McGill University

Electrical and Computer Engineering

Shape Analysis

Term Project

Shape from Shading Using Level Sets Method

Name: Mohannad Elhamod

Student ID: 260303471
Contents

I. Introduction ......................................................................................................................................................................... 3

II. Background ........................................................................................................................................................................ 3

III. The Theory ........................................................................................................................................................................ 3

   i. Problem Formulation ...................................................................................................................................................... 3

   ii. Assumptions .................................................................................................................................................................... 3

   iii. The Simple Case (Vertical Light Source) ..................................................................................................................... 4

   iv. The General Case ............................................................................................................................................................. 4

IV. Solution Formulation ....................................................................................................................................................... 5

   i. Solving The Simple Case Using Level Sets ....................................................................................................................... 5

   ii. Solving The General Case Using Level Sets ..................................................................................................................... 5

   iii. The Slope Limiter ......................................................................................................................................................... 6

   iv. The Initial Conditions .................................................................................................................................................... 6

   v. Implementation Details .................................................................................................................................................. 7

V. Examples and Results .......................................................................................................................................................... 9

VI. Conclusion ...................................................................................................................................................................... 12

VII. References .................................................................................................................................................................... 13
I. Introduction

Extracting the shape of an object from its image(s) is a problem that takes place in several subfields of computer vision. Several approaches tackle this problem from quite different perspectives. Some of those perspectives are “Shape from stereo”; “Shape from texture” and “Shape from Shading” [1]. Obviously, no single approach is enough to fully extract the topology of an object/scene from its image(s). The human brain appears to combine all these factors to make inferences about the shape.

In this paper, I intend to show the effectiveness of “Shape from Shading” in this area. The method adopted in this paper is level sets that embed active contours. The theory and implementation of this project are heavily based on the work done by Kimmel et al in [2] and [3].

After discussing the theory, the implementation is carried out in Matlab, and results are shown. Comments are mentioned both throughout the report and in the conclusion.

II. Background

The “Shape from Shading” problem has been tackled in many ways in the literature. Those methods can be categorized into iterative and non-iterative [2]. The iterative methods are based on minimizing both smoothness and brightness error; however, this is not guaranteed to give the desired solution and might not converge. Some of the non-iterative methods assume each point to be on a sphere. This assumption however is not always valid. There are other methods that sometimes give results as good as those in the proposed level-sets method, such as the viscosity solution, which discussion is beyond the scope of this project.

The level sets method has the advantage of dealing with shocks in a graceful way [2].

III. The Theory

i. Problem Formulation

“Shape from Shading” refers to reconstructing the 3D figure from its 2D image. The 2D image is captured by means of the light scattered from the object(s) in the scene, which is originally coming from a light source or different light sources. The image in this case is simply a gray-scale image, where the brightness of each pixel represents the amount of light received from its corresponding point in the 3D scene.

Reconstructing the 3D figure means inferring information about the depth/height of each point in space only from the brightness information of pixels in the top view image. In a nutshell, this is simply reduced to finding the two components of the 3D surface normal [2].

In this project, the level sets method is discussed in details. This method is based on the Huygens’s principle and the entropy condition, and uses the Hamilton-Jacobi equation to solve the problem.

ii. Assumptions

To simplify the problem and make it feasibly solvable, several requirements are enforced. These requirements however are not too harsh, and are acceptable in a real-life image. However, this again doesn’t mean that “Shape from Shading” is sufficient to universally solve any “Shape from X” problem.

The assumptions are:

a) We have one point-like light source that is infinitely far. This absolutely conforms to outdoor images, where the sun is the only dominant light source.

b) The objects in the scene are lambertian. They reflect light equally in all directions. This is also a safe assumption in many images, unless the image contains some reflective surface such as mirrors, water bodies and snow.
iii. The Simple Case (Vertical Light Source)

The simple case adds an additional restriction of having the light source vertical to the scene. Let’s assume that we have a light source that’s vertical to the scene, with the vector \( \hat{l} = (0, 0, 1) \) as its orientation. This specifies the direction of the light source and not its absolute position, because we assume the light source to be at infinity.

