PSM, Part 2, and Gradient Field Integration[1]

Lecture 17

See Section 7.4 in
Reinhard Klette: Concise Computer Vision
Springer-Verlag, London, 2014

[1]See last slide for copyright information.

3PSM - The Solution

$\mathbf{n}_P$ is collinear with the cross product

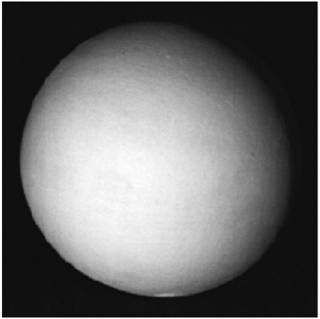
$$(E_1 u_2 ||\mathbf{s}_2||_2 \cdot \mathbf{s}_1 - E_2 u_1 ||\mathbf{s}_1||_2 \cdot \mathbf{s}_2) \times (E_1 u_3 ||\mathbf{s}_3||_2 \cdot \mathbf{s}_1 - E_3 u_1 ||\mathbf{s}_1||_2 \cdot \mathbf{s}_3)$$

Uniquely defines unit normal $\mathbf{n}_P^\circ$ pointing away from the camera

We only need relative intensities of light sources, no absolute measurements E_i

Agenda

Calibration of Light Source Direction



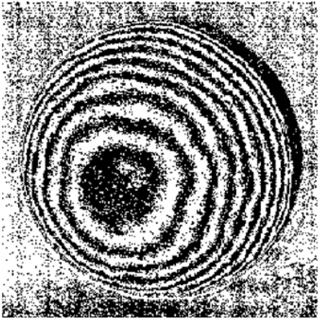


Image of a calibration sphere, ideally with uniform albedo and Lambertian reflectance

Illustration of detected isointensity "lines", showing the noisiness involved

Direction to Light Sources by Inverse PSM

Calibration of directions $\mathbf{s}_i$ to the three light sources

We apply *inverse photometric stereo*:

Use of a *calibration sphere* with (about) Lambertian reflectance and uniform albedo

Sphere about at location where object normals will be recovered.

1. Identify the circular border of the imaged sphere.
2. Calculate surface normals (of the sphere) at more than three points P (say, about 100) projected into pixel positions within the circle. How?
3. Identify direction $\mathbf{s}_i$ by least-square error optimization using the 3PSM solution equations
4. We have values u_i and normals $\mathbf{n}_P$; solve for unknown direction $\mathbf{s}_i$

Light Source Energy and Set-Up

Also measure this way (i.e. inverse PSM) energy ratios between intensities E_i of the three light sources.

Recommendation

Angle between two light source directions (centered around the viewing direction of the camera) should be about 56° for optimized PSM results

Agenda

1. Calibration of Light Sources
2. Albedo Recovery
3. Integration of Gradient Fields
4. Local Integration Methods
5. Global Optimization
6. Fourier-Transform Based Method

Albedo Recovery

For $i = 1, 2, 3$, consider equations

$$u_i = \frac{E_i}{c\pi} \cdot \rho(P) \cdot \frac{\mathbf{s}_i^\top \cdot \mathbf{n}_P}{||\mathbf{s_i}||_2 \cdot ||\mathbf{n}_P||_2}$$

We have three values u_i at p (projection of surface point P)

We have (approximate) values for unit vectors $\mathbf{s}_i^\circ$ and $\mathbf{n}_P^\circ$

We have relative intensities of the three light sources.

Only remaining unknown is $\rho(P)$

Combine first and second, first and third, and second and third image for a robust estimation of $\rho(P)$

How?

Why Albedo Recovery?

