Abstract—Shape from Shading is a technique that enables to
compute the 3-D shape of objects from just one input image. It
makes use of information on illumination and surface reflection
properties. The modern model we use is given by a partial
differential equation. We show how recent algorithms solving
the latter give good results for real-world input images. Also, we
point out that the use of parallelisation techniques for multi-core
machines enable to achieve computational times of just a few
seconds, even for real-world input images of several mega-pixels.

I. INTRODUCTION

Shape from Shading (SfS) is a classic problem in computer
vision. It amounts to compute at hand of just a single given
grey value image the 3-D shape of depicted objects. To this
end, SfS relies on information about the scene illumination and
about the reflection properties of depicted objects.

SfS is usually seen as a very difficult, ill-posed problem.
About ten years ago, in an important survey paper summaris-
ing the attempts up to that point, the failure to achieve useful
results with SfS was acknowledged [9]: There, the authors state
that SfS results are bad for synthetic input images whereas all
parameters are known, and that they are even worse for
real-world scenes. However, we show that it is possible to
achieve good results for real-world images. The method we
use relies on a recent model [7] building upon the works of
Prados et al. [4] and Tankus et al. [5]. Moreover, we show that
the use of sophisticated numerical algorithms enables to have
these results in just a few seconds of computational times,
even for large input images of several mega-pixels, cf. [2],
[6].

In what follows, we first summarise our model. We proceed
with a description of our approaches to the 3-D reconstruction
by SfS for real-world images. By this we show some results.
At the end we give a conclusion.

II. THE MODEL

Let \( x \in \mathbb{R}^2 \) be in the image domain \( \Omega \). Furthermore:

- \( u := u(x) \) denotes the sought 3-D depth.
- \( I := I(x) \) is the normalised brightness of the given grey-
value image.
- \( f \) is the focal length. It denotes the distance between the
optical center of the camera and the 2-D plane to which the
scene of interest is mapped.

For the depth \( u > 0 \) holds as the depicted scene is in front
of the camera, and that the distance is measured in terms of
multiples of \( f \). We will use as unknown variable \( v := \ln(u) \).

The algorithms we use solve the model equation given by
the Hamilton-Jacobi partial differential equation (PDE)

\[
\frac{f^2 W}{Q} (I - k_a I_a) - k_a I_d e^{-2v} - W k_a I_d e^{-2v} R^a = 0 
\]

with \( W := \sqrt{f^2 \nabla v^2 + (\nabla v \cdot x)^2 + Q^2} \)

\( Q := f / \left( \sqrt{x^2 + y^2 + f^2} \right) \)

\( R := 2 Q^2 / W^2 - 1 \)

Thereby, \( \nabla v = (v_x, v_y)^\top \) is the so-called gradient of \( v \),
and the lower indices in \( v_x \) and \( v_y \) denote partial derivatives.
The surface is described here by the Phong-model [3], which is a
simple surface reflection model from computer graphics. The
underlying brightness relation reads as

\[
I = k_a I_a + \sum_{\text{light sources}} \frac{1}{f^2} (k_d I_d \cos \phi + k_s I_s (\cos \theta)^a) \]

where \( I_a, I_d, \) and \( I_s \) are the intensities of the ambient,
diffuse, and specular components of light, respectively. The
constants \( k_a, k_d, \) and \( k_s \) with \( k_a + k_d + k_s \leq 1 \) denote the
ratio of ambient, diffuse, and specular reflection. The ambient
light represents light present everywhere in a given scene.
The intensity of diffusely reflected light in each direction is
proportional to the cosine of the angle \( \phi \) between surface
normal and light source direction. The amount of specular light
reflected towards the viewer is proportional to \( (\cos \theta)^a \), where
\( \theta \) is the angle between the ideal mirror reflection direction of
the incoming light and the viewer direction, \( a \) is a constant
modelling the roughness of the material.

III. THE ALGORITHMS

There are two main roads to approach the solution of the
hyperbolic PDE (1): (i) iterative techniques can be employed,
and (ii) the Fast Marching (FM) method can be used. We now
outline some details of corresponding algorithms.

As a building block for the discretisation of spatial deriva-
tives, a stable upwind-type discretisation is used within all
methods. Let $h_x$ and $h_y$ be the pixel widths in $x$- and $y$-direction, respectively. Denoting then by $v_{i,j}$ the value of $v$ at the mesh point $(ih_x, jh_y)^T$, the upwind difference for the $x$-direction is

$$v_x(ih_x, jh_y) \approx \min \left( 0, \frac{v_{i+1,j} - v_{i,j}}{h_x}, \frac{v_{i-1,j} - v_{i,j}}{h_x} \right)$$

(6)

Accordingly, $v_y$ is discretised.

**Iterative Algorithms**

Using straightforward pointwise discretisations of all other terms in (1), easy-to-code iterative methods can be constructed solving the problem. In that context, also relatively sophisticated acceleration techniques can be applied, e.g. the fast sweeping method, or a multigrid scheme, cf. [1] for detailed descriptions.

When applying iterative techniques, one needs to engineer the following procedure in order to obtain high-quality results for SfS for real-world input images [8]:

1) Segmentation of the object of interest. This relatively sophisticated step can be realised e.g. by using active contour models.

2) Segmentation of textured parts on the object surface. This segmentation step can be achieved by relatively simple methods like e.g. an adaptive thresholding.

3) Inpainting of textured regions. This step can be realised conveniently by using diffusion models.

A result of such a proceeding is given in Figure 1.

**Fast Marching Techniques**

Assuming that it is possible to estimate the true depths at singular points – i.e. points on the surface of objects of interest where the 3-D depth is minimal – one can apply the FM idea. It relies on the causality principle incorporated in the underlying PDE: Beginning with the singular points, one builds up the solution surface by increasing step-by-step the depth. This results in an algorithm of complexity $O(N \log N)$, where $N$ is the number of pixels. Moreover, novel approaches allow to parallelise the method on multi-core machines, cf. [2], [6] for details on FM for our setting and for its parallelisation.

The FM method does not need an initial segmentation as the iterative techniques do, but one needs to have good estimates of the depth at singular points. The bargain for this is that the method is efficient enough to give results for real-world images in a few seconds, cf. Figure 2 and the comments below it.

![Fig. 2](image-url) FM for a real-world image. Left. Original image, 8 mega-pixels. Right. FM solution, computed with parallelisation in 35.76 seconds on a Intel Core 2 Quad Q8200, 2.33 GHz, with 2.2 MB L2 Cache and 4 GB RAM.

**IV. Conclusion**

We have shown that it is possible to obtain by recent models and modern algorithms good results for SfS in reasonable computing time on standard parallel hardware. Especially, we have parallelised the FM method which gives one of the fastest and most efficient algorithms in the field.

**Acknowledgment**

The second author would like to thank the Deutsche Forschungsgemeinschaft (DFG) for financing his research.

**References**


