

Algorithms for shape from shading, lighting direction and motion

Reinhard Klette and Volker Rodehorst

Fachgebiet Computer Vision, Institut für Technische Informatik im FB 20
Technische Universität Berlin, Franklinstr. 28/29, D-10587 Berlin

Abstract. A very interesting approach to the integration of several features into the process of analyzing shape may be based on [1, Theorem 2, Section 3.5] by *Aloimonos* and *Shulman*. In this study, the implementation of this approach was tested by comparing the results of three different algorithms. For applying this approach, several specific model assumptions and situations in scene space have to be carefully considered. A dependence could be proved of the accuracy of shape results from the direction to the point light source, and from the complexity of the realized motion in 3-D space.

1 Introduction

Recent work in Computer Vision is emphasizing the importance of integrative approaches. The approach considered in this paper is one example for that direction of research. On the other hand, it seems unrealistic that complex Computer Vision tasks may be solved in a kind of an "one-step integrative solution". On that way, several features or intrinsic characteristics have to be computed which will contribute to the solution at some next levels of the solution process. So, also some type of modularization should be assumed. For combining several (integrative) steps, in [3] the specification of defined approaches by derivational units was proposed. *Derivational units* consist of a certain combining theorem, a brief description of the application of the theorem by a rule of qualitative reasoning, and a proposal how this desired application may be implemented following a certain derivational algorithm. The notion *derivational algorithm* was used because certain specifications about shape will be derived by this algorithm from specific assumptions and features. By a certain combining theorem it is not yet defined how this result can be used in practice. This obvious situation should be taken in mind for the fair evaluation of a certain combining theorem.

In this short note, the difficulties on the way of precise presentations of derivational units are demonstrated by the discussion of one integrative approach to shape analysis. Experiences with three different algorithms and implementations will be reported. For world coordinates (X, Y, Z) , the *shape* is defined by the surface normals $\mathbf{n}(X, Y, Z) = (p, q, \pm 1)$ resp. the surface gradients (p, q) of visible surface points (X, Y, Z) of scene objects. The *visible surface* may be identified with a (unique) function $Z(X, Y)$. For the gradient resp. shape it holds that

$$p = \frac{\partial Z}{\partial X} \quad \text{and} \quad q = \frac{\partial Z}{\partial Y} .$$

At first, the used model for scene space and mapping of the scene space into the pictorial plane is briefly introduced. The pictorial xy -plane is parallel to the XY -plane,

and the Z -axis is identical with the optical axis of the camera. A single point light source is assumed, and $\mathbf{s} = (s_1, s_2, s_3)$ denotes the direction to this point light source (In practical applications, using e.g. neon tubes, this may be the direction to the closest point of the light source.). It is assumed that this directional vector \mathbf{s} is constant for the scene objects under consideration. Also, assume that the length of \mathbf{s} is normalized by $|\mathbf{s}| = 1$. In the case of parallel projection, the projection equations are given by

$$x = X \quad \text{and} \quad y = Y.$$

In the scene space, planar surface patches of opaque rigid bodies are considered. A surface patch may be assumed in the plane $Z = pX + qY + d$. It is assumed that the surface normals $\mathbf{n}(X, Y, Z)$ are always pointing toward the pictorial plane. In the experiments with moving bodies in the scene, the motion is described by a 3-D rotation with rotational speed $(\omega_1, \omega_2, \omega_3)$, and a 3-D translation (t_1, t_2, t_3) .

If parallel projection is assumed, then by *Kanatani* [2, Section 7.5] it was proved that the local displacement u, v between two consecutive images is given by

$$\begin{aligned} \dot{x} &= u(x, y) = t_1 + p\omega_2 x + (q\omega_2 - \omega_3)y \quad \text{and} \\ \dot{y} &= v(x, y) = t_2 + (-p\omega_1 + \omega_3)x - q\omega_1 y. \end{aligned}$$

Let u_x, u_y, v_x, v_y be the following *displacement differences* (this is not a gradient approximation) in case that the Z -axis is pointing to the pictorial plane (case **neg**),

$$\begin{aligned} u_x &= u(x+I, y) - u(x, y), & v_x &= v(x+I, y) - v(x, y), \\ u_y &= u(x, y+I) - u(x, y), & v_y &= v(x, y+I) - v(x, y). \end{aligned}$$

