem, one needs to define a non-negative continuous functional $T : F \longrightarrow \mathbb{R}_+$, called *stabilizing functional*. Various kinds of stabilizing functionals have been used. For example :

i. Norms or semi-norms on Hilbert spaces [13, 14, 15].

ii. The *Tikhonov stabilizers* [9, 10]

$$\|f\|_p^2 = \sum_{m=0}^{p} \left[\int_{\mathbb{R}^d} \lambda_m(x) \sum_{j_1+\cdots+j_d=m} \frac{m!}{j_1! \ldots j_d!} \left(\frac{\partial^m f(x)}{\partial x_1^{j_1} \ldots \partial x_d^{j_d}} \right)^2 dx \right]. \tag{2-2}$$

A solution to the problem (2-1) is required to minimize the *smoothing functional*

$$\theta^\lambda(f) = \|Sf - \vec{y}\|^2 + \lambda\, T(f)^2, \tag{2-3}$$

for some appropriately chosen $\lambda \geq 0$.

By requiring minimization of (2-3) we are essentially narrowing the class of possible solutions, thus achieving existence, uniqueness and stability. There are many methods for obtaining a solution to (2-1) that minimizes (2-3). A detailed analysis can be found in the original work of Tikhonov in [9, 10], and in a survey of this theory presented in [5].

The value of λ determines the weight between the two quantities of (2-3) that are to be minimized. For λ close to zero the method emphasizes fidelity to the data, otherwise referred to as *closeness of fit*. For large λ the method lays emphasis on the minimization of $T(\cdot)$. In most cases this T is a measure of smoothness, hence a large λ emphasizes smooth solutions.

In other words, one could regard λ as a measure of our trust in the quality of the data. If we believe that the data are not corrupted then we can choose a small λ thus requiring from the reconstruction to be close to the data. If our belief in the quality of the data is low, then we choose a large λ and we virtually smooth over the data and consequently over the noise. Since the quality of the data is not usually known, the appropriate choice of λ is not an easy problem. It can be chosen so that the obtained solution satisfies certain heuristic criteria. A method that is often used is *cross-validation*. It is based on a statistical analysis of the properties of the data. More information on cross-validation can be found in [15, 16, 17].

Regularization theory offers the ability to solve problems with noisy information. Unfortunately, the theory does not offer any methods that can be used to estimate the error of the resulting reconstructions, as is the case with the setting described in [3, 4]. It is thus obvious that there is a tradeoff in the choice of the method. If the noise in the image is negligible then it pays to solve the problem as proposed in [3, 4]. If not, as is the case in the assumptions for this paper, the regularization theory principles must be employed.[1]

3. Problem formulation

3.1 Function space – Definitions.

Let,

$$F = \left\{ f \mid f : [0,1]^2 \longrightarrow \mathbb{R},\ D^{1,1} f \text{ absolutely cont.},\ \|D^{2,2} f\|_{L_2} \leq 1 \right\}, \tag{3-1}$$

be the space that contains the functions f that we want to approximate. The quantity $D^{i,j}$ is defined as $D^{i,j}(\cdot) = \frac{\partial^{i+j}(\cdot)}{\partial x^i \partial y^j}$.

Define the bilinear form $\langle \cdot, \cdot \rangle$

$$\langle f, g \rangle = \int_0^1 \int_0^1 D^{2,2} f(x,y)\, D^{2,2} g(x,y)\, dxdy, \tag{3-2}$$

and the semi-norm $\| \cdot \| = \langle \cdot, \cdot \rangle^{1/2}$. F, equipped with $\| \cdot \|$ is a metric space.

Also define the stabilizing functional

$$T(f)^2 = \int_0^1 \int_0^1 \left(D^{2,2} f(x,y) \right)^2 dxdy. \tag{3-3}$$

3.2 Information.

In the next step we will extract from the image(s) the information, that is contained in the shadows, and which will be denoted by $N(f)$. We have already mentioned what types of information can be extracted from the shadows. It can be seen that this information exists at an infinite number of points. To sample at a finite number of points we have to adopt a sampling strategy. As mentioned in the introduction, assume that we have a light positioned far from the surface.[2] Also assume that the location of the light source is known, and that it is oriented in such a way that the light rays can be visualized as lying on planes parallel to the x-axis. We draw k imaginary lines perpendicular to the y-axis, and we sample along these lines. The entire setup is shown in Fig. 2.

At the intersection of each of the k lines with the shadowed areas we can obtain the following information. From the position of the light source, we can immediately obtain the partial derivative of the function f, with respect to x, at the point (x_i, y_i), (x_i, y_i)

[1] Although there are formulations of the approximation problem where the noise can be assumed to have certain properties and where error bounds and optimality results can be obtained. Please see [11, 12] and references therein for more information.