The knowledge of the albedo is of importance for light-source independent modeling of the surface of an object, defined by geometry and texture (albedo)

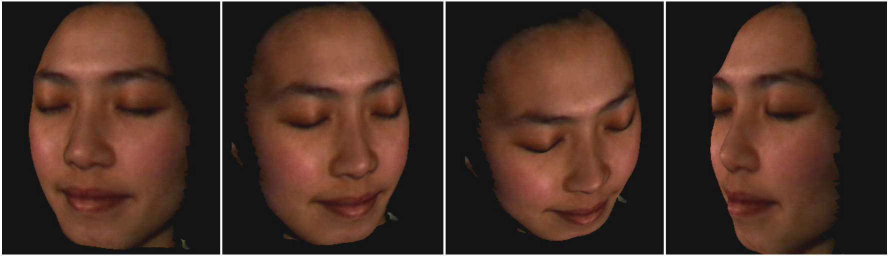
In general (if not limited to Lambertian reflectance), the albedo depends upon the wavelength of the illuminating light

As a first approximation, we may use light sources of different color, such as red, green, or blue light, to recover the related albedo values

Note that after knowing $\mathbf{s}^\circ$ and $\mathbf{n}^\circ$, we only have to change the wave length of illuminations (e.g. using transparent filters), assuming the object is not moving in between

Example: Human Faces

3PSM is of reasonable accuracy for recovering the albedo values of a human face



Face recovered by 3PSM (at University of Auckland in 2000)

Closed eyes avoid the recording of specularity

Agenda

Discrete Gradient Field

Discrete field of normals (or gradients) transformed into a surface by *integration*

Integration is not unique even when dealing with smooth surfaces

Result only determined up to an additive constant

Ill-Posedness of Discrete Integration

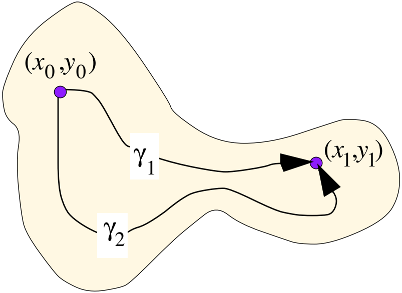
Results of PSM are discrete and erroneous surface gradient data

Surfaces often "non-smooth" (e.g. polyhedral)

Example: Camera looks onto a stair case, orthogonal to the front faces; recovered normals point straight towards camera

Densities of recovered surface normals do not correspond uniformly to local surface slopes

Integration Path



1. A surface patch defined on a simply-connected set
2. Its explicit surface function satisfies the integrability condition

Then: local integration along different paths will lead (in the continuous case) to identical elevation results at point (x_1, y_1), after starting at (x_0, y_0) with the same initial elevation value

Agenda

Theoretical Model for Ideal Inputs

Depth function $Z = Z(x, y)$ defined on simply-connected set Ω

At all $p \in \Omega$, Z satisfies the integrability condition

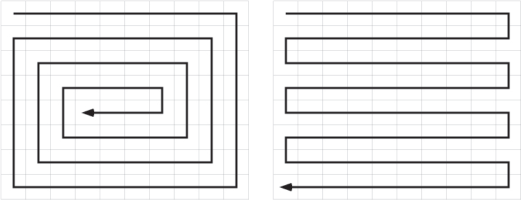
$$\frac{\partial Z^2}{\partial x \partial y} = \frac{\partial Z^2}{\partial y \partial x}$$

Then: Z can be recovered by starting at one point (x_0, y_0) with an initial value $Z(x_0, y_0)$, and then by integrating gradients along a path γ which is completely in the set

Different integration paths lead (theoretically) to an identical value at (x_1, y_1)

Local Integration Methods

Implement integration along selected paths (e.g. one or multiple scans through the image)



by using initial Z-values and local neighborhoods at a pixel location when updating Z-values incrementally

Example: Two-Scan Method

Task: Recover depth function Z such that

$$
\begin{aligned}
\frac{\partial Z}{\partial x}(x,y) &= a(x,y) \\
\frac{\partial Z}{\partial y}(x,y) &= b(x,y)
\end{aligned}
$$

for given gradient values $a_{x,y}$ and $b_{x,y}$ at pixel locations $(x,y) \in \Omega$

Local Increments

Line connecting $(x, y+1, Z_{x,y+1})$ and $(x+1, y+1, Z_{x+1,y+1})$ is approximately perpendicular to average normal between these two points

