Vision and Image Processing: Shading, Photometric Stereo

François Lauze

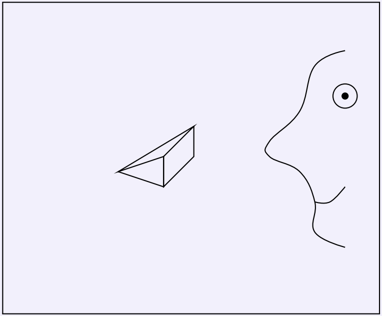
Department of Computer Science
University of Copenhagen

Plan for today and December 7th, 2020

- Image Formation and reflectance.
- Lighting Models.
- The Photometric Stereo Problem.

Outline

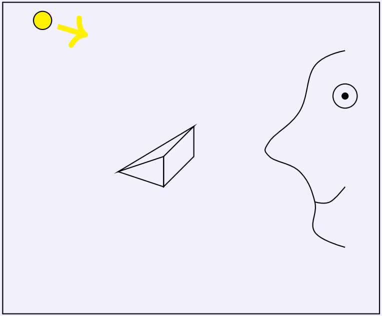
The Photometric Stereo (PS) Problem [Woodham, 1980]



- 1 fixed camera + 1 “fixed” scene
- m lightings

Slide by Y. Quéau

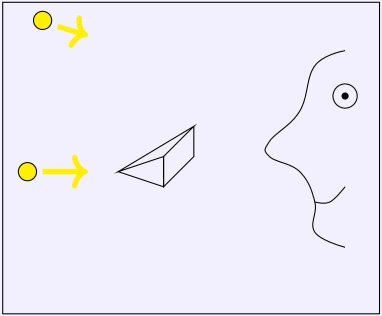
The Photometric Stereo (PS) Problem [Woodham, 1980]



- 1 fixed camera + 1 “fixed” scene
- *m* lightings

Slide by Y. Quéau

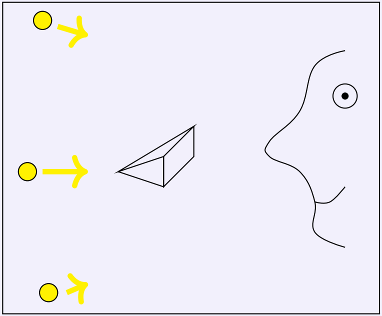
The Photometric Stereo (PS) Problem [Woodham, 1980]



- 1 fixed camera + 1 “fixed” scene
- m lightings

Slide by Y. Quéau

The Photometric Stereo (PS) Problem [Woodham, 1980]

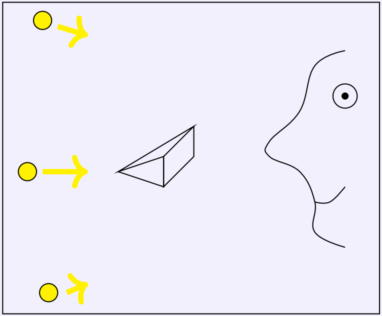


- 1 fixed camera + 1 “fixed” scene
- m lightings



Slide by Y. Quéau

The Photometric Stereo (PS) Problem [Woodham, 1980]



- 1 fixed camera + 1 “fixed” scene
- *m* lightings

Goal:

3D-reconstruction of the scene from the 2D images

Slide by Y. Quéau

Ingredients for Photometric Stereo

Photo metric Stereo

Ingredients for Photometric Stereo

Photometric Stereo

- φῶς/φωτός : (phõs, gen. photós): light

Ingredients for Photometric Stereo

Photometric Stereo

- φῶς/φωτός : (phõs, gen. photós): light
- μέτρον (métron): measure

Ingredients for Photometric Stereo

Photo metric Stereo

- φῶς/φωτός : (phõs, gen. photós): light
- μέτρον (métron): measure
- στερεός (stereós): solid/volume

Ingredients for Photometric Stereo

Photometric Stereo

- φῶς/φωτός : (phõs, gen. photós): light
- μέτρον (métron): measure
- στερεός (stereós): solid/volume

Suggest. Volume recovery from measured light. Needed:

Ingredients for Photometric Stereo

Photometric Stereo

- φῶς/φωτός : (phõs, gen. photós): light
- μέτρον (métron): measure
- στερεός (stereós): solid/volume

Suggest. Volume recovery from measured light. Needed:

- Understand reflectance: how is light reflected from an object.
- How can we measure it.
- How object geometry is linked to light.

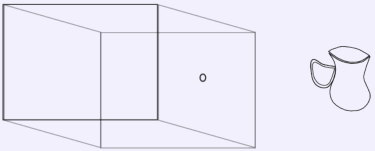
Outline

Light and Image Formation

Ingredients

- Object

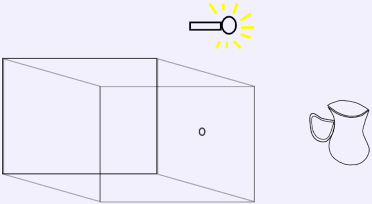
Light and Image Formation



Ingredients

- Object
- Camera

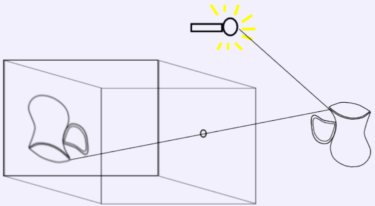
Light and Image Formation



Ingredients

- Object
- Camera
- Light source

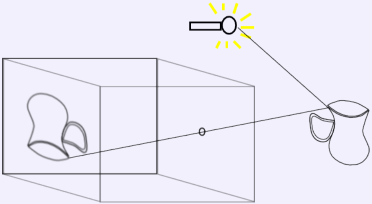
Light and Image Formation



Ingredients

- Object
- Camera
- Light source
- Light reflection by object surface.

Light and Image Formation

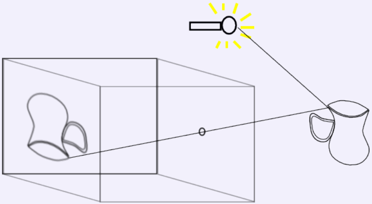


Ingredients

- Object
- Camera
- Light source
- Light reflection by object surface.

- Image formation inside camera: when light, scene and camera parameters known: reflectance function.

Light and Image Formation

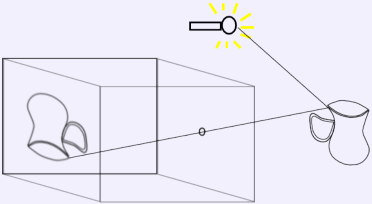


Ingredients

- Object
- Camera
- Light source
- Light reflection by object surface.

- Image formation inside camera: when light, scene and camera parameters known: reflectance function.
- BTW: Camera detectors react almost truly linearly to received luminance.

Light and Image Formation

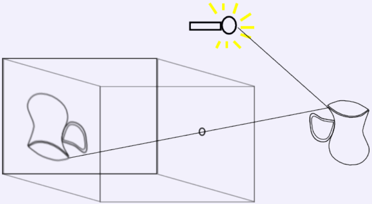


Ingredients

- Object
- Camera
- Light source
- Light reflection by object surface.

- Image formation inside camera: when light, scene and camera parameters known: reflectance function.
- BTW: Camera detectors react almost truly linearly to received luminance.

Light and Image Formation



Ingredients

- Object
- Camera
- Light source
- Light reflection by object surface.

- Image formation inside camera: when light, scene and camera parameters known: reflectance function.
- BTW: Camera detectors react almost truly linearly to received luminance.
- Can image formation model give enough information about the object surface to reconstruct it?

Materials and Light

Refraction and caustics due to moving water surface.

Refraction and color absorption due to amber.

Materials and Light

Refraction and caustics due to moving water surface.

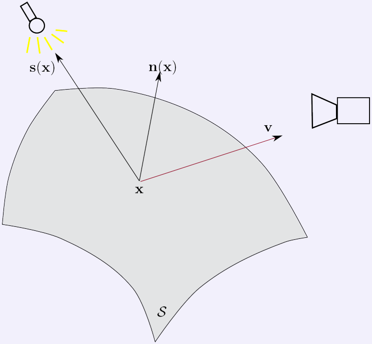
Refraction and color absorption due to amber.

quasi-monochromatic and mat!