Let \( p \) and \( q \) be the \( x \) and \( y \) components of the gradient on the 3D surface. The unit normal to the surface can be written as

\[
\hat{N} = \frac{\langle -p, -q, 1 \rangle}{\| \langle -p, -q, 1 \rangle \|} = \frac{\langle -p, -q, 1 \rangle}{\sqrt{1 + p^2 + q^2}}
\]

There is a relation between the intensity image and the gradient components [3]. For the lambertian surface case, we know that

\[
E(x, y) = R(q, p) = \hat{l}.\hat{N} = \frac{1}{\sqrt{1 + p^2 + q^2}}
\]

Where \( E \) is in the interval 0 to 1. This equation corresponds to an ambiguity cone with an elliptic base in space, and its two radii related to \( p \) and \( q \). The line segment connecting the tip of the cone and any point on the rim of the base is of a unity length. This is shown in Figure 1.

We need to find another equation to resolve this ambiguity. What we care about for solving the problem using level sets is \( \hat{n} \), the projection of the unit normal \( \hat{N} \) on the plane of the active contour, i.e. the zero level set.

Let the angle between \( \hat{l} \) and \( \hat{N} \) be \( \alpha \). Then, we can write

\[
E = \hat{N}.\hat{l} = \cos(\alpha) = \frac{1}{\sqrt{1 + p^2 + q^2}}
\]

Climbing a height of \( dz \) on the surface corresponds to moving a distance \( D \) along the projection of \( \hat{N} \) on the zero level set, i.e. along \( \hat{n} \). This can be written as

\[
D = dz \cot(\alpha) = \frac{dz}{\sqrt{p^2 + q^2}} = dz \frac{E}{\sqrt{1 - E^2}}
\]

And from this, a solution can be found by solving a nonlinear initial value PDE problem [3]. Despite the fact that this method, and any other method, claim to solve the “Shape from Shading” problem, no single method is able to resolve the +/- ambiguity. This is an intrinsic characteristic of the “Shape from Shading” problem [2].

iv. The General Case

In the more general case, the condition of a vertical light source is relaxed, and hence the light source orientation can be written in the form \( \hat{l} = (p_l, q_l, 1) \), where \( p_l \) and \( q_l \) are the \( x \) and \( y \) components of the light source orientation vector.

Following the same reasoning in the previous section, we can write

\[
E = \hat{N}.\hat{l} = \cos(\alpha) = \frac{1 + p \cdot p_l + q \cdot q_l}{\sqrt{1 + p^2 + q^2} \sqrt{1 + p_l^2 + q_l^2}}
\]

The solution to this equation in 3D space is a tilted cone as shown in Figure 2.

However, the procedure becomes hairy in this case, and its discussion is beyond the scope of this project.
IV. Solution Formulation

i. Solving The Simple Case Using Level Sets

Assuming the active contour to be embedded in an embedding function $\psi$, with the negative values inside the contour and the positive values outside, we can write

$$\psi_t = \frac{E}{\sqrt{1 - E^2}} \sqrt{\psi_x^2 + \psi_y^2}$$

Using this update method, the embedding function, and hence the zero-level contour, will be updated every time step $\Delta t$. However, the “Shape from Shading” problem is slightly different from the problems of area and length minimization, in the sense that as the zero-level active contour is updated, the pixels belonging to that contour have their height/depth assigned in the 3D reconstruction. This reconstruction is described by the formula

$$h^i = \left( -t + \frac{\psi_i}{\psi_t} \right)$$

Where $h^i$ represents the height/depth of pixel $i$. We can clearly see that time is an inherent factor in the height assignment equation. This actually indicates that different time steps lead to figures of different proportions, yet of the same shape, as will be shown later.