Thus: Dot product of slope of this line and average normal equal to zero

$$Z_{x+1,y+1} = Z_{x,y+1} + \frac{1}{2}\left(a_{x,y+1} + a_{x+1,y+1}\right)$$

Similarly:

$$Z_{x+1,y+1} = Z_{x+1,y} + \frac{1}{2}\left(b_{x+1,y} + b_{x+1,y+1}\right)$$

Adding both equations and dividing by 2

$$\begin{aligned} Z_{x+1,y+1} &= \frac{1}{2}\left(Z_{x,y+1} + Z_{x+1,y}\right) \\ &+ \frac{1}{4}\left(a_{x,y+1} + a_{x+1,y+1} + b_{x+1,y} + b_{x+1,y+1}\right) \end{aligned}$$

Two Stages of Algorithm

Total number of points on object surface is $N_{cols} \times N_{rows}$

Two arbitrary initial height values at $(1, 1)$ and at (N_{cols}, N_{rows})

Two-scan algorithm: first stage starts at $(1, 1)$, determines height values along x-axis and y-axis by discretizing weak integrability in terms of forward differences

$$\begin{aligned} Z_{x,1} &= Z_{x-1,1} + a_{x-1,1} \\ Z_{1,y} &= Z_{1,y-1} + b_{1,y-1} \end{aligned}$$

where $x = 2, ..., N_{cols}$ and $j = 2, ..., N_{rows}$, and scans image then vertically using the local increments defined on the last slide

Second Stage

Starts at corner (N_{cols}, N_{rows}), sets height values by

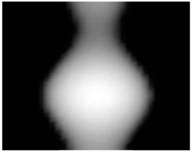
$$\begin{aligned} Z_{x-1,N_{rows}} &= Z_{x,N_{rows}} - a_{x,N_{rows}} \\ Z_{N_{cols},y-1} &= Z_{N_{cols},y} - b_{N_{cols},y} \end{aligned}$$

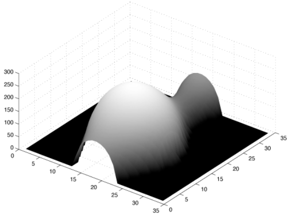
and scans the image horizontally using

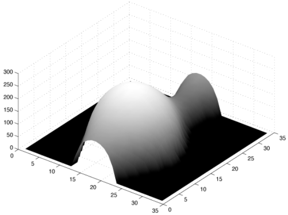
$$\begin{aligned} Z_{x-1,y-1} &= \frac{1}{2}(Z_{x-1,y} + Z_{x,y-1}) \\ &- \frac{1}{4}(a_{x-1,y} + a_{x,y} + b_{x,y-1} + b_{x,y}) \end{aligned}$$

Final step: Estimated height values are affected by the choice of the initial height values. Take average of results of both scans as final height value

Example for Two-scan Method







Original synthetic vase object

Ground truth: 3D plot of the vase object

Reconstruction result using the two-scan method

General: Local methods provide unreliable reconstructions for noisy gradient inputs since errors propagate along the scan paths

Generation of the Vase

Synthetic vase generated by using the following explicit surface equation

$$Z(x,y) = \sqrt{f^2(y) - x^2}$$

where

$$\begin{aligned} f(y) &= 0.15 - 0.1 \cdot y(6y+1)^2(y-1)^2(3y-2)^2 \\ &\text{for} \quad -0.5 \le x \le 0.5 \\ &\text{and} \quad 0.0 \le y \le 1.0 \end{aligned}$$

Agenda

1. Calibration of Light Sources
2. Albedo Recovery
3. Integration of Gradient Fields
4. Local Integration Methods
5. Global Optimization
6. Fourier-Transform Based Method

Global Integration

Gradient vector estimated at any $p \in \Omega$

Task: map this uniform and dense gradient vector field into a surface in 3D space which is likely to be the actual surface which caused the estimated gradient vector field

Depth values $Z(x, y)$ define labels at pixel locations (x, y)