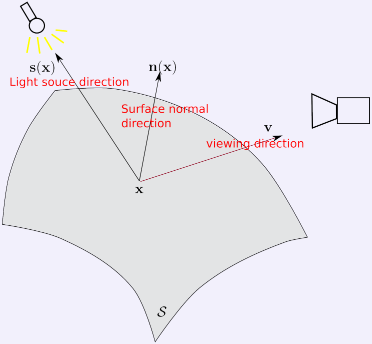
From now, we discuss only opaque scenes/objects (but come to me if you are interested in transparency and refraction).

Matte vs Brilliant

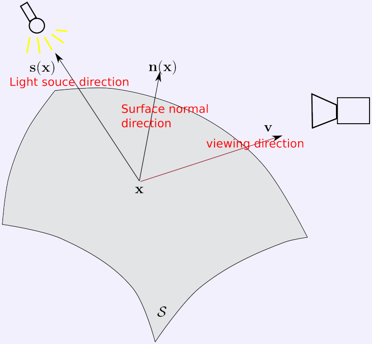
Bidirectional Reflectance Distribution Function – BRDF

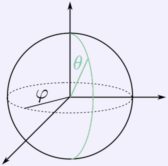


Bidirectional Reflectance Distribution Function – BRDF



Bidirectional Reflectance Distribution Function – BRDF

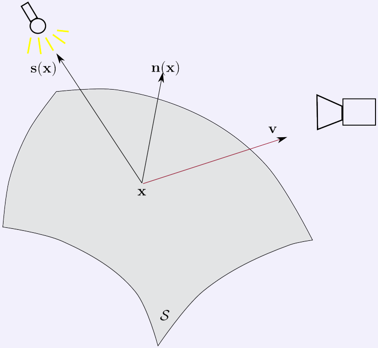




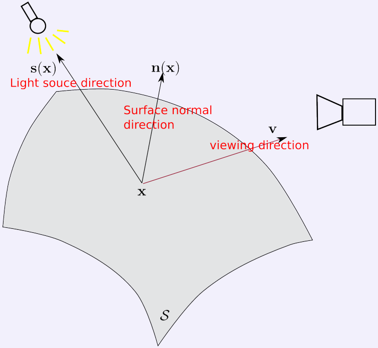
- $E^s(\mathbf{x}, \theta_i, \varphi_i)$: *Irradiance* at surface (at pos $\mathbf{x}$) in direction θ_i, φ_i
- $L^s(\mathbf{x}, \theta_e, \varphi_e)$: *Radiance* at surface (at pos $\mathbf{x}$) in direction θ_e, φ_e

$$\kappa(\mathbf{x}, \theta_i, \varphi_i, \theta_e, \varphi_e) = \frac{L^s(\mathbf{x}, \theta_e, \varphi_e)}{E^s(\mathbf{x}, \theta_i, \varphi_i)}$$

Bidirectional Reflectance Distribution Function – BRDF



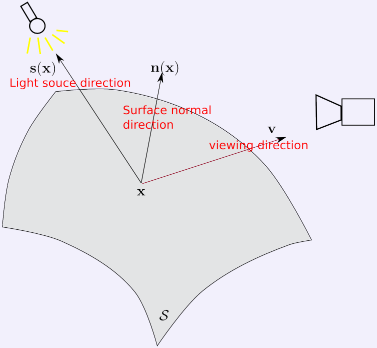
Bidirectional Reflectance Distribution Function – BRDF



- Luminance emitted by punctual object $\mathbf{x}$ on a surface $\mathcal{S}$ with normal direction $\mathbf{n}(\mathbf{x})$ at $\mathbf{x}$, in emission direction $\mathbf{v}$ characterized by spherical angles (θ_e, φ_e) w.r.t $\mathbf{n}(\mathbf{x})$:

$$L(\mathbf{x}, \theta_e, \varphi_e) = \int_{\theta_i=0}^{\frac{\pi}{2}} \int_{\varphi_i=0}^{2\pi} \kappa(\mathbf{x}, \theta_i, \varphi_i, \theta_e, \varphi_e) \bar{L}(\theta_i, \varphi_i) \sin\theta_i \, d\theta_i d\varphi.$$

Bidirectional Reflectance Distribution Function – BRDF

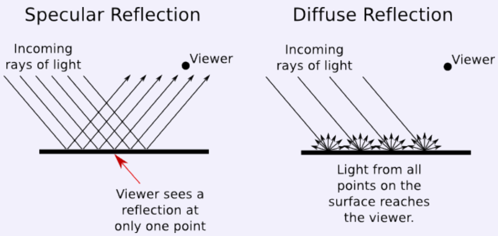


- Luminance emitted by punctual object $\mathbf{x}$ on a surface $\mathcal{S}$ with normal direction $\mathbf{n}(\mathbf{x})$ at $\mathbf{x}$, in emission direction $\mathbf{v}$ characterized by spherical angles (θ_e, φ_e) w.r.t $\mathbf{n}(\mathbf{x})$:

$$L(\mathbf{x}, \theta_e, \varphi_e) = \int_{\theta_i=0}^{\frac{\pi}{2}} \int_{\varphi_i=0}^{2\pi} \kappa(\mathbf{x}, \theta_i, \varphi_i, \theta_e, \varphi_e) \bar{L}(\theta_i, \varphi_i) \sin\theta_i \, d\theta_i d\varphi.$$

- Complicated equation, used in Computer Graphics. In Vision, try to guess a workable form for the reflection.

Specular Vs. Matte Objects



The two most standard reflection models: specular: mirror like surface, diffuse: rough surface (at very small scale): Lambertian model. Others, very polular: Phong, Gouraud, Torrance-Sparrow etc. . . especially useful in Computer Graphics.

Diffuse Reflection

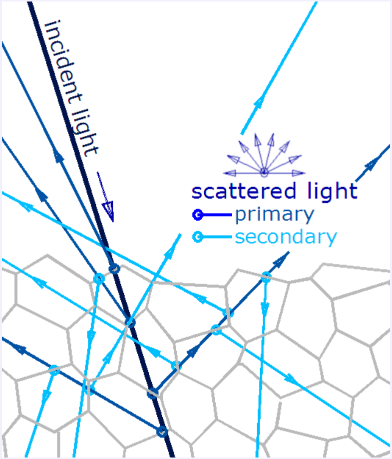


Image Source GianniG46, Wikipedia

Diffuse Reflection

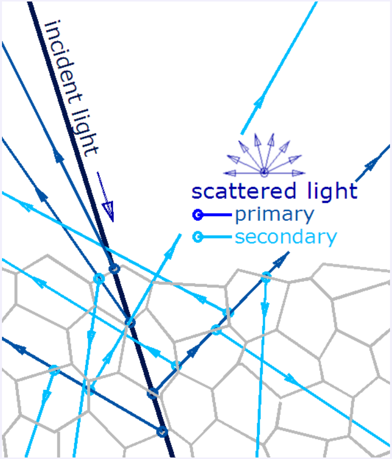


Image Source GianniG46, Wikipedia

Rough surface at micro-scale.

- Light bounces.
- Reflections in all directions
- Some light is absorbed.
- Only a percentage of light energy is reemitted.

Lambert's Cosine Law

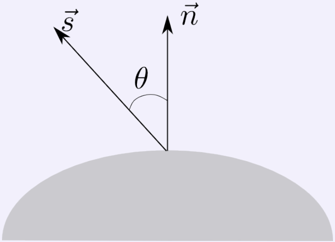
Reflectance

- Linearised Lambertian model: $I(\mathbf{p}) = \rho(\mathbf{x})\mathbf{s}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})$
- $\rho(\mathbf{x})$ is the *albedo* at $\mathbf{x}$ – material light absorption property, $\rho \in [0, 1]$. Assumes matte material such as chalk...

