Belief Propagation with Directional Statistics for solving the Shape-from-Shading problem

Tom S. F. Haines & Richard C. Wilson
thaines,wilson@cs.york.ac.uk

University of York

15th October 2008

Roadmap

Notes before we continue:

- Presenting an improved version of the algorithm.
- Binaries and test data can be obtained from *www.thaines.net*

Shape from Shading

- Takes as input a single image and a function from surface orientation to surface irradiance.
- The output is the shape of the object - either as surface orientation or depth.
- Surface irradiance only constrains one degree of freedom when surface orientation has two degrees of freedom - the problem is ill posed.
- Information is required to resolve the ambiguity - we use the typical smoothness assumption.
- We have no integration constraint. This would force the surface orientation to be consistent with a depth map.

Summary

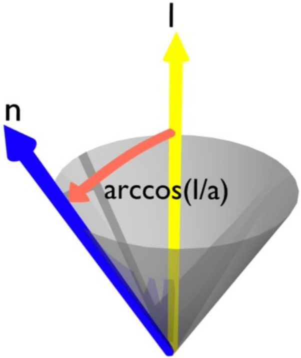
Input

Ideal Output

Actual Output

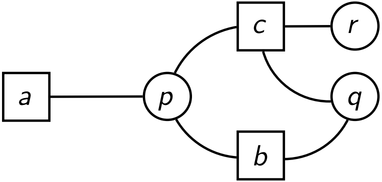
Lambertian Reflectance

- Governed by the equation $I = a(\hat{\mathbf{n}} \cdot \hat{\mathbf{l}})$, where I is irradiance, a albedo, $\hat{\mathbf{l}}$ the direction to the light source and $\hat{\mathbf{n}}$ is surface orientation.
- We know all except $\hat{\mathbf{n}}$, which we want to find.
- This constrains the surface orientation [Worthington & Hancock, 99] at each pixel onto a cone with axis-angle $\cos^{-1}(I/a)$.



Belief Propagation

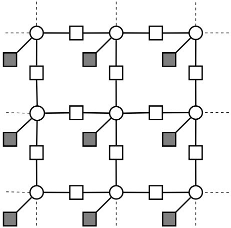
- Used to calculate the marginals of a set of random variables given the posterior probability - $P(\mathbf{X}) = \prod_{i \in F} P_i(u_i)$.
  - $\mathbf{X}$ is a set of random variables, for example $\mathbf{X} = \{p, q, r\}$.
  - F is a set of factors, for example $F = \{a, b, c\}$.
  - The u_i are the cliques, $\forall i \in F \bullet u_i \subset \mathbf{X}$, for example $u_a = \{p\}$, $u_b = \{p, q\}$, $u_c = \{p, q, r\}$.
- Can be represented graphically:

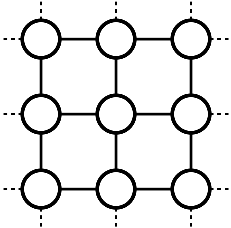


- Solved by passing messages in the graphical representation. Smaller cliques are faster ($|u_i|$).
- Graphs with loops require an iterative approach.

Markov Random Fields

- In this case we are concerned with a pairwise Markov Random Field.
- Each pixel in the image has its surface orientation represented by a random variable.
- Adjacent pixels share a factor that expresses the smoothing assumption.
- Each pixel has a factor derived from the Lambertian reflectance equation.





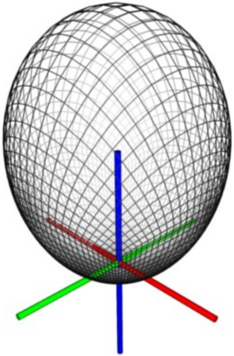
Directional Statistics

- We need to represent surface orientation, $\hat{\mathbf{x}} = \Re^3, |\hat{\mathbf{x}}| = 1$.
- Belief propagation requires a probabilistic representation.
- Traditional statistics won't work as angles wrap around ($360^\circ = 0^\circ$) - we need to use *Directional Statistics*.
- Need a distribution that satisfies two properties:
  - *Multiplication:* We need to multiply the distribution to get the *same* distribution, otherwise belief propagation would not be tractable.
  - *Cone:* We need a distribution which can represent the cone constraint, as obtained from the Lambertian reflectance equation.