We notice in the first equation that the factor $\frac{E}{\sqrt{1 - E^2}}$ suffers from two problems. The first problem is that when the intensity of the pixel goes to 1, i.e. absolutely white, the factor goes to infinity, and hence the update becomes unstable; the second problem occurs when the intensity goes to zero, i.e. absolutely black, the time update becomes zero, and $\psi^i$ freezes. These two extremes are critical points and need to be dealt with. The proposed solution is to threshold the image on both sides of the intensity spectrum.

ii. Solving The General Case Using Level Sets

The general case has a similar solution, with its update and height equations being more complicated. The update equation for the embedding function is

$$\psi_t = \frac{1}{\sqrt{1 + q_t^2 + p_t^2}} \cdot \frac{E}{\sqrt{1 - E^2}} \cdot \left( \frac{\psi_x^2 (1 + q_t^2) + \psi_y^2 (1 + p_t^2) - 2p_t q_t \psi_x \psi_y - (p_t \psi_x + q_t \psi_y)}{\sqrt{1 + q_t^2 + p_t^2}} \right)$$

The height equation becomes

$$h^i = \left( -t + \frac{\psi_i}{\psi_t} \right) \sqrt{1 + q_t^2 + p_t^2} + (x^i p_t + y^i q_t)$$

In the general case, the level sets become tilted in planes perpendicular to the light source vector as shown in Figure 3. Notice in Figure 3 that the topology, in both the special and general cases, is defined in terms of planes that are perpendicular to the light source direction, e.g. $p^l$, which is not parallel to the plane perpendicular to the viewing direction, e.g. $P$, in general.

Considering this observation more carefully, and looking at Figure 4. We notice that the “Shape from Shading” approach, or any other approach for what it matters, cannot infer any information about any surface that lies behind a point that has a tangent plane parallel to the light source direction. The region will simply be hiding in the shadow of another surface, and hence information about it is lost. This self occlusion dilemma is an inherent problem, and nothing can be done about it. Notice that all the possible surfaces shown after point $j$ in Figure 4 will give the same top view image, because the shaded area will appear dark due to self occlusion, and the topology information is lost. This suggests that the absolute black color is
always related to lack of information, and that explains the unstable behaviour of the algorithm close to the black end of the gray-scale spectrum, as will be shown later.

### iii. The Slope Limiter

The reconstruction of the 3D surface is a delicate process, where the smoothness of the surface has to be preserved. However, because finite time steps are used, and because there is always some noise in the data; the smoothness of the reconstructed surface will most likely be violated, and singularities will form.

To overcome this problem, Sethian [4] proposes using the slope limiter approximations for the spatial derivatives instead of the central difference schemes. These derivatives are summarized in the following equations.

\[
\psi_x = \text{minmod}(D_x^-, \psi, D_x^+ \psi) \\
\psi_x^2 = \max(D_x^- \psi, D_x^+ \psi, 0)^2
\]

\(D_x^-\) and \(D_x^+\) refer to the derivatives from left and right respectively, and the \text{minmod} function is defined as

\[
\text{minmod}(a, b) = \begin{cases} 
\text{sign}(\min(|a|, |b|)) & \text{if } ab > 0 \\
0 & \text{otherwise}
\end{cases}
\]

The equations are approximations of the first derivative and its square in the x direction. Similar approximations are used in the y direction.

The simple idea behind these approximations is enforcing the surface to change slowly, and suppressing any abrupt changes in it. For example, it is highly unlikely to have the gradient ascending at a pixel, and then descending in the neighbour pixel. This abrupt change forms a spike on the surface. The slope limiter approximations smooth these spikes and hence prevent the development of singularities. Figure 5 simplifies the slope limiter principle, where \(i\) is a pixel.

Many forms of the slope limiter approximations can be found in [5]. However, the \text{minmod} limiter is the only one used here.

### iv. The Initial Conditions

To start the process, we first have to specify the zero-level isophote of the 3D reconstruction by initializing the zero-level of the embedding function. [3] suggests defining the initial height to be the contour that separates the singular areas, i.e. pixels with brightness \(E_{\text{max}} < E \leq 1\), from the rest of the pixels. However, this method has a major deficiency if not used with caution. It can only be used when all the singular areas in the top view are supposed to have equal heights, because the algorithm assumes all the pixels belonging to the initial zero-level contour to be of the same height, which is not usually the case, as shown in Figure 6.

Figure 6 illustrates that singular areas (in red) are not necessarily of the same height. In fact, one of the regions is a peak, while the other is a pit. This calls attention to the importance of some minimal knowledge about the topology of the surface to be reconstructed, and proves the incompleteness of the “Shape from Shading” problem.