Lambert's Cosine Law

Reflectance

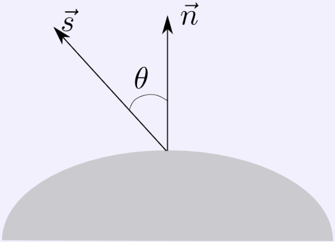
- Linearised Lambertian model: $I(\mathbf{p}) = \rho(\mathbf{x})\mathbf{s}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})$
- $\rho(\mathbf{x})$ is the *albedo* at $\mathbf{x}$ – material light absorption property, $\rho \in [0, 1]$. Assumes matte material such as chalk...

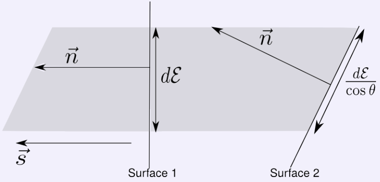


Lambert's Cosine Law

Reflectance

- Linearised Lambertian model: $I(\mathbf{p}) = \rho(\mathbf{x})\mathbf{s}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})$
- $\rho(\mathbf{x})$ is the *albedo* at $\mathbf{x}$ – material light absorption property, $\rho \in [0, 1]$. Assumes matte material such as chalk...

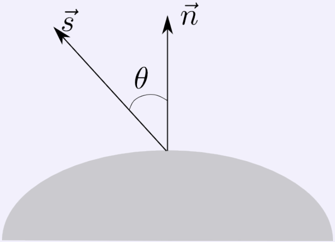


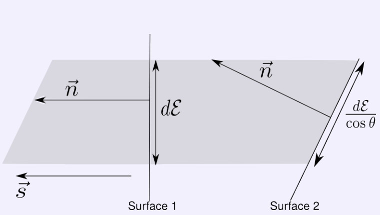


Lambert's Cosine Law

Reflectance

- Linearised Lambertian model: $I(\mathbf{p}) = \rho(\mathbf{x})\mathbf{s}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})$
- $\rho(\mathbf{x})$ is the *albedo* at $\mathbf{x}$ – material light absorption property, $\rho \in [0, 1]$. Assumes matte material such as chalk...

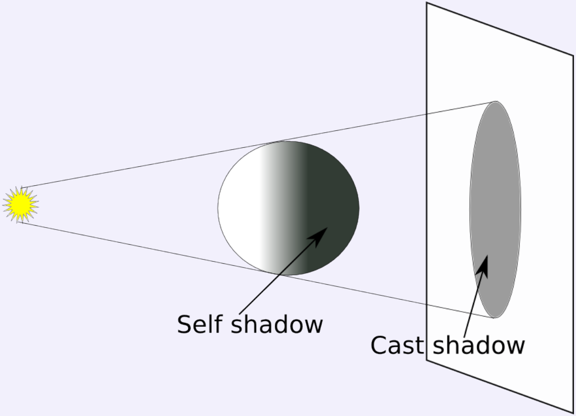




In words

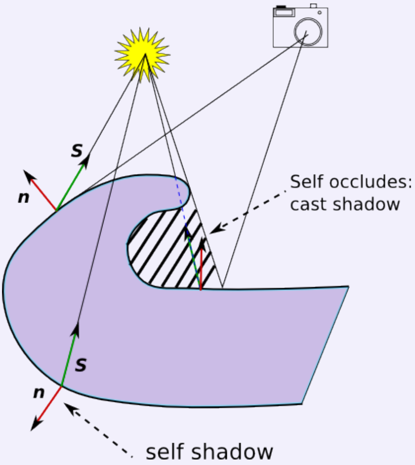
- Local orientation of object w.r.t. light: In surface 1: surface area matches ray "section". In surface 2: surface area larger than ray section, but receive same amount of light.

Shadows



- Self shadow: surface is behind the light source. $\mathbf{s} \cdot \mathbf{n} \leq 0$.
- Cast shadow: part of the scene occludes another part.

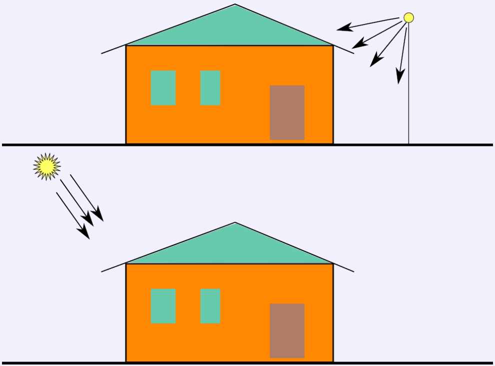
Shadows again



- Lambert's law and self-shadows: $I = \rho \max(\mathbf{s} \cdot \mathbf{n}, 0)$.
- Cast shadows: Non local phenomenon, Lambert's law is local...

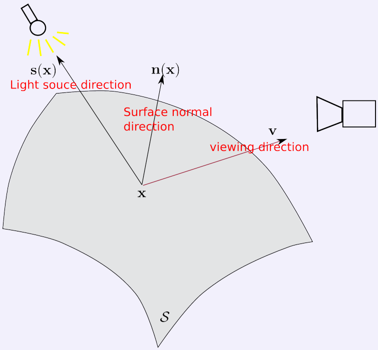
Outline

Type of Light Sources



- Top: near light source, radial.
- Bottom: far light source: parallel. Our choice in these lectures.
- Other types?

Shape From Shading (SfS) – B. Horn 1970



- Use Lambert's Law to gain information on visible surface via normal vector $\mathbf{n}(\mathbf{x})$.
- Need link between surface equations and normal vector.

Settings

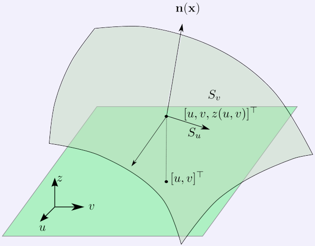
- Representation of the surface. Assume surface parameterized by $(u, v) \mapsto \mathcal{S}(u, v) \in \mathbb{R}^3$. Even better: depth map:

$$S(u, v) = [u, v, z(u, v)]^T : \text{ Monge Patch.}$$

Settings

- Representation of the surface. Assume surface parameterized by $(u, v) \mapsto \mathcal{S}(u, v) \in \mathbb{R}^3$. Even better: depth map:

$$S(u, v) = [u, v, z(u, v)]^T : \text{ Monge Patch.}$$



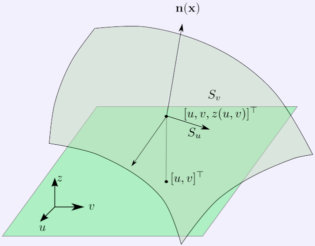
Tangent vectors at $\mathbf{x}(u, v) = [u, v, z(u, v)]^\top$

$$\frac{\partial \mathcal{S}}{\partial u} = S_u = \begin{bmatrix} 1 \\ 0 \\ \frac{\partial z}{\partial u} \end{bmatrix}, \quad \frac{\partial \mathcal{S}}{\partial v} = \mathcal{S}_v \begin{bmatrix} 0 \\ 1 \\ \frac{\partial z}{\partial v} \end{bmatrix},$$

Settings

- Representation of the surface. Assume surface parameterized by $(u,v) \mapsto \mathcal{S}(u,v) \in \mathbb{R}^3$. Even better: depth map:

$$S(u,v) = [u, v, z(u,v)]^T : \text{ Monge Patch.}$$



Tangent vectors at $\mathbf{x}(u,v) = [u, v, z(u,v)]^\top$

$$\frac{\partial \mathcal{S}}{\partial u} = S_u = \begin{bmatrix} 1 \\ 0 \\ \frac{\partial z}{\partial u} \end{bmatrix}, \quad \frac{\partial \mathcal{S}}{\partial v} = \mathcal{S}_v \begin{bmatrix} 0 \\ 1 \\ \frac{\partial z}{\partial v} \end{bmatrix},$$