Back to a labeling problem with error (or energy) minimization

Data term

$$E_{data}(Z) = \sum_{\Omega} [(Z_x - a)^2 + (Z_y - b)^2] + \lambda_0 \sum_{\Omega} [(Z_{xx} - a_x)^2 + (Z_{yy} - b_y)^2]$$

Smoothness term

$$E_{smooth}(Z) = \lambda_1 \sum_{\Omega} [Z_x^2 + Z_y^2] + \lambda_2 \sum_{\Omega} [Z_{xx}^2 + 2Z_{xy}^2 + Z_{yy}^2]$$

Notation

Z_x and Z_y

first-order partial derivatives of Z

a_x and b_y

first-order partial derivatives of a and b

Z_{xx}, $Z_{xy} = Z_{yx}$, and Z_{yy}

second-order partial derivatives of Z

$\lambda_0 \geq 0$ controls consistency between surface curvature and changes in available gradient data

$\lambda_1 \geq 0$ controls smoothness of surface

$\lambda_2 \geq 0$ controls smoothness of surface curvature

Total Energy and Two Algorithms

Determine surface Z (i.e. the labeling function) such that total error (or total energy)

$$E_{total}(Z) = E_{data}(Z) + E_{smooth}(Z)$$

is minimized

Two Algorithms

Frankot-Chellappa algorithm is for $\lambda_0 = \lambda_1 = \lambda_2 = 0$, thus not using the second part of the data constraint and no smoothness constraint at all

Wei-Klette algorithm also uses second-order derivatives (curvature) and smoothness optimization

Agenda

Optimization problem can be solved by using the theory of projections onto convex sets

Gradient field $(a_{x,y}, b_{x,y})$ is projected onto the nearest integrable gradient field in the least-square sense, using the Fourier transform for optimizing in the frequency domain

2D DFT of surface function $Z(x, y)$

$$\mathbf{Z}(u, v) = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} Z(x, y) \cdot \exp\left[-i2\pi \left(\frac{xu}{N_{cols}} + \frac{yv}{N_{rows}}\right)\right]$$

Inverse transform

$$Z(x, y) = \sum_{(u,v)\in\Omega} \mathbf{Z}(u, v) \cdot \exp\left[i2\pi \left(\frac{xu}{N_{cols}} + \frac{yv}{N_{rows}}\right)\right]$$

$i = \sqrt{-1}$ and u and v represent frequencies in 2D Fourier domain

More Fourier Pairs

$$
\begin{aligned}
Z_x(x,y) &\Leftrightarrow iu\ \mathbf{Z}(u,v) \\
Z_y(x,y) &\Leftrightarrow iv\ \mathbf{Z}(u,v) \\
Z_{xx}(x,y) &\Leftrightarrow -u^2\ \mathbf{Z}(u,v) \\
Z_{yy}(x,y) &\Leftrightarrow -v^2\ \mathbf{Z}(u,v) \\
Z_{xy}(x,y) &\Leftrightarrow -uv\ \mathbf{Z}(u,v)
\end{aligned}
$$

Define the appearance of considered derivatives of Z in frequency domain

$\mathbf{A}(u,v)$ and $\mathbf{B}(u,v)$ be Fourier transforms of gradients $A(x,y) = a_{x,y}$ and $B(x,y) = b_{x,y}$, respectively

Optimization in Frequency Domain

In conclusion to Parseval's Theorem: Equivalence of optimization problem in spatial domain to optimization problem in frequency domain

Minimize, where sums are for $(u, v) \in \Omega$:

$$\begin{aligned}
& \sum_{\Omega} \left[(iu\mathbf{Z} - \mathbf{A})^2 + (iv\mathbf{Z} - \mathbf{B})^2 \right] \\
+ \quad & \lambda_0 \sum_{\Omega} \left[\left(-u^2\mathbf{Z} - iu\mathbf{A}\right)^2 + \left(-v^2\mathbf{Z} - iv\mathbf{B}\right)^2 \right] \\
+ \quad & \lambda_1 \sum_{\Omega} \left[(iu\mathbf{Z})^2 + (iv\mathbf{Z})^2 \right] \\
+ \quad & \lambda_2 \sum_{\Omega} \left[\left(-u^2\mathbf{Z}\right)^2 + 2\,(-uv\mathbf{Z})^2 + \left(-v^2 Z_F\right)^2 \right]
\end{aligned}$$