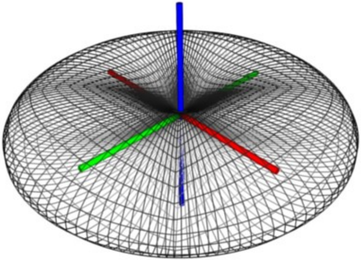
The Fisher-Bingham $[\mathrm{FB}_8]$ Distribution

- The Fisher distribution is $\propto \exp(\mathbf{u}^T \hat{\mathbf{x}})$, $|\mathbf{u}| = k$.
- The Bingham distribution is $\propto \exp(\hat{\mathbf{x}}^T \mathbf{A} \hat{\mathbf{x}})$. $(\mathbf{A}^T = \mathbf{A})$
- FB_8 is the multiplication of the Fisher distribution and the Bingham distribution, and is $\propto \exp(\mathbf{u}^T \hat{\mathbf{x}} + \hat{\mathbf{x}}^T \mathbf{A} \hat{\mathbf{x}})$.
- It satisfies the required properties:
    - *Multiplication:* If represented by $\Omega[\mathbf{u}, \mathbf{A}]$ then $\Omega[\mathbf{u}, \mathbf{A}]\Omega[\mathbf{v}, \mathbf{B}] = \Omega[\mathbf{u} + \mathbf{v}, \mathbf{A} + \mathbf{B}]$, another FB_8 distribution.
    - *Cone:* The Bingham-Mardia distribution, a sub-distribution of the FB_8 distribution, may represent the lambertian reflectance information. Given by $\Omega[2k\cos(\theta)\hat{\mathbf{u}}, -k\hat{\mathbf{u}}\hat{\mathbf{u}}^T]$ its maxima form a small circle defined by a cone with axis-angle θ.

Visualisation I

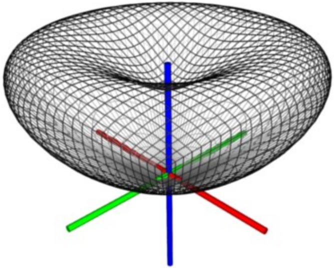


Fisher distribution, $k = 2$.

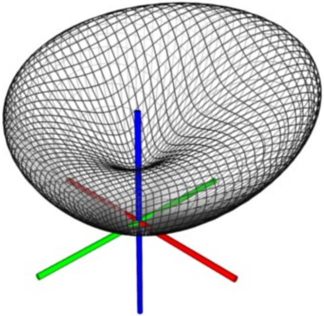


Bingham distribution, $\alpha = \beta = 5$.

Visualisation II



Bingham-Mardia distribution,
$k = 8, \texttt{angle} = 45^\circ$.

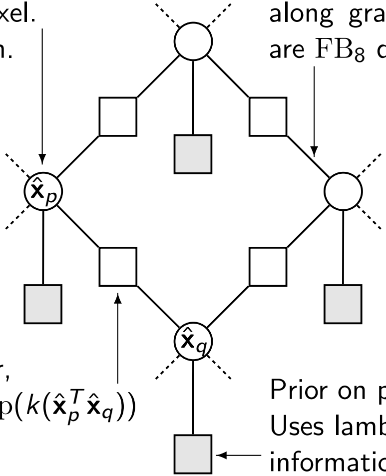


Fisher-Bingham distribution,
$k = 12, \alpha = \beta = 9.0$.

Putting it all together

Random variable - surface orientation of a pixel. A FB_8 distribution.

Solved by passing messages along graph edges. Messages are FB_8 distributions.