- Normal vector $\mathbf{n}(u,v) = \mathbf{n}(\mathbf{x}(u,v))$

$$\mathbf{n}(\mathbf{x}) = \frac{\mathcal{S}_u \times \mathcal{S}_v}{|\mathcal{S}_u \times \mathcal{S}_v|} = \frac{1}{\sqrt{|\nabla z|^2 + 1}} \begin{bmatrix} -z_u \\ -z_v \\ 1 \end{bmatrix}$$

Camera Model, Lambertian SfS equations

- Two Choices: pinhole and orthographic.
- Orthographic Camera Model.
  - $[x = u, y = v, z] \mapsto [u, v]$: orthographic projection.
  - Formula from previous slide: Coordinates of normal vector in orthographic projection:

$$\mathbf{n}_z = \frac{1}{\sqrt{|\nabla z|^2 + 1}}[-z_u, -z_v, 1]^T$$

- Pinhole Camera model.
  - Projection

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \mapsto \begin{bmatrix} u = \frac{fx}{z} \\ v = \frac{fy}{z} \end{bmatrix}$$

  - Normal vector in camera coordinates: complicated!

$$\mathbf{n}_z = \frac{1}{\sqrt{|\nabla z|^2 + \left(\frac{z+[u,v]\nabla z}{f}\right)^2}} \left[-z_u, -z_v, \frac{z + [u, v]\nabla z}{f}\right]^T$$

Camera Model, Lambertian SfS equations

- Two Choices: pinhole and orthographic.
- Orthographic Camera Model.
  - $[x = u, y = v, z] \mapsto [u, v]$: orthographic projection.
  - Formula from previous slide: Coordinates of normal vector in orthographic projection:

$$\mathbf{n}_z = \frac{1}{\sqrt{|\nabla z|^2 + 1}}[-z_u, -z_v, 1]^T$$

- Pinhole Camera model.
  - Projection

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \mapsto \begin{bmatrix} u = \frac{fx}{z} \\ v = \frac{fy}{z} \end{bmatrix}$$

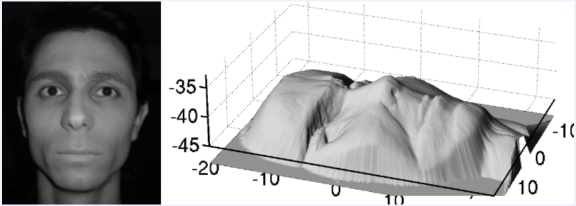
  - Normal vector in camera coordinates: complicated!

$$\mathbf{n}_z = \frac{1}{\sqrt{|\nabla z|^2 + \left(\frac{z+[u,v]\nabla z}{f}\right)^2}} \left[-z_u, -z_v, \frac{z + [u, v]\nabla z}{f}\right]^T$$

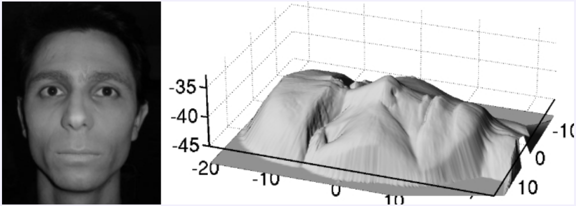
Solve for z: $I = \rho \mathbf{n}_z \cdot \mathbf{s}$

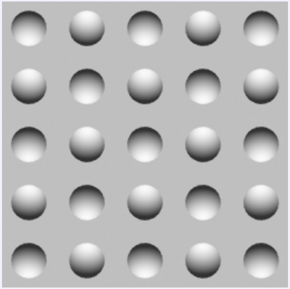
Complicated Math and Research Topics

Examples, Problems



Examples, Problems





SfS (Counter)Example

SfS (Counter)Example

SfS (Counter)Example

Why Does It Go Wrong

Assume Light **s** known and constant (far light source with known intensity).Per Pixel:

- Number of unknowns:
  - Normal vector $\mathbf{n}(u, v)$, 3 components $[\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3]$, but

    $$\|\mathbf{n}\| = 1 : \mathbf{n}_1^2 + \mathbf{n}_2^2 + \mathbf{n}_3^2 = 1.$$

    2 degrees of freedom (DoF).
  - Albedo $\rho(u, v)$: 1 value, 1 DoF.
  - Total: 3 DoF.
- Known information per pixel:
  - Reflectance $I(u, v) = \rho(u, v)\mathbf{s} \cdot \mathbf{n}(u, v)$: 1 equation linking 3 unknowns.
- Remaining DoFs: 2.
- For unambiguous solution, need remaining DoF = 0.
- In counterexample, 1DoF removed by assuming $\rho(u, v) \equiv 1$: Wrong!

Woodham Original PS – 1980

- How to remove Degrees of Freedom?

Woodham Original PS – 1980

- How to remove Degrees of Freedom?
- Woodham Solution: Use more images I^1, I^2,..., I^k.

Woodham Original PS – 1980

- How to remove Degrees of Freedom?
- Woodham Solution: Use more images $I^1, I^2, \ldots, I^k$.
- Don't change Camera Position – keep same depth function, normals, change lights.

Woodham Original PS – 1980

- How to remove Degrees of Freedom?
- Woodham Solution: Use more images I^1, I^2,..., I^k.
- Don't change Camera Position – keep same depth function, normals, change lights.
- k far (parallel) light sources $\mathbf{s}^1$,...,$\mathbf{s}^k$: k equations

$$\begin{cases} I^1(u,v) & = \rho(u,v)\mathbf{s}^1 \cdot \mathbf{n}(u,v) \\ I^2(u,v) & = \rho(u,v)\mathbf{s}^2 \cdot \mathbf{n}(u,v) \\ \dots & \dots \\ I^k(u,v) & = \rho(u,v)\mathbf{s}^k \cdot \mathbf{n}(u,v) \end{cases}$$

$$(\mathbf{s}^i \cdot \mathbf{n} = \mathbf{s}^i_1 \mathbf{n}^1 + \mathbf{s}^i_2 \mathbf{n}_2 + \mathbf{s}^i_3 \mathbf{n}_3)$$

Woodham Original PS – 1980

- How to remove Degrees of Freedom?
- Woodham Solution: Use more images I^1, I^2,..., I^k.
- Don't change Camera Position – keep same depth function, normals, change lights.
- k far (parallel) light sources $\mathbf{s}^1,\dots,\mathbf{s}^k$: k equations

$$\begin{cases} I^1(u,v) & = \rho(u,v)\mathbf{s}^1 \cdot \mathbf{n}(u,v) \\ I^2(u,v) & = \rho(u,v)\mathbf{s}^2 \cdot \mathbf{n}(u,v) \\ \dots & \dots \\ I^k(u,v) & = \rho(u,v)\mathbf{s}^k \cdot \mathbf{n}(u,v) \end{cases}$$

$$(\mathbf{s}^i \cdot \mathbf{n} = \mathbf{s}_1^i\,\mathbf{n}^1 + \mathbf{s}_2^i\,\mathbf{n}_2 + \mathbf{s}_3^i\,\mathbf{n}_3)$$

- Which k to choose? 3 DoF: $k \geq 3$. Exactly 3, more?

Woodham Original PS – 1980

- How to remove Degrees of Freedom?
- Woodham Solution: Use more images $I^1, I^2, \dots, I^k$.
- Don't change Camera Position – keep same depth function, normals, change lights.
- k far (parallel) light sources $\mathbf{s}^1, \dots, \mathbf{s}^k$: k equations

$$\begin{cases} I^1(u,v) & = \rho(u,v)\mathbf{s}^1 \cdot \mathbf{n}(u,v) \\ I^2(u,v) & = \rho(u,v)\mathbf{s}^2 \cdot \mathbf{n}(u,v) \\ \dots & \dots \\ I^k(u,v) & = \rho(u,v)\mathbf{s}^k \cdot \mathbf{n}(u,v) \end{cases}$$

$$(\mathbf{s}^i \cdot \mathbf{n} = \mathbf{s}^i_1 \mathbf{n}^1 + \mathbf{s}^i_2 \mathbf{n}_2 + \mathbf{s}^i_3 \mathbf{n}_3)$$

- Which k to choose? 3 DoF: $k \geq 3$. Exactly 3, more?
- Answer is geometric!