Start of Solution Process

Above expression expanded into

$$\sum_{\Omega} \left[u^2\mathbf{ZZ}^\star - iu\mathbf{ZA}^\star + iu\mathbf{Z}^\star\mathbf{A} + \mathbf{AA}^\star \right.$$
$$\left. + v^2\mathbf{ZZ}^\star - iv\mathbf{ZB}^\star + iv\mathbf{Z}^\star\mathbf{B} + \mathbf{BB}^\star \right]$$
$$+\lambda_0 \sum_{\Omega} \left[u^4\mathbf{ZZ}^\star - iu^3\mathbf{ZA}^\star + iu^3\mathbf{Z}^\star\mathbf{A} + u^2\mathbf{AA}^\star \right.$$
$$\left. + v^4\mathbf{ZZ}^\star - iv^3\mathbf{ZB}^\star + iv^3\mathbf{Z}^\star\mathbf{B} + v^2\mathbf{BB}^\star \right]$$
$$+\lambda_1 \sum_{\Omega} \left(u^2 + v^2 \right) \mathbf{ZZ}^\star$$
$$+\lambda_2 \sum_{\Omega} \left(u^4 + 2u^2v^2 + v^4 \right) \mathbf{ZZ}^\star$$

$\star$ denotes the complex conjugate, and sums for $(u, v) \in \Omega$

Optimization in Frequency Space

Differentiating the above expression with respect to $\mathbf{Z}^\star$ and setting the result to zero, we can deduce the necessary condition for a minimum of the cost function

For each $(u, v) \in \Omega$ we have

$$\left(u^2\mathbf{Z} + iu\mathbf{A} + v^2\mathbf{Z} + iv\mathbf{B}\right) + \lambda_0\left(u^4\mathbf{Z} + iu^3\mathbf{A} + v^4\mathbf{Z} + iv^3\mathbf{B}\right) \\ +\lambda_1\left(u^2 + v^2\right)\mathbf{Z} + \lambda_2\left(u^4 + 2u^2v^2 + v^4\right)\mathbf{Z} = 0$$

A rearrangement of this equation yields

$$\left[\lambda_0\left(u^4 + v^4\right) + (1 + \lambda_1)\left(u^2 + v^2\right) + \lambda_2\left(u^2 + v^2\right)^2\right]\mathbf{Z}(u, v) \\ +i\left(u + \lambda_0 u^3\right)\mathbf{A}(u, v) + i\left(v + \lambda_0 v^3\right)\mathbf{B}(u, v) = 0$$

Solution

Solve the above equation except for $(u, v) \neq (0, 0)$:

$$\mathbf{Z}(u, v) = \frac{-i\left(u + \lambda_0 u^3\right)\mathbf{A}(u, v) - i\left(v + \lambda_0 v^3\right)\mathbf{B}(u, v)}{\lambda_0\left(u^4 + v^4\right) + \left(1 + \lambda_1\right)\left(u^2 + v^2\right) + \lambda_2\left(u^2 + v^2\right)^2}$$

Result

This is the Fourier transform of the unknown surface function $Z(x, y)$ expressed as a function of the Fourier transforms of the given gradients $A(x, y) = a_{x,y}$ and $B(x, y) = b_{x,y}$

Algorithm Part 1: Forward Transform

input gradients $a(x, y), b(x, y)$; parameters λ_0, λ_1, and λ_2
for $(x, y) \in \Omega$ **do**
    if $(|a(x, y)| < c_{\max}$ & $|b(x, y)| < c_{\max})$ **then**
        A1(x,y)=a(x,y);   A2(x,y)=0;
        B1(x,y)=b(x,y);   B2(x,y)=0;
    else
        A1(x,y)=0;   A2(x,y)=0;
        B1(x,y)=0;   B2(x,y)=0;
    end if
end for
Calculate Fourier transform in place: A1(u,v), A2(u,v);
Calculate Fourier transform in place: B1(u,v), B2(u,v);

