Introduction to Computer Vision

Michael J. Black

Oct 2009

Finish PCA and classification

Start motion estimation

Photo Forensics

Hany Farid, Dartmouth

Tuesday, October 27th at 4pm

(Marcuvitz Auditorium in Sidney Frank)

In an attempt to quell rumors regarding the health of North Korea's leader Kim Jong-Il, the North Korean government released a series of photographs showing a healthy and active Kim Jong-Il. Shortly after their release the BBC claimed that the photographs were doctored. The article pointed to purported visual incongruities, which were claimed to be the result of photo tampering.

The BBC was wrong. Because judgments of photo authenticity are often made by eye, we wondered how reliable the human visual system is in detecting discrepancies that might arise from photo tampering. We describe three experiments that show that the human visual system is remarkably inept at detecting simple geometric inconsistencies in shadows, reflections, and planar perspective distortions. We also describe computational methods that can be applied to detect the inconsistencies that seem to elude the human visual system.

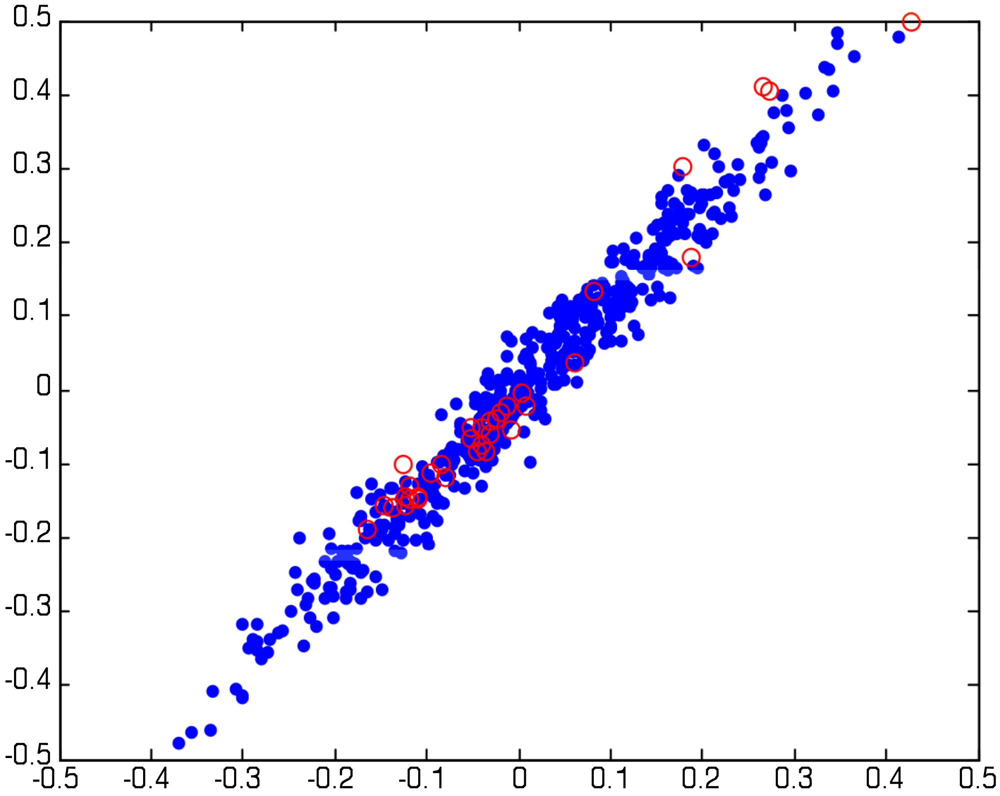
If you're curious, here is a video clip of Hany on Nova Science Now describing some of this work: http://www.pbs.org/wgbh/nova/sciencenow/0301/03.html

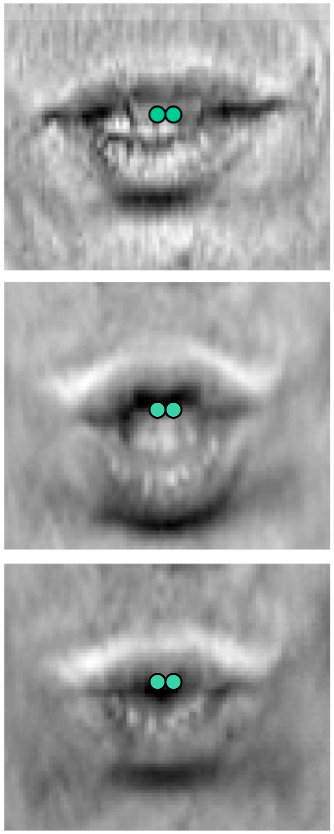
Goals

- Finish probability and classification
- Clear up any questions about PCA and asign 2 parts 1 and 2.