Smoothing factor, $P_{pq}(\hat{\mathbf{x}}_p, \hat{\mathbf{x}}_q) \propto \exp(k(\hat{\mathbf{x}}_p^T \hat{\mathbf{x}}_q))$

Prior on pixel orientation. Uses lambertian reflectance information, $\Omega[2k(I_q/a)\hat{\mathbf{l}}, -k\hat{\mathbf{l}}\hat{\mathbf{l}}^T]$

The Problem with FB_8 Belief Propagation

- We implement the smoothing assumption as a joint probability distribution, $P_{pq}(\hat{\mathbf{x}}_p, \hat{\mathbf{x}}_q) \propto \exp(k(\hat{\mathbf{x}}_p^T \hat{\mathbf{x}}_q))$
- Using the message passing equations for belief propagation this gives a message from p to q of $\int_{\mathbb{S}^2} \exp(k(\hat{\mathbf{x}}_p^T \hat{\mathbf{x}}_q)) t(\hat{\mathbf{x}}_p) \delta \hat{\mathbf{x}}_p$ where $t(\hat{\mathbf{x}}_p)$ is a FB_8 distribution.
- This is recognisable as the convolution of a FB_8 distribution by a Fisher distribution, the solution of which is not a FB_8 distribution.
- But the messages need to be FB_8 distributions - an approximation has to be made.

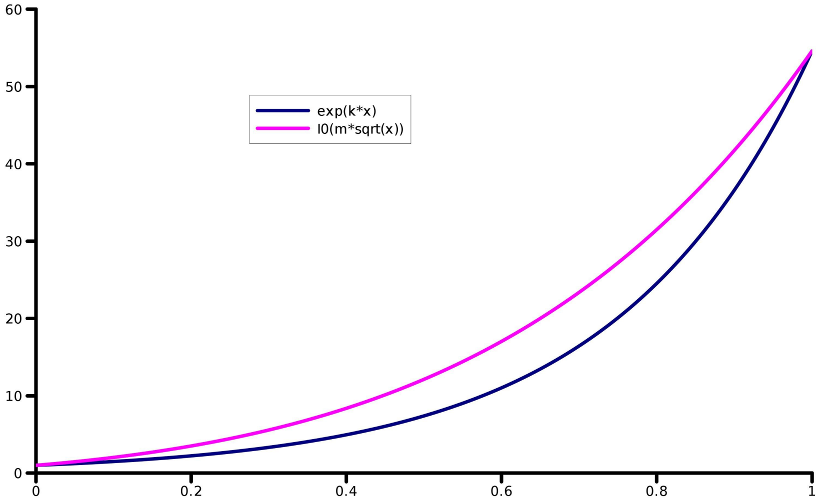
Convolution

- Convolution of the FB_8 distribution is approximated with three steps:
  1. The FB_8 distribution is converted to a mixture of Fisher distributions.
  2. Each Fisher distribution is convolved.
  3. The FB_8 distribution is fitted to the resulting mixture of Fisher distributions.
- The solution to the second step comes from [Mardia & Jupp, 00], where a 2D approximation is given. Extension to 3D is simple.

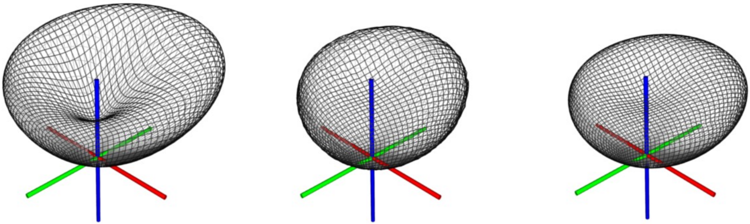
Convolution Justification

- The approximation error of the convolution procedure is large.
- However, the FB_8 to mixture conversion is such that all six critical points keep their directions and probability ratios, i.e. the error at the minimas, maximas and other critical points is effectively zero.
- The novel approximation used approximates $\exp(x)$ with $\mathbf{I}_0(\sqrt{x})$, the modified Bessel function of the first kind, order zero. These are both exponentially shaped functions, parametrised so they match at the critical points. Therefore the accuracy is highest at the critical points, with error increasing as you move away from them.
- As we are ultimately interested in critical points alone the approximation does not prove to be a problem.