[2] This way we can assume that the light rays are parallel to each other.

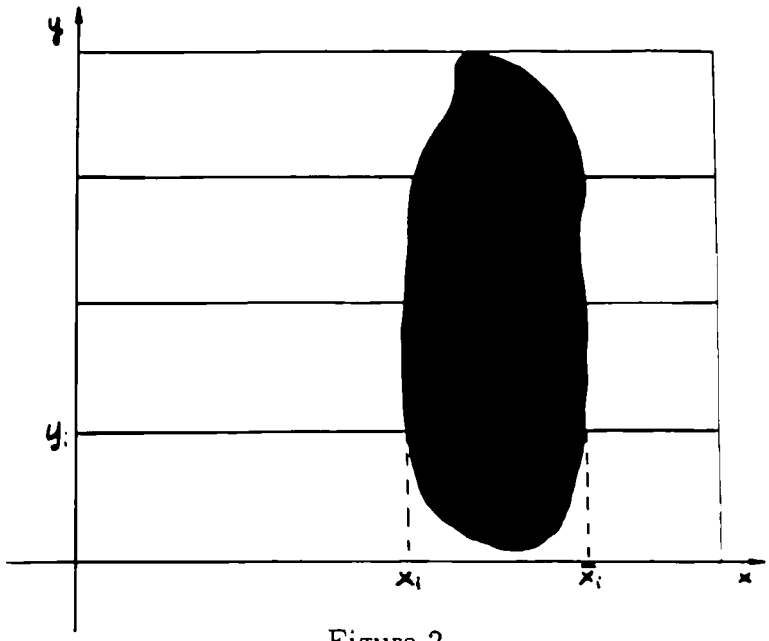

Figure 2.

being the beginning of the shadow (see Fig. 3). We can also obtain the difference between the two function values $f(x_i, y_i) - f(\bar{x}_i, y_i)$, at the beginning and at the end of the shadow respectively, given by $\frac{\partial f}{\partial x}(x_i, y_i)(x_i - \bar{x}_i)$.

For a light falling on the surface along the y-axis we can obtain similar information. In particular, for a given shadowed area starting at (x_i, y_i), and ending at $(x_i, \bar{y}_i)$, we can obtain the partial derivative of f with respect to y and the difference $f(x_i, y_i) - f(x_i, \bar{y}_i)$ which is given by $\frac{\partial f}{\partial y}(x_i, y_i)(y_i - \bar{y}_i)$.

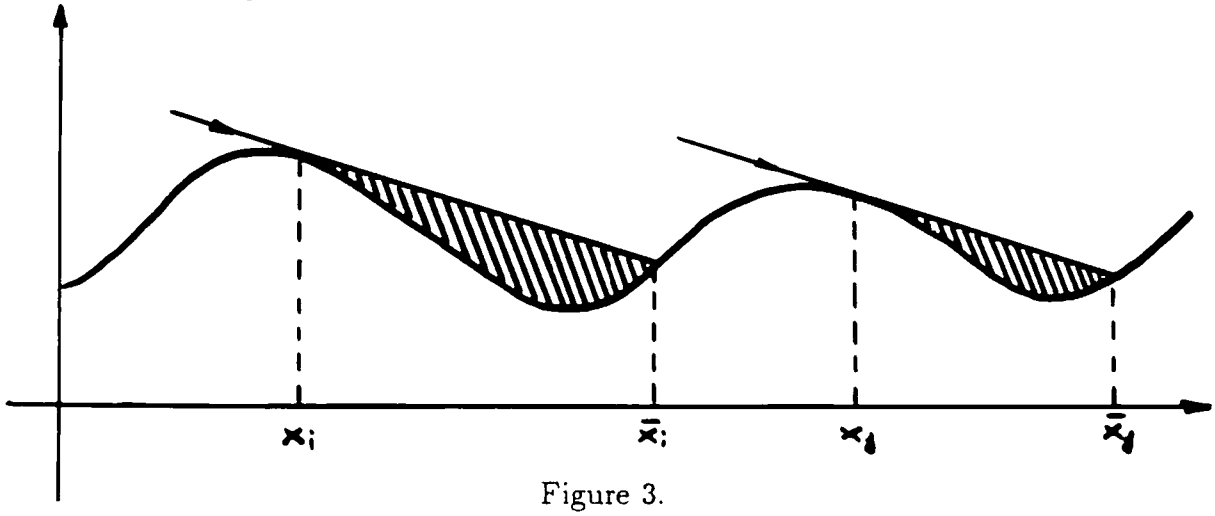

Figure 3.