In contrast, in the case that the Z -axis is pointing into scene space (case **pos**), the *displacement differences* are defined by

$$\begin{aligned} u_x &= u(x+I, y) - u(x, y), & v_x &= v(x+I, y) - v(x, y), \\ u_y &= u(x, y-I) - u(x, y), & v_y &= v(x, y-I) - v(x, y), \end{aligned}$$

ensuring the "same orientation of the differences with respect to the Z -axis". By using the displacement differences for the case **pos**, from the both equations for $\dot{x} = u(x, y)$ and $\dot{y} = v(x, y)$ it follows that the shape parameters may be uniquely calculated by

$$p(x, y) = \frac{\omega_3 - v_x(x, y)}{\omega_1} = \frac{u_x(x, y)}{\omega_2} \quad \text{and} \quad q(x, y) = \frac{u_y(x, y) + \omega_3}{\omega_2} = \frac{-v_y(x, y)}{\omega_1}.$$

This result coincides (in principle) with the first two expressions of [1, Theorem 3.b), but in the second expression a negative sign was obtained. Thus, for known rotational speed of the surface patch, the gradient resp. shape may be computed if the local displacement was correctly calculated before. In the following we assume that the rotational speed of surfaces in the scene is unknown, and we also assume that there is some algorithm available for computing the local displacement values quite correctly.

2 Shape from shading, displacement, lighting direction

The approach to the integration of several features into the process of analyzing shape given by [1, Theorem 2, Section 3.5], is as follows. For two intensity images f_1 and f_2 from a series of images, the direction $\mathbf{s} = (s_1, s_2, s_3)$ to the point light source,

the local displacement $u(x,y)$ and $v(x,y)$, and from the ratio

$$r(x,y) = \frac{f_2(x + u(x,y), y + v(x,y))}{f_1(x,y)}, \text{ for } f_1(x,y) \neq 0,$$

of intensity values of "moving image points", a constraint for shape (p,q) was derived. As model assumptions, rigid bodies in the scene, parallel projection

$$x = X \text{ and } y = Y \text{ with } \mathbf{n}(X,Y,Z) = \mathbf{n}(x,y),$$

a point light source with constant direction, and Lambert reflection

$$f(x,y) = \rho \cdot \mathbf{n}(x,y) \cdot \mathbf{s}$$

with constant albedo ρ are used.

Unfortunately, this theorem is erroneously printed, e.g. even in the case of two identical intensity images f_1 and f_2 , i.e. no local displacement at all, from the printed theorem still a constraint could follow. But, the given proof is correct up to the non-published final computation of the algebraic constraint. And, it is worth to note that the assumed orientation of the Z -axis in the projection model has essential influence on the resulting algebraic expressions.

At first, assume that the Z -axis points to the pictorial plane, i.e. scene objects have negative Z -coordinates, as is the case in [1]. This model assumption was abbreviated by acronym **neg** above. Then, a correct formulation of the theorem is as follows.

For the displacement differences u_x, u_y, v_x, v_y in case **neg**, the following abbreviations will be used:

$$\begin{aligned} A &= (u_x + 1)(v_y + 1) - v_x u_y, & B &= u_y s_2 - s_1(v_y + 1), \\ C &= v_x s_1 - s_2(u_x + 1), & D &= u_y^2 + v_y(v_y + 2), \\ E &= v_x^2 + u_x(u_x + 2) \text{ and } & F &= 2u_y(u_x + 1) + 2v_x(v_y + 1). \end{aligned}$$

Theorem 1: In case **neg**, the following constraint holds for shape $p = p(x,y)$ and $q = q(x,y)$,

$$ap^2 + bq^2 + cpq + dp + eq + f = 0$$

with

$$\begin{aligned} a &= r^2 s_1^2 - B^2, & b &= r^2 s_2^2 - C^2, \\ c &= 2[r^2 s_1 s_2 - BC], & d &= 2r s_1 s_3(r - A), \\ e &= 2r s_2 s_3(r - A) & \text{and } & f = s_3^2 [A(A - 2r) + r^2] + C^2 D + BCF + B^2 E. \end{aligned}$$

In the proof of this constraint, the surface normals of visible surfaces, given by $\mathbf{n} = (p,q,1) = (1,0,-p) \times (0,1,-q)$, are considered. In the static case $f_1 = f_2$, it follows that $a = b = \dots = f = 0$ as expected.