Another more effective way for defining the initial height is by setting it to enclose only one of the singular areas, while everything around this contour would have positive value.

Another effective way to do it is to set the initial height contour to enclose the whole top view image (almost at the edges). This requires the edges of
the image to belong to the “background” of the scene, where the surface becomes almost flat, and hence requires the real surface to be reconstructed to be completely in the middle region of the image. This method is usually proven to be the best, and is usually the one used in the examples.

v. Implementation Details

Although the algorithm seems straightforward, some subtleties have to be discussed. Choosing the right time step is a quite delicate part of defining the initial parameters. If the time step is chosen to be big (larger than 0.5 in my implementation), the embedding function will be unstable. However, choosing large time steps has proven not to be disastrous because each pixel is only updated once, and hence even if the embedding function becomes unstable after a while, what really matters is that the update of the embedding function at each pixel is stable only until it becomes part of the zero-level contour, and so its corresponding height on the reconstructed surface is assigned. After that, the value of the embedding function at that pixel becomes irrelevant as the height is already assigned. Nevertheless, it’s still preferable to use relatively small time steps in order to avoid any instability before the height of the pixel is assigned. Clearly, on the other hand, too small time steps are not preferable, especially that the process becomes very slow at the dark regions due to the factor.

The effect of different time steps is illustrated in the following example. Figure 7 shows several 3D reconstructions of the same image with different time steps. This clearly demonstrates the direct dependence on time in the “Shape from Shading using level sets” problem, rather than it just being a speed factor. Here, it actually changes the proportions and size of the 3D reconstruction, while preserving the general structure (notice how the height of the reconstructed surface is 405, 380, 112 and 75 for time steps 0.05, 0.1, 0.2 and 0.3 respectively).

![Figure 7 – The effect of changing the time step (0.05, 0.1, 0.2, 0.3)](image)

It is worth noting that using slower time steps is actually more accurate. This is because more pixels are hastily updated when the process is faster, and hence the reconstructed surface will be less detailed. Very similar results are obtained by changing the elbido/light scattering constant $\rho$, which is a material property. If $\rho$ is plugged in, the intensity factor becomes $\frac{E}{\sqrt{1-E^2}}$. $\rho$ is a positive number that’s less than 1. It’s assumed to be 1 by default. It's worth mentioning that, generally, not all the parts of the image have the same $\rho$, and if $\rho$ for those parts is drastically different, the reconstructed surface might not make much sense.
Speeding up the process is achieved by assigning all pixels with \( E_{\min} > E \geq 0 \) the constant value \( E_{\min} \). This first solves the problem of update freezing due to the factor \( \frac{E}{\sqrt{1-e^2}} \), and ensures a minimum acceptable speed at dark regions. This is an acceptable solution because, usually, regions that are too dark suffer from height inaccuracy and surfaces become quite unrepresentative there. This is shown in the following example.

Figure 8 shows running the same algorithm for the same image for different \( E_{\min} \). Different black color thresholds give rise to different elevations. This is because the darker an area, the steeper the surface would be, and hence the surface would reach higher elevations. Similar observations can be noticed for different white color thresholds.

![Figure 8 - Different output for different black color thresholds (0.05, 0.1)](image)

After setting the initial conditions, the image is smoothed with a small size Gaussian (\( \sigma = 1 \)). This of course gets rid of the noise and imperfections in the image. However, for low resolution images, this will truncate some details, and hence the reconstruction will not be so accurate.

Mathematically, each pixel is assigned its elevation when its corresponding value in the embedding function becomes zero. However, this can’t be realized in a discretized update algorithm, for the fact that the value is very unlikely to become exactly zero. To solve this problem, the elevation assignment is carried out when the corresponding value in the embedding function changes sign. The change in sign gives evidence that the pixel actually passed through the zero level set if time were to be continuous.

The algorithm terminates when all the values of the embedding function become of the same sign, which ensures that all pixels have been updated.

The edges of the brightness image are not updated as they don’t have enough neighbourhoods to calculate the derivatives. Their pixels are always assigned the values of their closest inner neighbours.
V. Examples and Results

The following few examples will show both the strengths and weaknesses of “Shape from Shading using level sets”. Comments will accompany each example to illustrate those aspects.