So, for every intersection of a sampling line and a shadowed region we can obtain a data pair of the form,

$$\left\langle \frac{\partial f}{\partial x}(x_i, y_i), f(x_i, y_i) - f(\bar{x}_i, \bar{y}_i) \right\rangle, \quad \bar{y}_i = y_i \tag{3-4}$$

or of the form,

$$\left\langle \frac{\partial f}{\partial y}(x_i, y_i), f(x_i, y_i) - f(\bar{x}_i, \bar{y}_i) \right\rangle, \quad \bar{x}_i = x_i. \tag{3-5}$$

In each one of the images in our sample there are 0, 1 or more shadowed areas. From each one of those shadowed areas we can obtain a pair of the form (3-4) or (3-5). If we group all the data resulting from this sampling we obtain the vector,

$$N(f) = \left[\frac{\partial f}{\partial x}(x_1, y_1), \ldots, \frac{\partial f}{\partial x}(x_k, y_k), \frac{\partial f}{\partial y}(x_{k+1}, y_{k+1}), \ldots, \frac{\partial f}{\partial y}(x_n, y_n), \right.$$
$$\left. f(x_1, y_1) - f(\bar{x}_1, \bar{y}_1), \ldots, f(x_n, y_n) - f(\bar{x}_n, \bar{y}_n)\right]. \quad (3\text{-}6)$$

$N(f)$ is the data vector that contains the correct unperturbed data. These are unknown in our case. We are instead given the perturbed data vector $\vec{u}$.

Another piece of information can be obtained from the shadows. Assume that $l_i(x, y)$ is the straight line segment passing through the points (x_i, y_i), $(\bar{x}_i, y_i)$. Then, in this interval the function f will have values smaller than l_i, see Fig. 3. It holds,

$$f(x, y_i) < l_i(x, y_i), \quad \forall x \in [x_i, \bar{x}_i]. \quad (3\text{-}7)$$

If the light falls in the direction along the y-axis, then the obtained inequality is

$$f(x_i, y) < l_i(x_i, y), \quad \forall y \in [y_i, \bar{y}_i]. \quad (3\text{-}8)$$

It is feasible to incorporate information like the above in our sample as has been shown in [3, 4]. From actual test runs though it has been noted that the inequalities (3-7) and (3-8) usually hold without being explicitly incorporated in the sample. To simplify our analysis we will not use this non-linear information any further. The interested reader is referred to the above bibliography for further details.

4. Solution of the problem – The smoothing spline algorithm

To solve the problem we need to find a function in F that minimizes the smoothing functional

$$\theta^\lambda(f) = \|N(f) - \vec{u}\|^2 + \lambda\, T(f)^2. \quad (4\text{-}1)$$

It is known, see [11, 12] etc., that the spline σ satisfies $\|\sigma\| \leq \|f\|$, $\forall f \in F$, such that $N(f) = N(\sigma) = \vec{y}$.

Then (4-1) becomes

$$\begin{aligned}
\theta^\lambda(f) &\geq \|N(f) - \vec{u}\|^2 + \lambda\, \|D^{2,2}\sigma\|^2 \\
&= \|\vec{y} - \vec{u}\|^2 + \lambda \left\| \sum_{i=1}^{2n} a_i g_i \right\|^2 \\
&= \|\vec{y} - \vec{u}\|^2 + \lambda \sum_{i,j=1}^{2n} a_i a_j \langle g_i, g_j \rangle \\
&= \|\vec{y} - \vec{u}\|^2 + \lambda\, \langle \mathbf{G}\vec{a}, \vec{a} \rangle
\end{aligned}$$

$$= \|\vec{y} - \vec{u}\|^2 + \lambda \, \langle \vec{y}, \mathbf{G}^{-1} \vec{y} \rangle$$
$$= \sum_{i=1}^{2n} (y_i - u_i)^2 + \lambda \sum_{i,j=1}^{2n} c_{ij} y_i y_j, \tag{4-2}$$

where $\{g_i\}_{i=1,\ldots,2n}$ are the representers of the information $N(f)$. They have the property $\langle f, g_i \rangle = \frac{\partial f}{\partial x}(x_i, y_i)$ or $\frac{\partial f}{\partial y}(x_i, y_i)$, for $i = 1, \ldots, n$, and $\langle f, g_i \rangle = f(x_i, y_i) - f(\bar{x}_i, \bar{y}_i)$, for $i = n+1, \ldots, 2n$.