Visualisation of Conversion



Visualisation of Full Approximation



Input. Convolved, brute force. Convolved, approximation.

- Convolved by a Fisher distribution with $k = 12$.

Final Details

- Given the ability to pass messages the remainder of the algorithm is typical belief propagation.
- The concentration value of the smoothing term can be set constant over the entire image or modulated depending on how likely adjacent pixels are to be the same orientation.
- The prior information for each pixel is derived from the Lambertian lighting information, gradient information and a boundary constraint.

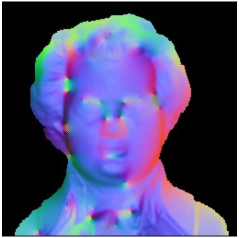
Post Processing

- Running belief propagation to convergence gives us a set of probability distributions - we want actual directions.
- Finding the most likely directions of a FB_8 distribution is done by solving the related *closest/furthest point on an ellipsoid* problem.
- Each distribution has two maxima. Due to the Concave-Convex ambiguity selecting between them with local information will not produce a consistent global solution.
- Min-sum belief propagation is used to make the choice.

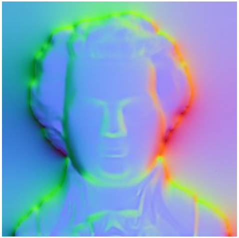
Results: Mozart at 90°


Mozart, lit head on.


Ground truth


[Worthington & Hancock, 99]

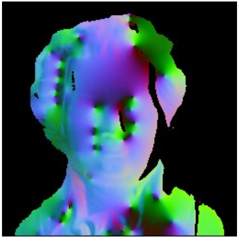

[Lee & Kuo, 94]


Presented algorithm

Results: Mozart at 45°

Mozart, lit at 45° from head on.

Ground truth



[Worthington & Hancock, 99]

[Lee & Kuo, 94]

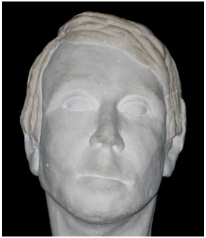
Presented algorithm

Quantitative Results: Synthetic

Mozart 90°	< 1°	< 2°	< 3°	< 4°	< 5°	< 10°	< 15°	< 20°	< 25°
Lee & Kuo	0.1	0.5	1.3	2.3	3.8	17.4	36.2	52.3	66.4
Worthington & Hancock	3.3	7.3	11.3	15.3	19.6	34.9	47.5	57.0	64.3
Presented Algorithm	0.4	1.4	3.3	6.2	10.1	30.7	48.9	65.5	75.5

Mozart 45°	< 1°	< 2°	< 3°	< 4°	< 5°	< 10°	< 15°	< 20°	< 25°
Lee & Kuo	0.2	0.7	1.5	2.5	3.8	16.1	35.0	54.7	67.2
Worthington & Hancock	1.3	3.2	4.8	6.3	8.0	15.3	23.0	30.7	37.7
Presented Algorithm	0.2	0.5	1.0	1.5	2.0	5.7	10.0	14.7	19.9

Results: Head

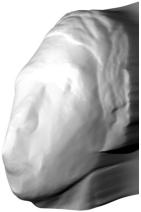


Head, lit head on.

Ground truth

[Worthington & Hancock, 99]

[Lee & Kuo, 94]

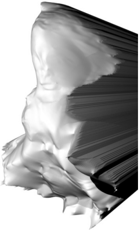


Presented algorithm

Results: Venus

Venus, lit head on.