Algorithm Part 2: Optimize in Frequency Domain

for $(u, v) \in \Omega$ **do**
  if $(u \neq 0 \ \& \ v \neq 0)$ **then**
    $\Delta = \lambda_0 \left(u^4 + v^4\right) + (1 + \lambda_1)\left(u^2 + v^2\right) + \lambda_2 \left(u^2 + v^2\right)^2$;
    $H1(u, v) = [(u + \lambda_0 u^3) A2(u, v) + (v + \lambda_0 v^3) B2(u, v)] / \Delta$;
    $H2(u, v) = [-(u + \lambda_0 u^3) A1(u, v) - (v + \lambda_0 v^3) B1(u, v)] / \Delta$;
  else
    $H1(0, 0) =$ average depth; $H2(0, 0) = 0$;
  end if
end for

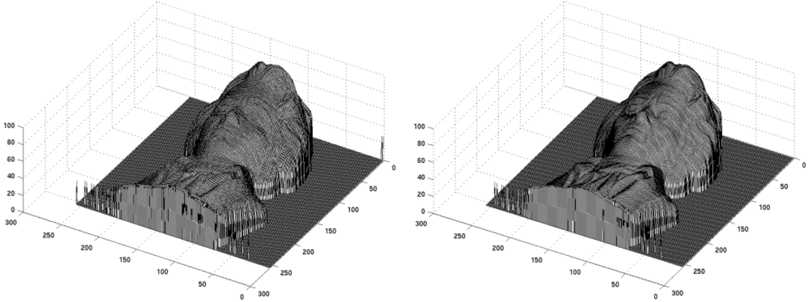
Algorithm Part 3: Backward Transform

1: Calculate inverse Fourier transform of H1(u,v) and H2(u,v) in place: H1(x,y), H2(x,y);
2: **for** $(x, y) \in \Omega$ **do**
3: $Z(x, y) = H1(x, y)$;
4: **end for**

Example 1

Image triplet of a Beethoven statue used as input for 3PSM

Example



Left: Recovered surface using the Frankot-Chellappa algorithm

Right: Recovered surface using the Wei-Klette algorithm with $\lambda_0 = 0.5$ and $\lambda_1 = \lambda_2 = 0$

Comments

Constant c_{max} eliminates gradient estimates which define angles with the image plane close to $\pi/2$

A value such as $c_{max} = 12$ is an option

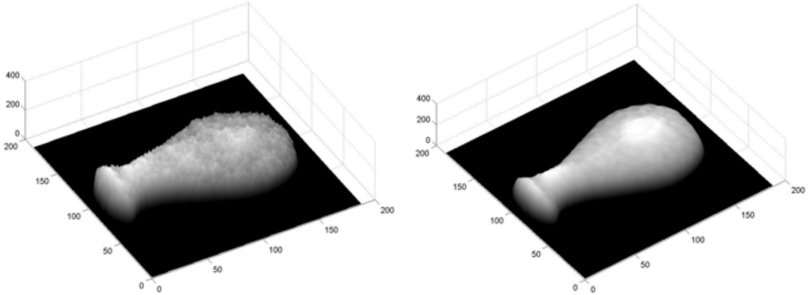
Real parts are stored in arrays A1, B1, and H1, and imaginary parts in arrays A2, B2, and H2

Average height can be estimated for the visible scene

Parameters λ_0, λ_1 and λ_2 should be chosen based on experimental evidence for the given scene

Test on Noisy Gradients

Generate discrete gradient vector field for synthetic vase and add Gaussian noise (with a mean, set to zero, and a standard deviation, set to 0.01)



Left: Frankot-Chellappa. *Right*: Wei-Klette with $\lambda_0 = 0$, $\lambda_1 = 0.1$, and $\lambda_2 = 1$

Copyright Information