We want to minimize (4-2) with respect to $y_i, \forall i$. Note that $\theta^\lambda(\sigma)$ is actually a function of $\vec{y}$. The minimum is attained when $\partial\theta / \partial y_i = 0, \forall i$. Then,

$$\frac{\partial \theta}{\partial y_i} = 2\,(y_i + u_i) + \lambda\, 2 \sum_{j=1}^{2n} c_{ij} y_j = 0$$
$$\Longrightarrow \quad 2\, y_i + 2\, \lambda \sum_{j=1}^{2n} c_{ij} y_j = 2\, u_i$$
$$\Longrightarrow \quad 2\, \vec{y} + 2\, \lambda \mathbf{G}^{-1} \vec{y} = 2\, \vec{u}$$
$$\Longrightarrow \quad \mathbf{G}\vec{y} + \lambda\, \vec{y} = \mathbf{G}\vec{u}. \tag{4-3}$$

Solving the system (4-3) will yield the unknown vector $\vec{y}$ that minimizes (4-1). Consequently, the spline $\sigma = \sum a_i g_i$ interpolating the data $\vec{y}$ minimizes (4-2). To construct the spline algorithm we can substitute $\mathbf{G}\vec{a}$ for $\vec{y}$ in (4-3).[3] Then, (4-3) becomes

$$\mathbf{G}\,\mathbf{G}\,\vec{a} + \lambda\, \mathbf{G}\,\vec{a} = \mathbf{G}\,\vec{u}$$
$$\Longrightarrow \quad (\mathbf{G} + \lambda\, \mathbf{I})\, \vec{a} = \vec{u}. \tag{4-4}$$

(4-4) is a system of equations with unknown $\vec{a}$. By solving it we can obtain the spline algorithm $\varphi^{ss}(x, y) = \sum_{i=1}^{2n} a_i g_i(x, y)$.

REMARK 4.1. *For $\lambda = 0$ the algorithm φ^{ss} interpolates the unperturbed data vector $\vec{y}$. In this case φ^{ss} coincides with the optimal spline algorithm proposed in [3, 4].*

REMARK 4.2. *By formulating the regularization problem in such a way, we managed to obtain the same solution from two entirely different theoretical formulations. We can therefore claim that in this setting, and for $\lambda = 0$, the regularization theory approach coincides with the approximation theoretic one.*

For $\lambda \neq 0$ the algorithm φ^{ss} does not interpolate the data $\vec{u}$ but passes close to them, how close depending on the magnitude of λ. This has a smoothing effect and for this reason φ^{ss} is called *smoothing spline algorithm.*

[3] $\mathbf{G}\vec{a} = \vec{y}$ follows from $\langle \sigma, g_i \rangle = y_i, \forall i$. This was also used in the derivation of (4-2).

5. Application of the Algorithm - Numerical Runs

5.1 Algorithm implementation.

To implement the smoothing spline algorithm we must obtain the values of the coefficients a_i. This is done by solving the system of equations (4-4) where $\mathbf{G}$ is the Gramm matrix $\{\langle g_i, g_j \rangle\}_{i,j=1}^{2n}$ and $\{g_i\}_{i=1,\ldots,2n}$ are given by (I-1), (I-2), (I-3) and (I-4) of the Appendix. The system is solved by a direct method without the need for pivoting since it is full, symmetric, and positive definite.

5.2 Test runs.

We have constructed a broad series of functions, and have run the algorithm using these as test surfaces. Simulated shadows have been created using a computer program. This program is given the mathematical formula for the function that will act as the unknown surface and a series of light angles. It returns the locations of the beginning and the end of the simulated shadow and the partial derivatives of the function at that point. The test function is consequently forgotten.

From early test runs, we have observed that smooth functions can be approximated easily with almost non-observable error, using a small number of sample points. The functions that are the most difficult to approximate, are the ones that have as low regularity as possible. In the class F these functions are piecewise quadratic polynomials which are constructed as the product of quadratic polynomials of one variable. We will show the performance of the algorithm φ^{ss} on these functions.

We start the series of test runs with a function consisting of 100 polynomial pieces. We use two different light angles from each direction, two along the x-axis and two along the y-axis. The function and the obtained reconstruction can be seen in Figure 4.