In contrast, in the case that the Z -axis is pointing into scene space (**pos**), for the displacement differences the modified abbreviations

$$\begin{aligned} A &= (u_x + 1)(v_y - 1) - v_x u_y, & B &= u_y s_2 - s_1(v_y - 1), \\ C &= v_x s_1 - s_2(u_x + 1), & D &= u_y^2 + v_y(v_y - 2), \end{aligned}$$

$$E = v_x^2 + u_x(u_x + 2) \text{ and } F = 2u_y(u_x + 1) + 2v_x(v_y - 1)$$

will be used.

Theorem 2: In case **pos**, the constraint

$$ap^2 + bq^2 + cpq + dp + eq + f = 0$$

for shape $p = p(x,y)$ and $q = q(x,y)$ holds for

$$\begin{aligned} a &= r^2 s_1^2 - B^2, & b &= r^2 s_2^2 - C^2, \\ c &= 2[r^2 s_1 s_2 + BC], & d &= -2r s_1 s_3 (r + A), \\ e &= -2r s_2 s_3 (r + A) \quad \text{and} \quad f &= s_3^2 [A(A + 2r) + r^2] + C^2 D + BCF + B^2 E. \end{aligned}$$

In this case, in the proof the surface normals of visible surfaces are given by $\mathbf{n} = (p, q, -1) = (1, 0, p) \times (0, -1, -q)$.

These differences in the cases **neg** and **pos** should illustrate how sensitive constraints will react on modifications in the model.

This integrative algebraic relation by Aloimonos and Shulman [1] between several features may be read in different ways. For calculating the lighting direction, a scene object with known shape may be used. But, typically in a scene analysis approach, for analyzing shape this relation is defining a valuable constraint for shape. Some experimental results applying this approach are given already in [1].

3 Three Derivational Algorithms

By testing this approach on synthetic images, some detailed studies were possible. Because of the small values of displacement differences, roundings in the used algebraic expressions (as changes between **neg** and **pos**) have dramatic impact on the obtained practical results. The quality of the computable shape parameters depends upon the complexity of the 3-D motion, where complex motion leads to better results. If only translation is assumed, then the results are very poor (For the case of parallel projection, the simple additional influence of the translation on local displacement was cited above. Thus, this additive constant disappears in the difference functions.)

The second-order polynomial in p and q in Theorem 1 or 2 allows four different algebraic solutions (p, q) in general. Thus, by three different polynomials (constraints) a unique solution may be possible, and more than three constraints lead to some balance calculation.

Starting with a first constraint $a_1 p^2 + b_1 q^2 + c_1 p q + d_1 p + e_1 q + f_1 = 0$ for images f_0 and f_1 , then the two solutions

$$p = \frac{-qc_1 - d_1 \pm \sqrt{g}}{2a_1}$$

follow with abbreviations

$$\begin{aligned} g &= Gq^2 + Hq + J, & G &= c_1^2 - 4a_1 b_1, \\ H &= 2c_1 d_1 - 4a_1 e_1, \quad \text{and} & J &= d_1^2 - 4a_1 f_1. \end{aligned}$$

By insertion of these solutions of the first constraint into the second constraint $a_2 p^2 + b_2 q^2 + c_2 p q + d_2 p + e_2 q + f_2 = 0$ for images f_0 and f_2 , in the same point (x, y) and thus for the same values p and q , it follows that

$$h_1 (h_2 q^2 + h_3 \sqrt{g} q + h_4 \sqrt{g} + h_5 q + h_6) = 0$$

with coefficients

$$h_1 = \frac{1}{4a_1^2} ,$$

$$h_2 = a_2 (c_1^2 + G) + 4b_2 a_1^2 - 2c_1 c_2 a_1 ,$$

$$h_3 = 2 (c_2 a_1 - a_2 c_1) \quad \text{resp.} \quad h_3 = 2 (a_2 c_1 - c_2 a_1) ,$$

$$h_4 = 2 (d_2 a_1 - a_2 d_1) \quad \text{resp.} \quad h_4 = 2 (a_2 d_1 - d_2 a_1) ,$$