Woodham Original PS – 1980

- How to remove Degrees of Freedom?
- Woodham Solution: Use more images I^1, I^2,..., I^k.
- Don't change Camera Position – keep same depth function, normals, change lights.
- k far (parallel) light sources $\mathbf{s}^1,\dots,\mathbf{s}^k$: k equations

$$\begin{cases} I^1(u,v) & = \rho(u,v)\mathbf{s}^1 \cdot \mathbf{n}(u,v) \\ I^2(u,v) & = \rho(u,v)\mathbf{s}^2 \cdot \mathbf{n}(u,v) \\ \dots & \dots \\ I^k(u,v) & = \rho(u,v)\mathbf{s}^k \cdot \mathbf{n}(u,v) \end{cases}$$

$$(\mathbf{s}^i \cdot \mathbf{n} = \mathbf{s}^i_1 \mathbf{n}^1 + \mathbf{s}^i_2 \mathbf{n}_2 + \mathbf{s}^i_3 \mathbf{n}_3)$$

- Which k to choose? 3 DoF: $k \geq 3$. Exactly 3, more?
- Answer is geometric!
- From $(u,v) \mapsto \mathbf{n}(u,v)$ to surface? Integration of normals: out of scope; Matlab / Python functions will be provided.

Linear System for Normal + Albedo

- Woodham idea: Normal + Albedo Simultaneously

$$\mathbf{m} := \rho\mathbf{n}, \quad \rho = \|\mathbf{m}\|, \mathbf{n} = \frac{\mathbf{m}}{\|\mathbf{m}\|}$$

Linear System for Normal + Albedo

- Woodham idea: Normal + Albedo Simultaneously

$$\mathbf{m} := \rho\mathbf{n}, \quad \rho = \|\mathbf{m}\|, \mathbf{n} = \frac{\mathbf{m}}{\|\mathbf{m}\|}$$

- Problem if $\rho = 0$: absolutely black object – shadow.

Linear System for Normal + Albedo

- Woodham idea: Normal + Albedo Simultaneously

$$\mathbf{m} := \rho\mathbf{n}, \quad \rho = \|\mathbf{m}\|, \mathbf{n} = \frac{\mathbf{m}}{\|\mathbf{m}\|}$$

- Problem if $\rho = 0$: absolutely black object – shadow.
- k Lambert's laws for a given pixel at $[u, v]$

$$\underbrace{\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_k \end{bmatrix}}_{\text{Image vector } \mathbf{I}, k\times 1} = \underbrace{\begin{bmatrix} \mathbf{s}_1^1 & \mathbf{s}_1^2 & \mathbf{s}_1^3 \\ \mathbf{s}_2^1 & \mathbf{s}_2^2 & \mathbf{s}_2^3 \\ \vdots & \vdots & \vdots \\ \mathbf{s}_k^1 & \mathbf{s}_k^2 & \mathbf{s}_k^3 \end{bmatrix}}_{\text{Light matrix } M_{\mathbf{s}}, k\times 3} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

Linear System for Normal + Albedo

- Woodham idea: Normal + Albedo Simultaneously

$$\mathbf{m} := \rho \mathbf{n}, \quad \rho = \|\mathbf{m}\|, \mathbf{n} = \frac{\mathbf{m}}{\|\mathbf{m}\|}$$

- Problem if $\rho = 0$: absolutely black object – shadow.
- k Lambert's laws for a given pixel at $[u, v]$

$$\underbrace{\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_k \end{bmatrix}}_{\text{Image vector } \mathbf{I}, k\times 1} = \underbrace{\begin{bmatrix} \mathbf{s}_1^1 & \mathbf{s}_1^2 & \mathbf{s}_1^3 \\ \mathbf{s}_2^1 & \mathbf{s}_2^2 & \mathbf{s}_2^3 \\ \vdots & \vdots & \vdots \\ \mathbf{s}_k^1 & \mathbf{s}_k^2 & \mathbf{s}_k^3 \end{bmatrix}}_{\text{Light matrix } M_{\mathbf{s}}, k\times 3} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

- Number of solutions?

Linear System for Normal + Albedo

- Woodham idea: Normal + Albedo Simultaneously

$$\mathbf{m} := \rho\mathbf{n}, \quad \rho = \|\mathbf{m}\|, \mathbf{n} = \frac{\mathbf{m}}{\|\mathbf{m}\|}$$

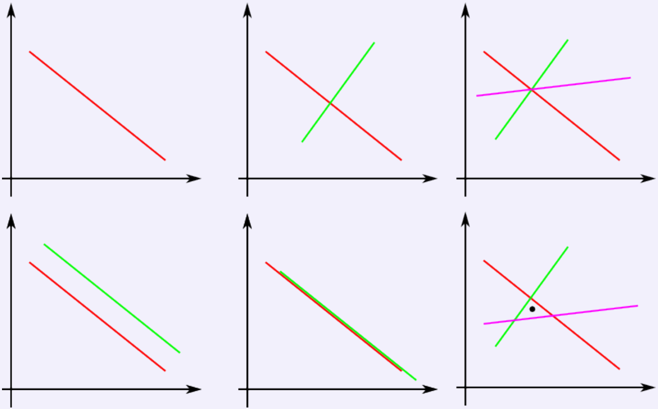
- Problem if $\rho = 0$: absolutely black object – shadow.
- k Lambert's laws for a given pixel at $[u, v]$

$$\underbrace{\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_k \end{bmatrix}}_{\text{Image vector } \mathbf{I}, k\times 1} = \underbrace{\begin{bmatrix} \mathbf{s}_1^1 & \mathbf{s}_1^2 & \mathbf{s}_1^3 \\ \mathbf{s}_2^1 & \mathbf{s}_2^2 & \mathbf{s}_2^3 \\ \vdots & \vdots & \vdots \\ \mathbf{s}_k^1 & \mathbf{s}_k^2 & \mathbf{s}_k^3 \end{bmatrix}}_{\text{Light matrix } M_{\mathbf{s}}, k\times 3} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

- Number of solutions?
- Can vary from none to a lot!

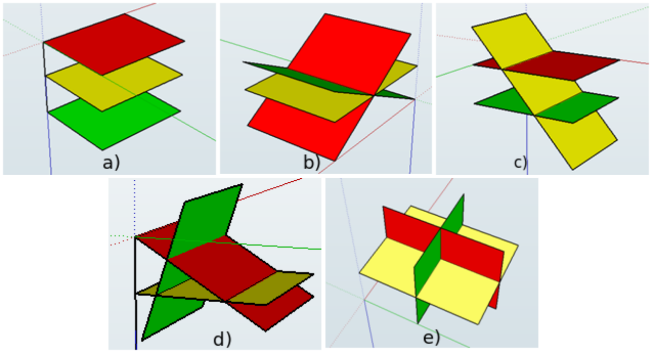
Linear Algebra Again

- System of equations in 2 unknown.



- Need at least 2 independent equations! ...but not 3!
- For 3 or more: concept of “closest solution in least squares”.

In 3D



Same light source, incompatible measurements, b) coplanar light sources, compatible measurements, c) 2 light source, 3 incompatible measurements, d) coplanar light sources, incompatible measurements, e) 3 non coplanar light sources

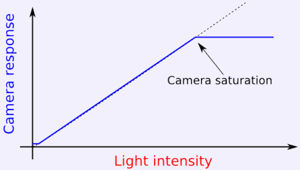
Picture from Guillermo Bautista, mathandmultimedia.com

What Can Go Wrong?

- Parameters / devices:
  - Measurement errors: Camera?
  - Coplanar light sources: example?
- Reflectance
  - Lambert's law only valid for matte materials: specularities.
  - Shadows / penumbra: non black cast shadow areas.
- More?