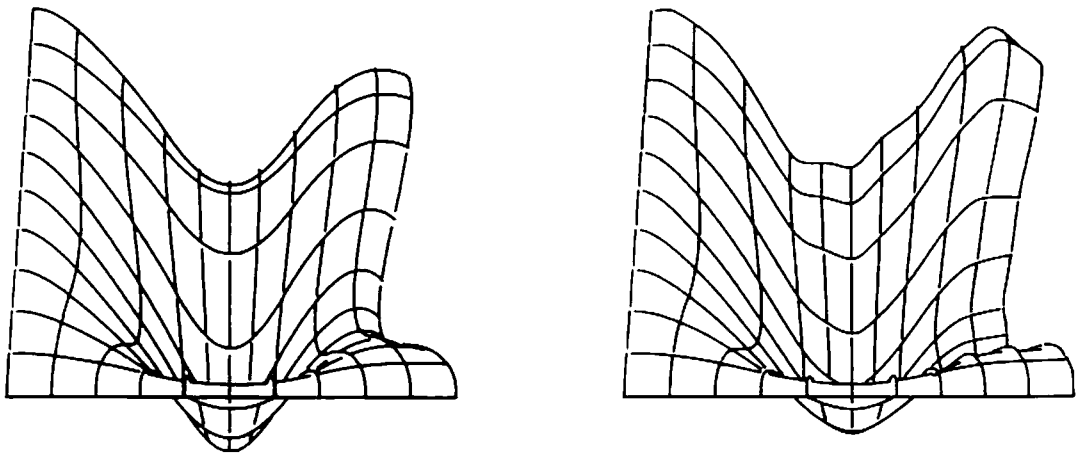

Figure 4.

In Figure 5 we show a function consisting of 200 polynomial pieces. We again draw the reconstruction together with the function for comparison purposes. The information was obtained by using 4 different lighting angles in each direction.

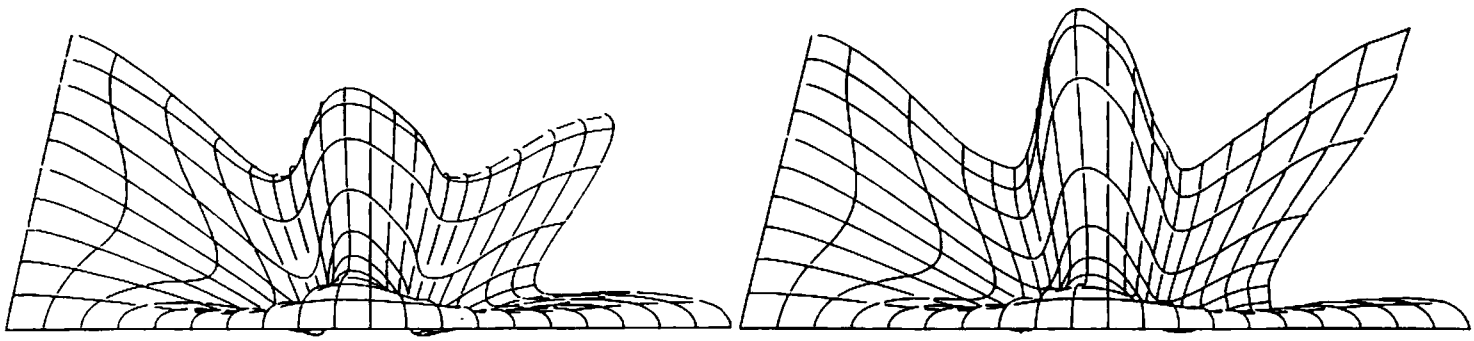

Figure 5.

Finally, in Figure 6 we show one of the most difficult functions that we have constructed. It consists of 400 polynomial pieces with large jumps in the second derivatives.[4] The function has been approximated using a sample created by using six different lighting angles in each direction.