$$h_5 = a_2 (H + 2d_1 c_1) + 2a_1 (2e_2 a_1 - d_2 c_1 - c_2 d_1) ,$$

$$h_6 = a_2 (d_1^2 + J) - 2d_1 d_2 a_1 + 4f_2 a_1^2 .$$

By some algebraic operations, this simplifies to a fourth order polynomial

$$k_1 q^4 + k_2 q^3 + k_3 q^2 + k_4 q + k_5 = 0$$

with coefficients

$$k_1 = Gh_3^2 - h_2^2 ,$$

$$k_2 = 2Gh_3 h_4 + Hh_3^2 - 2h_2 h_5 ,$$

$$k_3 = Gh_4^2 + 2Hh_3 h_4 + Jh_3^2 - 2h_2 h_6 - h_5^2 ,$$

$$k_4 = Hh_4^2 + 2Jh_3 h_4 - 2h_5 h_6 ,$$

$$k_5 = Jh_4^2 - h_6^2 .$$

For the algorithmic solution of this fourth order polynomial, different methods may be applied leading to different *derivational algorithms* for deriving shape from shading, motion, and lighting direction.

Algorithm 1 (algebraic method): There are four (complex) solutions of this polynomial, and these solutions are equal to

$$q = \frac{-3k_2 N^{1/4} + \sqrt{3} (N^{3/4} + \sqrt{2} \sqrt{O-P})}{12k_1 N^{1/4}} ,$$

$$q = \frac{-3k_2 N^{1/4} + \sqrt{3} (N^{3/4} - \sqrt{2} \sqrt{O-P})}{12k_1 N^{1/4}} ,$$

$$q = \frac{-3k_2 N^{1/4} - \sqrt{3} (N^{3/4} - \sqrt{2} \sqrt{O+P})}{12k_1 N^{1/4}} , \quad \text{and}$$

$$q = \frac{-3k_2 N^{1/4} - \sqrt{3} (N^{3/4} + \sqrt{2} \sqrt{O+P})}{12k_1 N^{1/4}} .$$

using the abbreviations

$$\begin{aligned} K &= k_4^2 (4k_1 k_3^3 - k_2^2 k_3^2) + 4(k_2^3 k_4^3 + k_5 k_2^2 k_3^3) + k_1 k_5 (6k_2^2 k_4^2 - 16k_3^4) \\ &\quad - 18k_3 (k_2 k_4^3 k_1 + k_2^3 k_4 k_5) + 27(k_4^4 k_1^2 + k_5^2 k_2^4) + 80k_2 k_4 k_5 k_1 k_3^2 \\ &\quad + k_5^2 k_1^2 (128k_3^2 + 192k_2 k_4) - 144k_3 (k_5 k_1^2 k_4^2 + k_5^2 k_1 k_2^2) - 256k_3^3 k_1^3 , \end{aligned}$$

$$L = 2k_3^3 - 9k_3 (k_2 k_4 + 8k_5 k_1) + 27(k_4^2 k_1 + k_5 k_2^2) ,$$

$$\begin{aligned}
M &= k_1(4(L+3)\sqrt{3}\sqrt{K})^{1/3} + (4(L-3)\sqrt{3}\sqrt{K})^{1/3}, \\
N &= \sqrt{3k_2^2 - 8k_1k_3 + 2M}, \\
O &= \sqrt{N(3k_2^2 - 8k_3k_1 - M)}, \text{ and} \\
P &= 3\sqrt{3}(k_2^3 - 4k_2k_1k_3 + 8k_1^2k_4).
\end{aligned}$$

The computed four values of q are used to calculate eight related values of p . A unique final solution could be computed by implementing evidence rules.

Algorithm 2 (numeric method): The fourth order polynomial in q given above may be solved by a numeric iterative procedure following [4]. Here, at first a function ϕ for the iteration

$$q^{v+1} := \phi(q^v), \quad v = 0, 1, 2, \dots$$

has to be determined, and then in case of the convergence of the sequence $\{q^v\}$ the limit is considered as a solution to the given polynomial.