Ground truth



[Worthington & Hancock, 99]

[Lee & Kuo, 94]

Presented algorithm

Quantitative Results: Real I

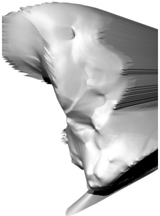
Head	$< 1^\circ$	$< 2^\circ$	$< 3^\circ$	$< 4^\circ$	$< 5^\circ$	$< 10^\circ$	$< 15^\circ$	$< 20^\circ$	$< 25^\circ$
Lee & Kuo	0.3	1.1	2.3	3.8	5.7	18.8	34.1	47.1	58.8
Worthington & Hancock	0.1	0.7	1.4	2.6	4.0	13.6	25.3	38.6	51.7
Presented Algorithm	0.5	1.9	4.2	7.3	11.1	33.8	49.8	62.2	72.2

Venus	$< 1^\circ$	$< 2^\circ$	$< 3^\circ$	$< 4^\circ$	$< 5^\circ$	$< 10^\circ$	$< 15^\circ$	$< 20^\circ$	$< 25^\circ$
Lee & Kuo	0.0	0.4	0.8	1.5	2.4	10.6	23.6	36.1	48.4
Worthington & Hancock	0.1	0.5	1.1	1.8	2.7	9.2	16.8	24.4	32.9
Presented Algorithm	0.1	0.5	1.1	2.1	3.4	14.5	28.9	42.1	53.1

Results: Bard

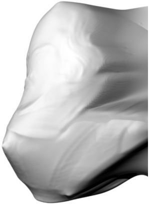
Bard, lit head on.

Ground truth



[Worthington & Hancock, 99]

[Lee & Kuo, 94]

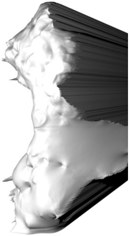


Presented algorithm

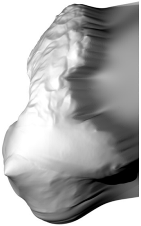
Results: Sunev

Sunev, lit head on.

Ground truth



[Worthington & Hancock, 99]



Presented algorithm

Quantitative Results: Real II

Bard	$< 1^\circ$	$< 2^\circ$	$< 3^\circ$	$< 4^\circ$	$< 5^\circ$	$< 10^\circ$	$< 15^\circ$	$< 20^\circ$	$< 25^\circ$
Lee & Kuo	0.0	0.3	0.6	1.2	1.9	8.1	19.4	27.5	33.9
Worthington & Hancock	0.1	0.5	1.1	1.9	2.9	9.0	15.2	22.3	30.1
Presented Algorithm	0.0	0.5	1.1	2.0	2.9	10.7	18.2	25.3	32.2

Sunev	$< 1^\circ$	$< 2^\circ$	$< 3^\circ$	$< 4^\circ$	$< 5^\circ$	$< 10^\circ$	$< 15^\circ$	$< 20^\circ$	$< 25^\circ$
Lee & Kuo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.7
Worthington & Hancock	0.1	0.4	0.8	1.5	2.2	7.7	15.2	23.7	32.0
Presented Algorithm	0.2	0.6	1.3	2.2	3.4	13.5	27.7	41.8	54.2

Time & Space

- Time:
  - Lee & Kuo takes 20 hours.
  - Worthington & Hancock takes 1 hour, 50 minutes.
  - The presented algorithm takes 3 minutes and 14 seconds.
- The above should be taken with caution as Lee & Kuo and Worthington & Hancock both require an iteration count be set. These have been set higher than necessary to obtain a good rather then fast result.
- Space:
  - Lee & Kuo stores depth in a hierarchy - 1 1/3 floats per pixel.
  - Worthington & Hancock stores surface orientation - 3 floats per pixel.
  - The presented algorithm however has to store four messages for each pixel, each consisting of a FB_8 distribution at 12 floats, - 48 floats per pixel.

Conclusions

- A robust shape from shading algorithm has been presented.
- Results show it to be better than the presented competition for real input shot with the light source at the camera.
- The lack of an integration constraint is its biggest weakness...
- ...especially as it leads to poor handling of angled light sources due to the use of gradient information.
- The probabilistic nature of the algorithm allows tight integration with other sources of information.

Questions?

- *Remember:* Test data available at *www.thaines.net*