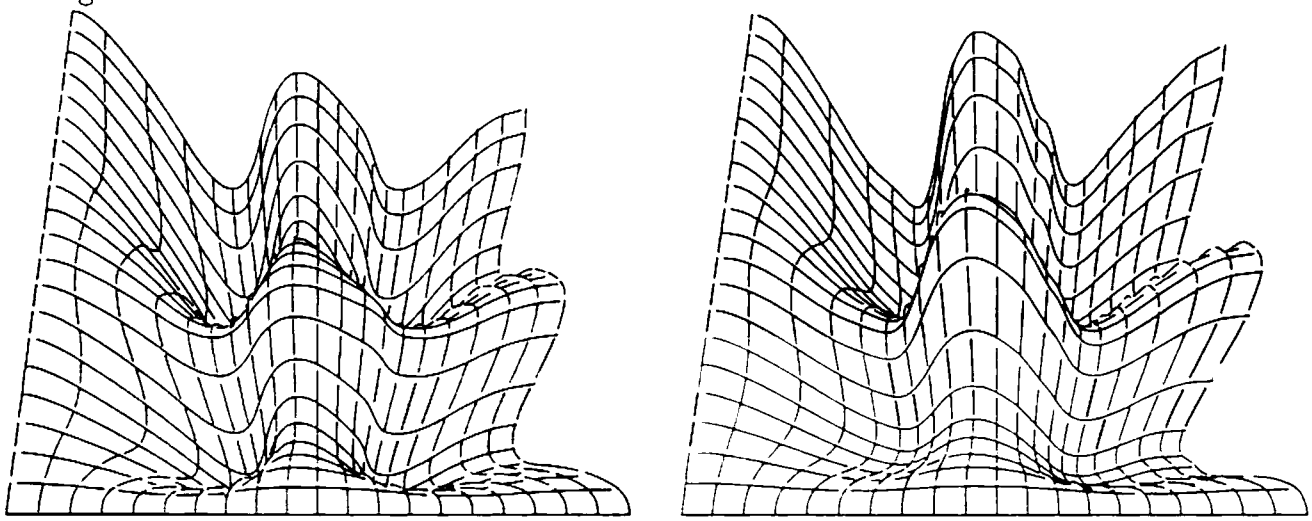

Figure 6.

As a next step we experimented with the application of our algorithm on actual images. A surface that could belong in the space F was constructed from wood and cardboard is shown in Fig. 7. The camera was placed at a considerable distance from the surface, to avoid perspective distortions. We used a spotlight placed also at a distance, to simulate a point light source. The light was moved in different locations to create different shadowed areas, and for every new position of the light we took a 256 by 256 image. The camera remained stationary throughout the experiment.

For the particular test runs that we will be showing we used four different placements of the light. Each of the obtained images is like the one shown in Fig. 8.

[4] Which means that $\|D^{2,2}f\| \leq 1$ does not hold. Instead, $\|D^{2,2}f\| \leq C$, for large C holds. The mathematical formulation does not change, but the visual effect is noticeable.

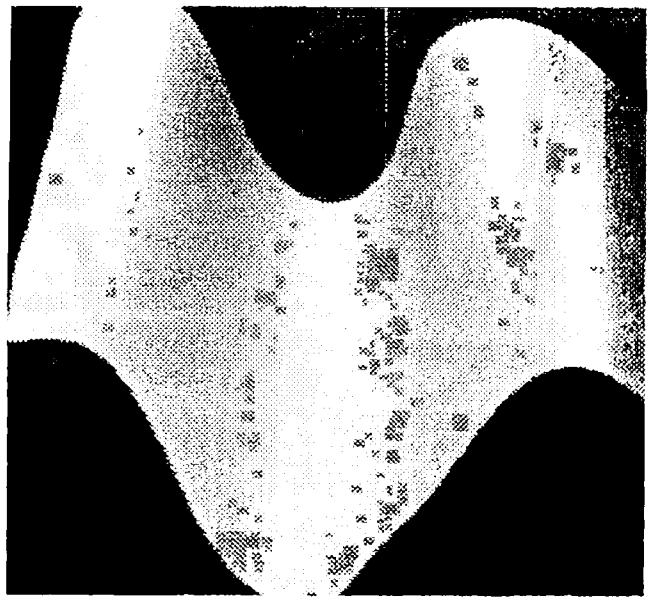

Figure 7.

Figure 8.

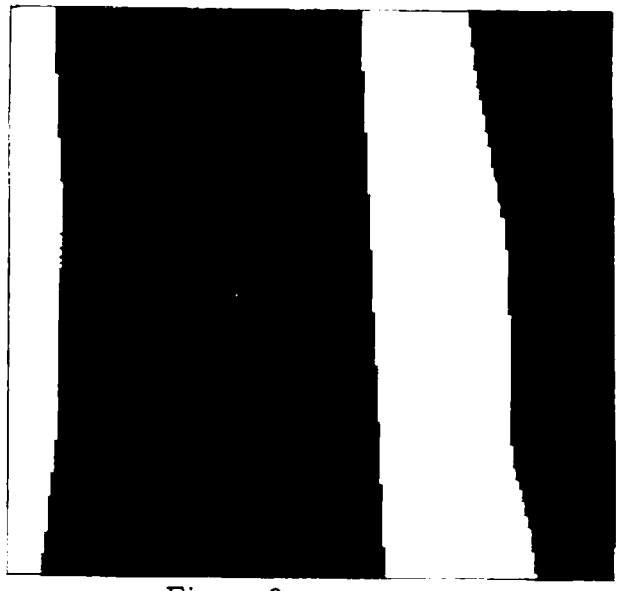

Figure 9.

Consequently, the intensity histogram of each one of the images was obtained, and they were thresholded accordingly. A simple filtering algorithm was finally applied. One of the resulting thresholded, filtered shadow images is shown in Fig. 9.