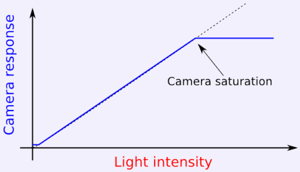
What Can Go Wrong?

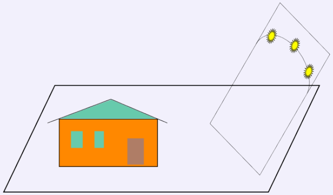
- Parameters / devices:
  - Measurement errors: Camera?
  - Coplanar light sources: example?
- Reflectance
  - Lambert's law only valid for matte materials: specularities.
  - Shadows / penumbra: non black cast shadow areas.
- More?



What Can Go Wrong?

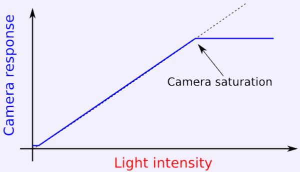
- Parameters / devices:
  - Measurement errors: Camera?
  - Coplanar light sources: example?
- Reflectance
  - Lambert's law only valid for matte materials: specularities.
  - Shadows / penumbra: non black cast shadow areas.
- More?

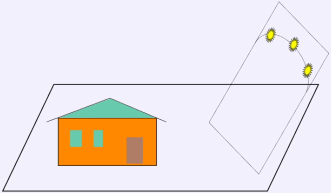




What Can Go Wrong?

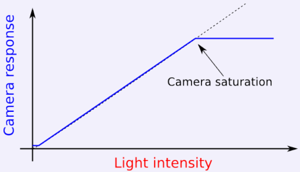
- Parameters / devices:
  - Measurement errors: Camera?
  - Coplanar light sources: example?
- Reflectance
  - Lambert's law only valid for matte materials: specularities.
  - Shadows / penumbra: non black cast shadow areas.
- More?

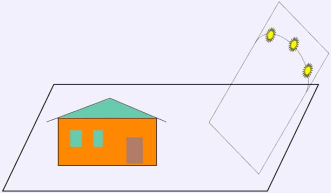


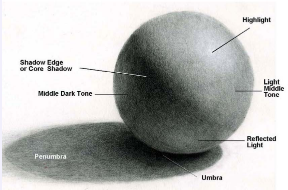


What Can Go Wrong?

- Parameters / devices:
  - Measurement errors: Camera?
  - Coplanar light sources: example?
- Reflectance
  - Lambert's law only valid for matte materials: specularities.
  - Shadows / penumbra: non black cast shadow areas.
- More?







Solving For Normals: Pseudo-Inverse Approach

Assume 3 or more non coplanar light sources.

- Equation $\mathbf{I} = M_\mathbf{s}\,\mathbf{m}$ may not have solution if $\mathbf{M_s}$ is $k \times 3$, $k > 3$.
- Via Moore-Penrose Pseudo-Inverse $M_\mathbf{s}^\dagger$, solution of

$$\mathbf{m} \text{ such that } \|\mathbf{I} - M_\mathbf{s}\mathbf{m}\|^2 = \min: \quad \mathbf{m} = M_\mathbf{s}^\dagger \mathbf{I}.$$

Pros.

- Good news: Matlab `pinv` function, Python `pinv` function in package `numpy.linalg`.

Cons.

- Does not separate wrong and accurate measurements.

Solving For Normals: Equations Selection - I

In a nutshell.

- Per pixel: find best constraints: 3 “good measurements” I^a, I^b, I^c and corresponding “good lights” $\mathbf{s}^a$, $\mathbf{s}^b$, $\mathbf{s}^c$.

$$\underbrace{\begin{bmatrix} I^a \\ I^b \\ I^c \end{bmatrix}}_{\mathbf{I}_{abc}} = \underbrace{\begin{bmatrix} \mathbf{s}_1^a & \mathbf{s}_2^a & \mathbf{s}_3^a \\ \mathbf{s}_1^b & \mathbf{s}_2^b & \mathbf{s}_3^b \\ \mathbf{s}_1^c & \mathbf{s}_2^c & \mathbf{s}_3^c \end{bmatrix}}_{M_{abc}} \begin{bmatrix} \mathbf{m}^1 \\ \mathbf{m}^2 \\ \mathbf{m}^3 \end{bmatrix}$$

Good measurements.

- I_i good measurement: not too small (shadow, penumbra) and not too large (saturation, potential specularity).
- Good light sources: $\det M_{abc}$ as large as possible.

Solving For Normals: Equations Selection - II

Pros.

- Best system of equations per pixel.

Cons.

- Lack of spatial coherence: different lights can be chosen for neighbor pixels.
- Need thresholds for intensities: parameters of the algorithm.
- More complicated to code.

Size Matters.

- For relatively small k: equation selection can be a good idea.
- For very large k (1000, more...) pseudo-inverse very good: statistical reason.

Algorithm in a Nutshell

Input.

- k known parallel light sources $\mathbf{s}_1, \ldots \mathbf{s}_k$. k recorded images $I_1, \ldots, I_k$.

Normals and Albedo recovery.

- For each (valid) pixel $[u, v]$ in image domain:

1. Solve $\mathbf{m}(u, v)$ either via pseudo-inverse or equation selection.
2. Get albedo and normal:

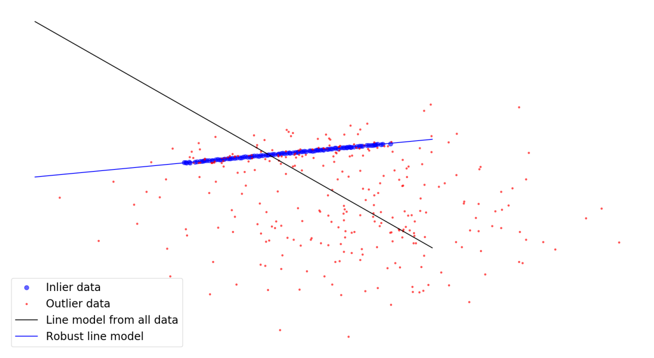
$$\rho(u, v) = \|\mathbf{m}(u, v)\|, \quad \mathbf{n}(u, v) = \frac{1}{\rho(u, v)}\mathbf{m}(u, v).$$

Surface Recovery.

- Get surface from normals via surface integration.
- May "paint surface" with albedo.

Robust Statistics

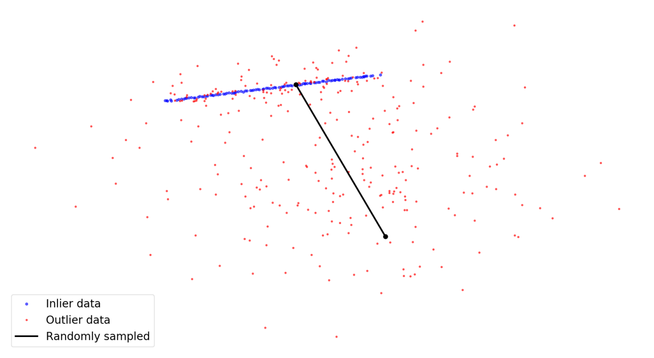
- Classical Least-Square estimate may fail due to outliers in data



- LS estimated line: the black one.
- The correct line: blue.
- Extremely active research topic, mixing statistics and very complicated optimisation.
- We explore a classical one: RANSAC

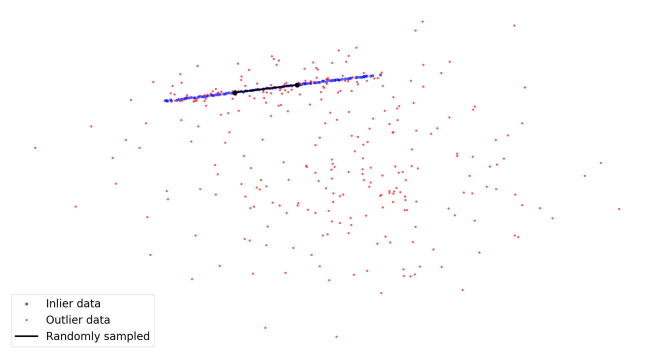
A Classical in Computer Vision: RANSAC

- RANSAC = RANdom SAmplic Consensus: Fishler & Bolles, 1981.
- Idea: if the data contains enough *inliers*, random sampling should provided some, after possibly many repetitions.