The first example clearly demonstrates the +/- ambiguity mentioned earlier. Figure 9 shows the resultant reconstructed surface of a brightness image that can be seen by us, humans, in two different ways. If one stares at the brightness image, he/she could see it as a surface with two pimples, a surface with two dimples, or a surface with one dimple and one pimple. The output actually depends on the initial zero level. It can be clearly seen that all of the cases show almost exactly the same top view if the same vertical lighting is projected on the surface. This proves the “duality” of “Shape from Shading”, and that the +/- ambiguity is an inherent problem. For panoramic views of the examples below, please refer to my website [6]. This example also demonstrates the incompleteness of the algorithm and the importance of user-defining of the initial zero level in order to get the desired results.

![Brightness Image](image)

**Figure 9 - Illustrating how multiple solutions exist for the same brightness image**

In the second example, the importance of knowing the light angle with respect to the viewing angle beforehand is illustrated. Using the same brightness image in Figure 10 with 4 different light angles as inputs to the algorithm, the 3D reconstruction output turns out to be quite different, and yet amazingly, still manages to give roughly the same top view. Notice how in the first case (where the light source is assumed to be vertical), the algorithm had to form singularities on the surface to achieve the solution, i.e. the top view, implied by the brightness image. Still, the top view in the first case shows few cracks on the surface, which is justified by the infeasibility of the assumed light angle. For a human being, the light is obviously coming from an oblique angle, rather than being a vertical one. Notice how the surface becomes much smoother when the angle is set to be 26°, 63°, and 45°. The reconstruction is not perfect and this might be due to some deficiency in the implementation, but it’s still fine. Although the true light angle is unknown to us, a simple prediction proved the algorithm to be giving reasonable output. For panoramic views of the examples below, please refer to my website [6]. This again proves that the algorithm is not autonomous, and that it’s not expected to automatically predict the light angle just like humans do.
The following few examples in Figure 11 to Figure 15 illustrate the effectiveness of the algorithm when dealing with synthetic images, and the smoothness of the reconstructed surfaces for those images.

Figure 10 - The effect of light angle. Notice how all cases roughly give the same top view, despite the huge difference among the 3D surfaces.

Figure 11 - Vase Reconstruction

Figure 12 - Coin Reconstruction
Figure 13 – Bear Reconstruction

Figure 14 – Synthetic Face Reconstruction

Figure 15 - Pot Reconstruction
The last set of figures shows that the algorithm is doing well. Sometimes, the reconstruction doesn’t have a fine surface. This is because the image is of low resolution (less than 150x150). Still, playing with the figures in the web page [6] allows for better examination of the 3D reconstruction.

Notice that in Figure 14, the top view is pretty much like the original brightness image. However, the side view appears to be very disproportionate. This is a normal and well known issue in “Shape from Shading”. In fact, it was proven that different people have different sense of depth for the same brightness image.

The algorithm was also run on two faces, a synthetic one, and a real one. This is shown in Figure 16 and Figure 17, respectively. In both cases, generally, the output is pretty acceptable. Interestingly, we notice that the eyes are always interpreted as peaks or pits. Again, we also notice the disproportionate dimensions of the face when we take a side view, although the top view seems almost perfect.

As mentioned earlier, if the image consists of surfaces with materials of different $\rho$, the reconstruction would not make much sense. This is evident in Figure 17. The hair was manually removed because it intrinsically has a black color, and then would confuse the image because the surface is assumed to reflect light equally in all directions at all points.

VI. Conclusion

We have seen one of the approaches and implementations in the literature for “Shape from Shading”. The level sets method is proven to do well. However, it still suffers from problems. Some of these problems are specific to this approach, such as choosing the initial parameters, e.g. the time step size, thresholding the limits of the intensity spectrum, choosing the zero level contour...etc. Other problems are inherent to the problem, and might be solved by integrating “Shape from Shading” with other 3D reconstruction methodologies. An example for such problems is the +/- ambiguity.

Finally, the implementation was pretty straightforward, and the results were pretty decent and satisfactory for such an implementation that took short time to code, which means that better results can be obtained if the code is optimized and advanced enhancements are applied.
VII. References