Shadow data were obtained from each of the four filtered images. Those were fed to the algorithm and resulted in the reconstruction shown in Figure 10.

5.3 The smoothing effect.

We would like to show here the effect of smoothing on the reconstructions, and also contrast the above results with ones that were produced without any smoothing ($\lambda = 0$).

The effect of the noise on the data is more apparent as the size of the sample increases. We will also intensify it by producing data with low numerical accuracy. In Figure 11 we present a reconstruction to the surface of Figure 4 obtained without any smoothing. The effect of the noise is observable as a jaggedness at the back of the reconstruction. Apparently by increasing the value of λ, the effects of the noise decrease and for $\lambda = 0.5$

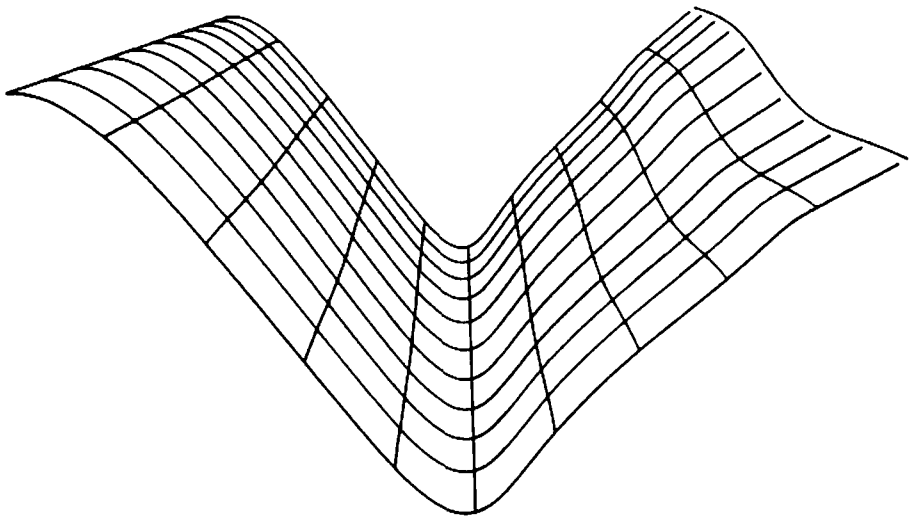

Figure 10.

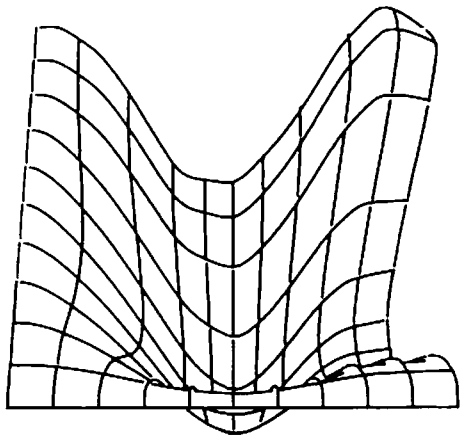

Figure 11

the resulting approximation does not show any noise effects, see Fig. 11.

Further increasing the value of λ will result in an even smoother reconstruction, see Fig. 12, which will not be very close to the data. As discussed in section 2 the choice of a *good* value for λ is not always obvious.

6. Conclusion

In this paper we showed how to obtain a regularized solution to the shape from shadows problem. This approach is useful whenever there is strong evidence that the data is noisy and will therefore create serious problems in the reconstruction obtained through previously studied methods.

Since the amount of noise may not always be known we produced a solution that is similar to the approximation theoretic one. The two solutions become identical in the absence of noise in the data. An advantage of our approach is that we can show that two seemingly different theoretical frameworks that have been used often for the solution of computer vision problems can yield similar solutions.

Following the theoretical formulation of the problem, we presented a series of test runs.

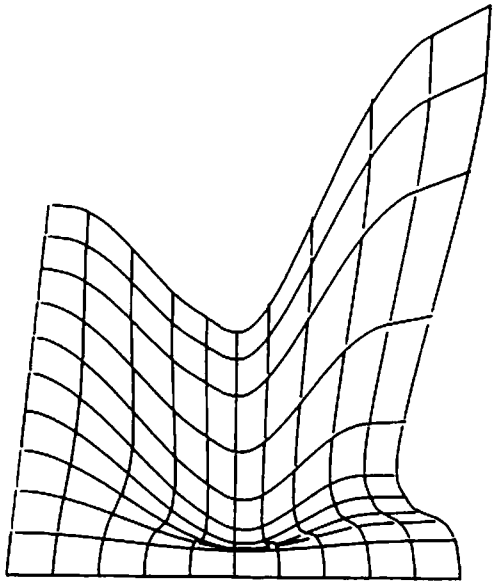

Figure 12

We showed the effect of the noise on the reconstruction if no smoothing is allowed. Consequently a series of test runs from simulated and from real shadow data were performed and the resulting reconstructions were presented along with the original surfaces.