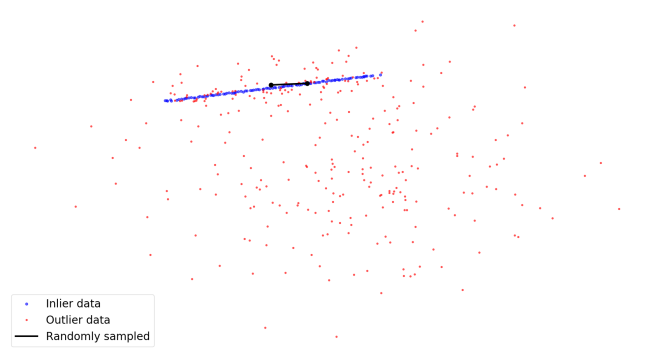
A Classical in Computer Vision: RANSAC

- RANSAC = RANdom SAmplic Consensus: Fishler & Bolles, 1981.
- Idea: if the data contains enough *inliers*, random sampling should provided some, after possibly many repetitions.



A Classical in Computer Vision: RANSAC

- RANSAC = RANdom SAmplic Consensus: Fishler & Bolles, 1981.
- Idea: if the data contains enough *inliers*, random sampling should provided some, after possibly many repetitions.



2D Line Fitting RANSAC

- From two points (x_1, y_1) and (x_2, y_2): a line $L : y = ax + b$,

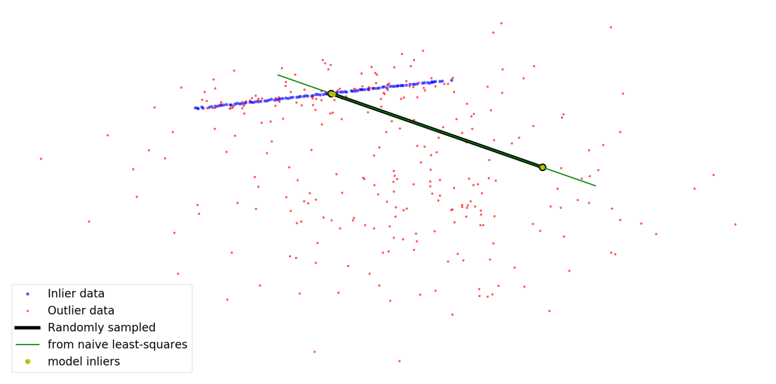
$$y = \underbrace{\frac{y_2 - y_1}{x_2 - x_1}}_{a} x + \underbrace{\frac{y_1 x_2 - y_2 x_1}{x_2 - x_1}}_{b}$$

- Does it fit the data well enough?
- For each data point (x_i, y_i): compute residual / fitting square error

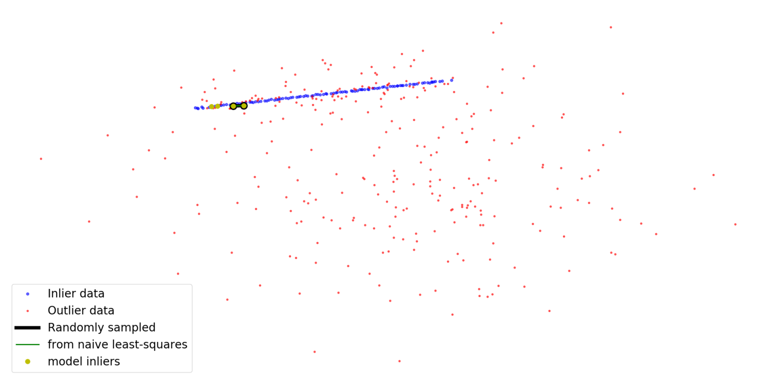
$$r_i = |y_i - ax_i - b|^2$$

- if $r_i < \tau$: predefined threshold: declare (x_i, y_i) inlier.
- if $r_i \geq \tau$: declare (x_i, y_i) outlier.
- Choose the model (a, b) with most inliers.
- Reestimate the model only from inliers (least-squares)

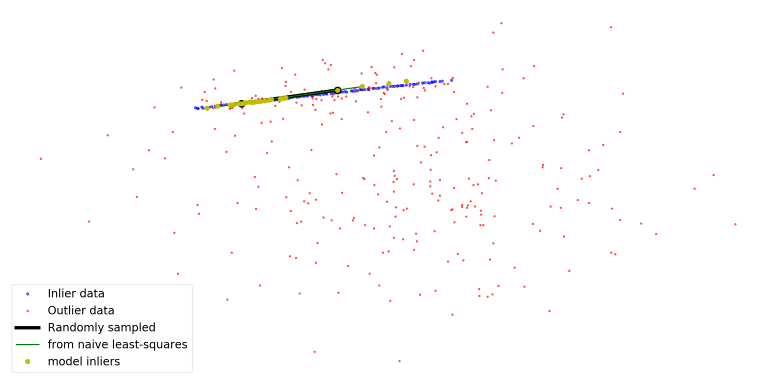
Sample run in 2D: Bad



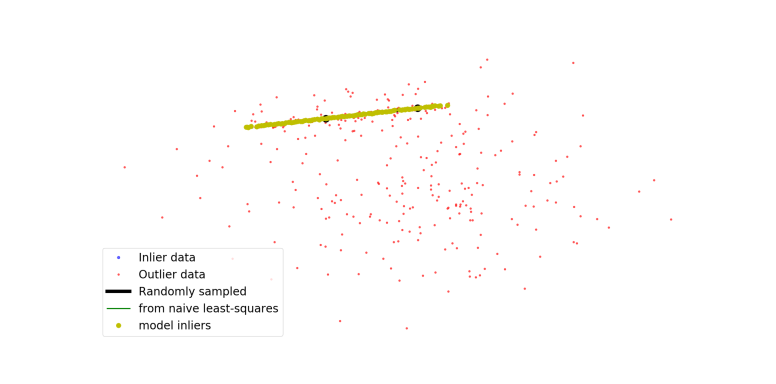
Sample run in 2D: Better



Sample run in 2D: Getting C lose



Sample run in 2D: Near Perfect



How many trials

- If model needs at least n points to be fitted, can we guarantee that one can sample n inliers at a given trial?
- Number of trial can be estimated from inlier frequency

$$\text{inlier frequency} = \frac{\text{number of inliers}}{\text{total number of data points}}$$

- inlier frequency can be estimated from trial!
- Each trial reports a number of inlier for selected model.
- Take this number as lower bound for number of inliers.

How many trials

- Fishler and Bolles show: Assume inlier frequency is w, and n inliers are necessary to fit a model
- to be sure with probability z that one can sample n good points in a trial, number of trials N at least

$$N \approx \frac{\log(1 - z)}{\log(1 - w^n)}$$

- Depends critically on threshold for accepting / rejecting fits!

RANSAC and estimation of $\rho \mathbf{n} = \mathbf{m}$

- Data + Observation:

RANSAC and estimation of $\rho \mathbf{n} = \mathbf{m}$

- Data + Observation:
  - k light source vectors $s_1, \ldots, s_k$

RANSAC and estimation of $\rho \mathbf{n} = \mathbf{m}$

- Data + Observation:
  - k light source vectors $s_1, \ldots, s_k$
  - Per pixel, k intensity values $I_1(p), \ldots I_k(p)$.

RANSAC and estimation of $\rho\mathbf{n} = \mathbf{m}$

- Data + Observation:
  - k light source vectors $s_1, \dots, s_k$
  - Per pixel, k intensity values $I_1(p), \dots I_k(p)$.
- Relation between model $\mathbf{m}$ and observations: Lambert's law

$$I_i(p) \approx s_i^T \mathbf{m}(p)$$

RANSAC and estimation of $\rho\mathbf{n} = \mathbf{m}$

- Data + Observation:
  - k light source vectors $s_1, \dots, s_k$
  - Per pixel, k intensity values $I_1(p), \dots I_k(p)$.
- Relation between model **m** and observations: Lambert's law

$$I_i(p) \approx s_i^T \mathbf{m}(p)$$

- Minimal number of light vectors and corresponding measured intensities for determining a **m**?