Algorithm 3 (LSE optimization): For a sequence of images f_0, f_1, f_2, \dots , image f_0 is taken as reference where for points (x,y) in image f_0 the gradients have to be computed. According to the given constraint in Theorem 1 resp. Theorem 2, for f_0 and $f_i, i \geq 1$, the difference of

$$\text{error}_i(p,q) = ap^2 + bq^2 + cpq + dp + eq + f$$

to zero evaluates a gradient (p,q) , and during processing of the image sequence, the arithmetic mean value of squares of these errors $\text{error}_i(p,q)$ has to be minimized (LSE - optimization).

These three algorithms were implemented, and several synthetic scenes were used as input. The synthetic object of a Lambertian sphere allows a good comparison for all directions of surface normals in visible surface points, but complex graphical objects gave a good visual impression about the different qualities of shape reconstruction. The position of the point light source, i.e. vector s , was assumed to be identical for each triplet of runs of the three different algorithms. This position of the light source has some influence to the results in that sense that surface normals with directions orthogonal to the direction of vector s are hard to compute exactly in general.

By comparing these three derivational algorithms, the algebraic method is computationally fast, very instable, and imaginary numbers appear and have to be approximated. By this algorithm, only normals $\mathbf{n} = (p, q, -1)$ may be recognized which form an angle less than 90° with the direction s to the point light source.

The numeric method is more stable with respect to the number of solutions (different picture points (x,y)) and their quality. If surface plane and picture plane are close to being nearly orthogonal, erroneous results may be avoided by a threshold. Imaginary terms during iteration are taken as they are, and not approximated. Typically, a very fast convergence was considered. An additional treatment of the results by a Newton iteration did not lead to essential improvements.

The LSE-optimization has given in all experiments the best results. By specifying the resolution during minimization, the quality of the results may be selected according to possible computing time limitations. Even in the case of a coarse resolution, say 10° , the computed gradients are quite accurate. Drawbacks of this algorithm are

large time complexity and missing solutions for gradient directions close to 90° .

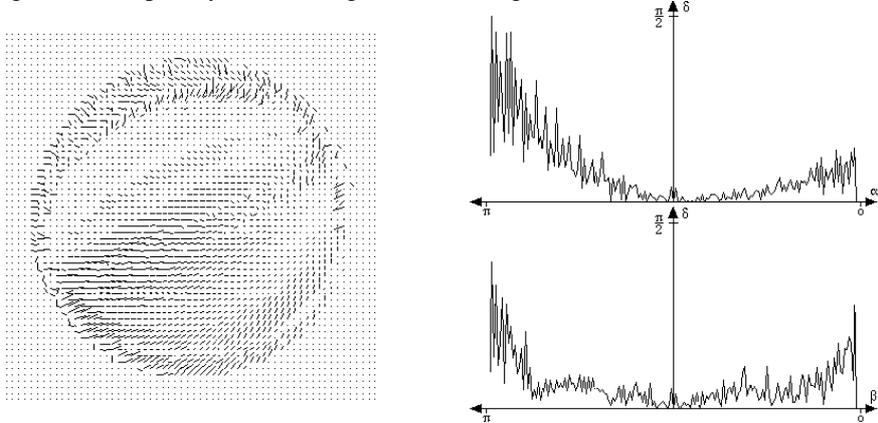


Fig. 1. Needle map of difference vectors (!) and differences between computed and actual normals in case of using the algebraic method.

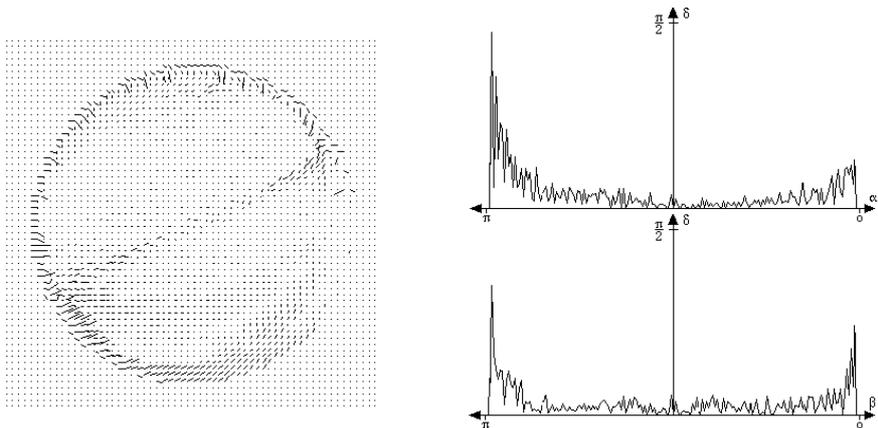


Fig. 2. Needle map of difference vectors (!) and differences between computed and actual normals in case of using the numeric method.