Appendix I. - The Representers of the Information

The representers of the information $N(f)$ are given by :

$$D^{2,2}g_i(x,y) = \frac{(x_i - x)_+^0(y_i - y)_+ - (x_{i-1} - x)_+^0(y_{i-1} - y)_+}{\sqrt{x_i - x_{i-1}}}, \qquad i = 1,\ldots,k \qquad \text{(I-1)}$$

when the light falls along the x-axis, and

$$D^{2,2}g_i(x,y) = \frac{(y_i - y)_+^0(x_i - x)_+ - (y_{i-1} - y)_+^0(x_{i-1} - x)_+}{\sqrt{y_i - y_{i-1}}}, \qquad i = k+1,\ldots,n \quad \text{(I-2)}$$

for light along the y-axis, where $(a-b)_+^0 = 1$ for $a = b$ and 0 otherwise,

$$D^{2,2}g_{n+i}(x,y) = (\bar{x}_i - x)_+(\bar{y}_i - y)_+ - (x_i - x)_+(y_i - y)_+ - (x_i - x)_+^0(\bar{x}_i - x_i)(y_i - y)_+, \quad i = 1,\ldots,k \quad \text{(I-3)}$$

$$D^{2,2}g_{n+i}(x,y) = (\bar{y}_i - y)_+(\bar{x}_i - x)_+ - (y_i - y)_+(x_i - x)_+ - (y_i - y)_+^0(\bar{y}_i - y_i)(x_i - x)_+, \quad i = k+1,\ldots,n \quad \text{(I-4)}$$

where $(a-b)_+ = a - b$ for $a > b$ and 0 otherwise.

REFERENCES

[1] Hadamard, J., *Sur les problèmes aux derivées partielles et leur signification physique*, Princeton Univ. Bulletin **13** (1902).

[2] Hatzitheodorou, M. G., *The application of approximation and complexity theory methods to the solution of computer vision problems*, Computer Science dep., Columbia University, Tech Rep. (1988).

[3] Hatzitheodorou, M. G., *2-Dimensional shape from shadows. A Hilbert space setting*, Computer Science dep., Columbia University, Tech Rep. (1988).

[4] Hatzitheodorou, M. G., and Kender J.R., *An optimal algorithm for the derivation of shape from shadows*, Proceedings, IEEE conference on computer vision and pattern recognition (1988).

[5] Jackowski, T., *Techniques for solving ill-posed problems in computer vision*, Computer Science dep., Columbia University, Tech Rep. (1989).

[6] Kender, J., and Smith, E., *Shape from darkness. Deriving surface information from dynamic shadows*, Proceedings, AAAI (1986).

[7] Poggio, T., *Computer vision*, Artificial Inteligence Lab., MIT, Tech. Rep. (1986).

[8] Poggio, T., and Torre, V., *Ill-posed problems and regularization analysis in early vision*, Artificial Inteligence Lab., MIT, Tech. Rep. (1984).

[9] Tikhonov, A. N., *Solution of incorrectly formulated problems and the regularization method*, Soviet Math. Dokl. **4** (1963).

[10] Tikhonov, A. N., and Arsenin, V. Y., "Solutions of ill-posed problems," V.H.Winston and Sons, 1977.

[11] Traub, J. F., Wasilkowski, G., and Woźniakowski, H., "Information-based complexity," Academic Press, 1988.

[12] Traub, J. F., and Woźniakowski, H., "A general theory of optimal algorithms," Academic Press, 1980.

[13] Wahba, G., *Practical approximate solutions to linear operator equations when the data are noisy*, SIAM J. Numer. Anal. **14**, No 4 (1977).

[14] ———, *Surface fitting with scattered noisy data on Euclidian D-space and on the sphere*, Rocky Mountain Journal of Mathematics **14**, No 1 (1984).

[15] ———, *Cross-validated spline methods for the estimation of multivariate functions from data on functionals*, Proceedings, 50th anniv. conference, Iowa Statistical Lab., The Iowa State Univ. Press (1984).

[16] Wahba, G., and Wendelberger, J., *Some new mathematical methods for variational objective analysis using splines and cross-validation*, Monthly Weather Review **108** (1980).

[17] Wendelberger, J., *Smoothing noisy data with multidimensional splines and generalized cross-validation*, Department of Statistics, University of Wisconsin-Madison, Ph. D. thesis (1982).