RANSAC and estimation of $\rho \mathbf{n} = \mathbf{m}$

- Data + Observation:
  - k light source vectors $s_1, \ldots, s_k$
  - Per pixel, k intensity values $I_1(p), \ldots I_k(p)$.
- Relation between model **m** and observations: Lambert's law

$$I_i(p) \approx s_i^T \mathbf{m}(p)$$

- Minimal number of light vectors and corresponding measured intensities for determining a **m**? 3! (why?)

RANSAC and estimation of $\rho\mathbf{n} = \mathbf{m}$

- Data + Observation:
  - k light source vectors $s_1, \dots, s_k$
  - Per pixel, k intensity values $I_1(p), \dots I_k(p)$.
- Relation between model **m** and observations: Lambert's law

$$I_i(p) \approx s_i^T \mathbf{m}(p)$$

- Minimal number of light vectors and corresponding measured intensities for determining a **m**? 3! (why?)
- Can happen that 3 selected light vector are nearly colinear:

RANSAC and estimation of $\rho\mathbf{n} = \mathbf{m}$

- Data + Observation:
  - k light source vectors $s_1, \dots, s_k$
  - Per pixel, k intensity values $I_1(p), \dots I_k(p)$.
- Relation between model $\mathbf{m}$ and observations: Lambert's law

$$I_i(p) \approx s_i^T \mathbf{m}(p)$$

- Minimal number of light vectors and corresponding measured intensities for determining a $\mathbf{m}$? 3! (why?)
- Can happen that 3 selected light vector are nearly colinear: Retry!

RANSAC and estimation of $\rho \mathbf{n} = \mathbf{m}$

- Data + Observation:
  - k light source vectors $s_1, \dots, s_k$
  - Per pixel, k intensity values $I_1(p), \dots I_k(p)$.
- Relation between model **m** and observations: Lambert's law

$$I_i(p) \approx s_i^T \mathbf{m}(p)$$

- Minimal number of light vectors and corresponding measured intensities for determining a **m**? 3! (why?)
- Can happen that 3 selected light vector are nearly colinear: Retry!
- NOTE: One RANSAC must be run for EACH pixel! (Not as bad as it seems...)

RANSAC and estimation of $\rho\mathbf{n} = \mathbf{m}$

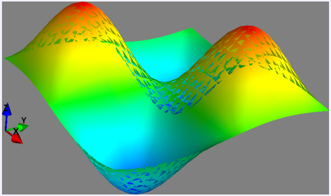
- Data + Observation:
  - k light source vectors $s_1, \dots, s_k$
  - Per pixel, k intensity values $I_1(p), \dots I_k(p)$.
- Relation between model **m** and observations: Lambert's law

$$I_i(p) \approx s_i^T \mathbf{m}(p)$$

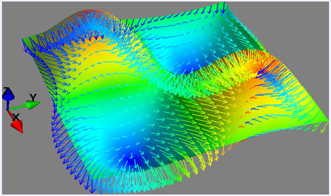
- Minimal number of light vectors and corresponding measured intensities for determining a **m**? 3! (why?)
- Can happen that 3 selected light vector are nearly colinear: Retry!
- NOTE: One RANSAC must be run for EACH pixel! (Not as bad as it seems...)
- Good news: a special RANSAC `ransac_3dvector()` for estimations of **m** vectors is available from the `ps_utils.py` module (Absalon).

Outline

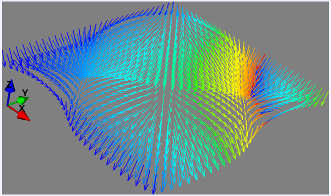
Nice Normal Field. Good Integration



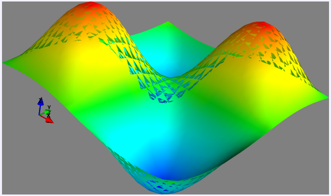
Original surface



With normal field plotted on it

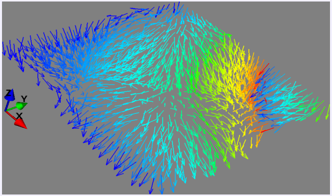


Just the normal field

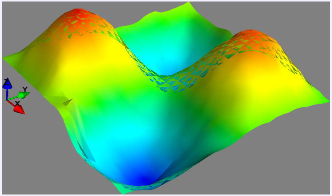


Numerical integration

Not So Nice Normal Field, Noisy Result

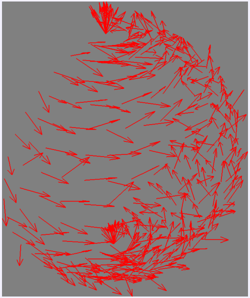


Noisy normal field

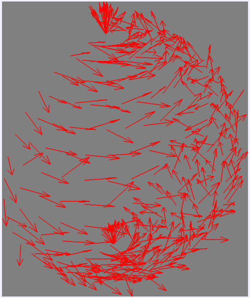


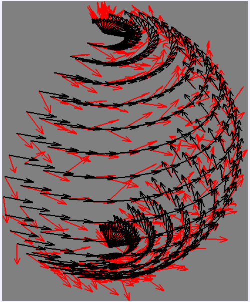
Noisy surface recovery

Noisy Vector field?. Denoising



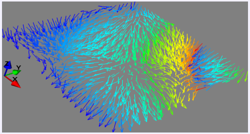
Noisy Vector field?. Denoising



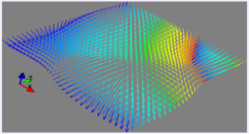


- Gaussian Convolution adapted to normal fields:
- each new vector value must still have norm 1.
- A bit complicated, uses differential geometry:-)

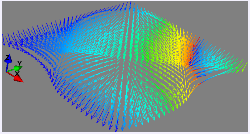
Denoising and integration



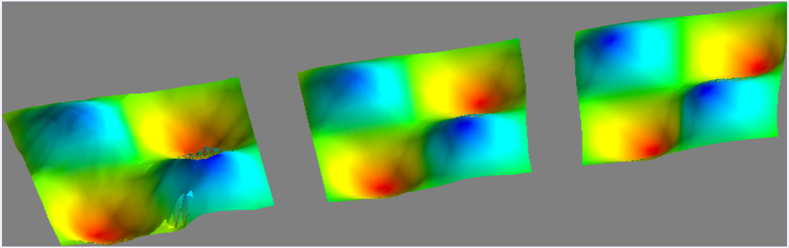
Noisy normal field



Denoised field



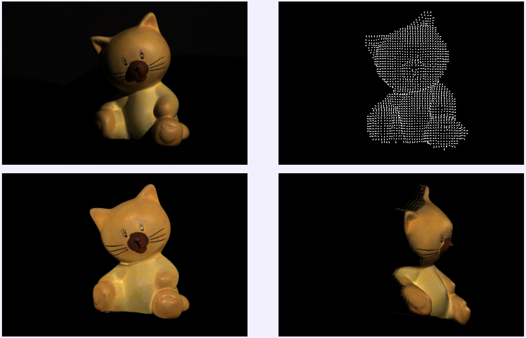
Original clean



- In `ps_utils.py` module, `smooth_normal_field()` can do it.
- For the assignment, keep its default parameters.

Outline

Process



Top left: 1 of the input images, right: normal field. Bottom left: albedo, right: a 3D reconstruction.

Mezigue

Mezigue

Mezigue

Literature, available in AbasIon

- Horn, Shape from shading 1970, 1975
- Woodham, Photometric Stereo, 1980
- Fishler-Bolles, RANSAC, 1981

Summary

- φῶς/φωτός
- μέτρον
- στερεός

Summary

- φῶς/φωτός
- μέτρον
- στερεός

Summary

- φῶς/φωτός
- μέτρον
- στερεός