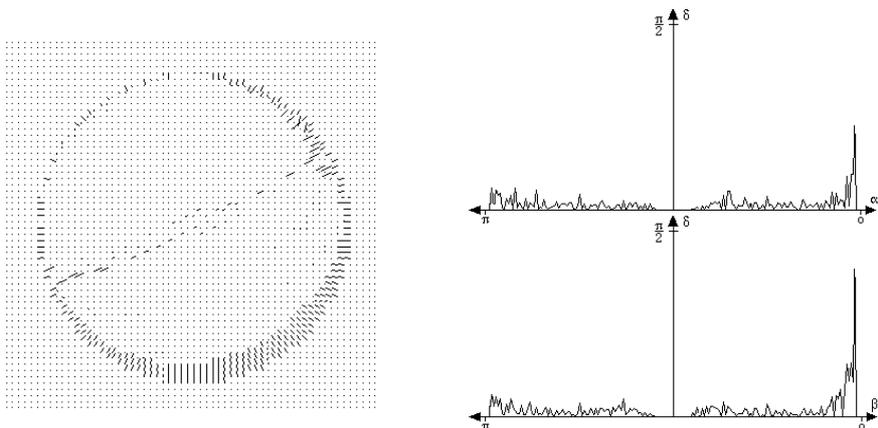


Fig. 3. Needle map of difference vectors (!) and differences between computed and actual normals in case of using the numeric method.

normals in case of using the LSE optimization.

For the illustration of the three algorithms, in Figs. 1, 2 and 3, the resulting needle maps of difference vectors (!) between surface normals and computed normals are shown in case of a Lambertian sphere. In these figures, the two diagrams have the following interpretation. In the upper diagram, the differences of computed normals to the ideal normals (average of errors) are illustrated for the p -direction, i.e. on the left the errors for normals in directions $-90^\circ \dots 0^\circ$, and on the right the errors for normals in directions $0^\circ \dots 90^\circ$ are shown.

The p -direction means that the left half of the diagram illustrates the left hemisphere, and the right half the right hemisphere. The lower diagram corresponds to the q -direction, i.e. on the left the lower hemisphere, and on the right the upper hemisphere is illustrated. The errors are shown as values between 0 and 90° . As shown in these diagrams, the errors increase in each case if surface plane and picture plane are close to being nearly orthogonal, and in Fig. 3, for the LSE-optimization, the errors are smaller than in Figs. 1 and 2, for the algebraic resp. numeric method.

4 Summary

For a known integrative approach to shape analysis several algorithmic solutions were realized and compared. Some detailed behaviour of the approach could be analyzed, as, for example:

The direction s to the point light source has essential influence on the computed normals. The critical situation is given if the direction s coincides with the optical axis, i.e. the Z -axis, because in this case $a = b = \dots = f = 0$ follows for the given constraint. In case of the sphere, see Figs. 1, 2, and 3, the normals on a diagonal which is orthogonal to direction s , are hard to recognize because coefficients d , e and f of the constraint are close to zero. Also, the $3-D$ motion of the objects has essential impact. There should be rotations with respect to all three coordinate axis. Normals orthogonal to the motion direction may be recognized very stable.

The realized algorithms are massively parallel. But, also in sequential mode they may run even on a PC for demonstrational purposes.

References

1. J. Aloimonos, D. Shulman: Integration of Visual Modules. Boston: Academic Press 1989.
2. K. Kanatani: Group-Theoretical Methods in Image Understanding. Berlin: Springer 1990.
3. R. Klette: 1992. Integrative Approaches to 'Shape from Motion'. in: R. Klette, W.G. Kropatsch (eds.): Theoretical foundations of computer vision. Berlin: Akademie Verlag 1992, pp. 203 - 214.
4. D.E. Muller: A Method for Solving Algebraic Equations Using an Automatic Computer. Math. Tables Aids Comp. 10, 208 - 215 (1956).