- Introduce motion estimation

Example: Covariance





Points in PCA space

[U,D]=eig(C)

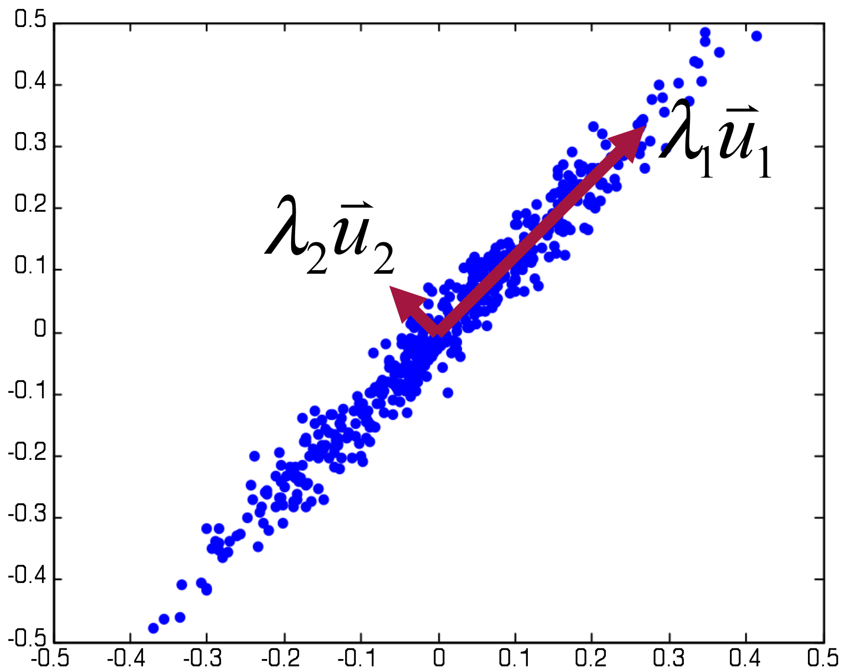
U =

-0.7876	0.6162
0.6162	0.7876

D =

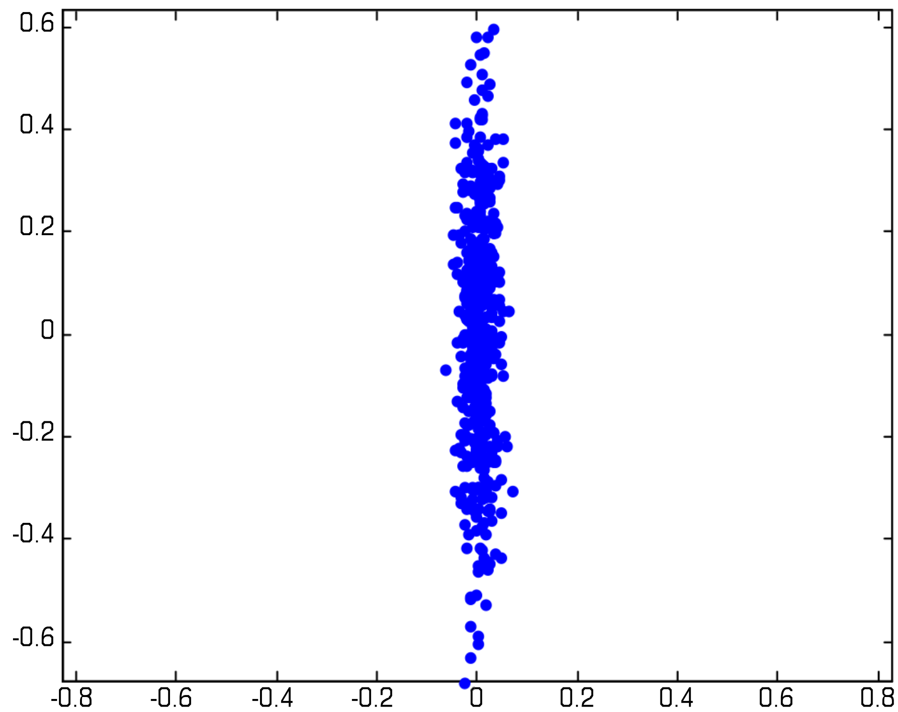
0.0004	0
0	0.0617

plot(r(:,1), r(:,2), '.');



Points in PCA space

```
[U, D]=eig(C)
% project points onto basis
coeffs = r*U;
plot(coeffs(:,1), coeffs(:,2), '.');
axis true
```



$$\tilde{x}^n = \bar{x}^n U$$

$\bar{x}^n$ zero mean $\qquad \mathrm{cov}(\tilde{x}) = D$

Mahalanobis distance

$$p(\vec{x}) = \frac{1}{(2\pi)^{D/2} |C|^{1/2}} \exp\left(-\frac{1}{2} (\vec{x} - \vec{\mu})^T C^{-1} (\vec{x} - \vec{\mu}) \right)$$

$$\widetilde{x} = \vec{x} - \vec{\mu}$$

$$d(\widetilde{x}) = \left(\widetilde{x}^T C^{-1} \widetilde{x} \right)$$

$$C = USU^T$$

Mahalanobis Distance

$$d(\tilde{x}) = \left(\tilde{x}^T C^{-1} \tilde{x}\right)$$

$$= \tilde{x}^T (USU^T)^{-1} \tilde{x}$$

$$= \tilde{x}^T U S^{-1} U^T \tilde{x}$$

$$= y^T S^{-1} y$$

$$= \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i}$$

$$\approx \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i}$$

Linear coefficients

$$y = U^T \tilde{x}$$

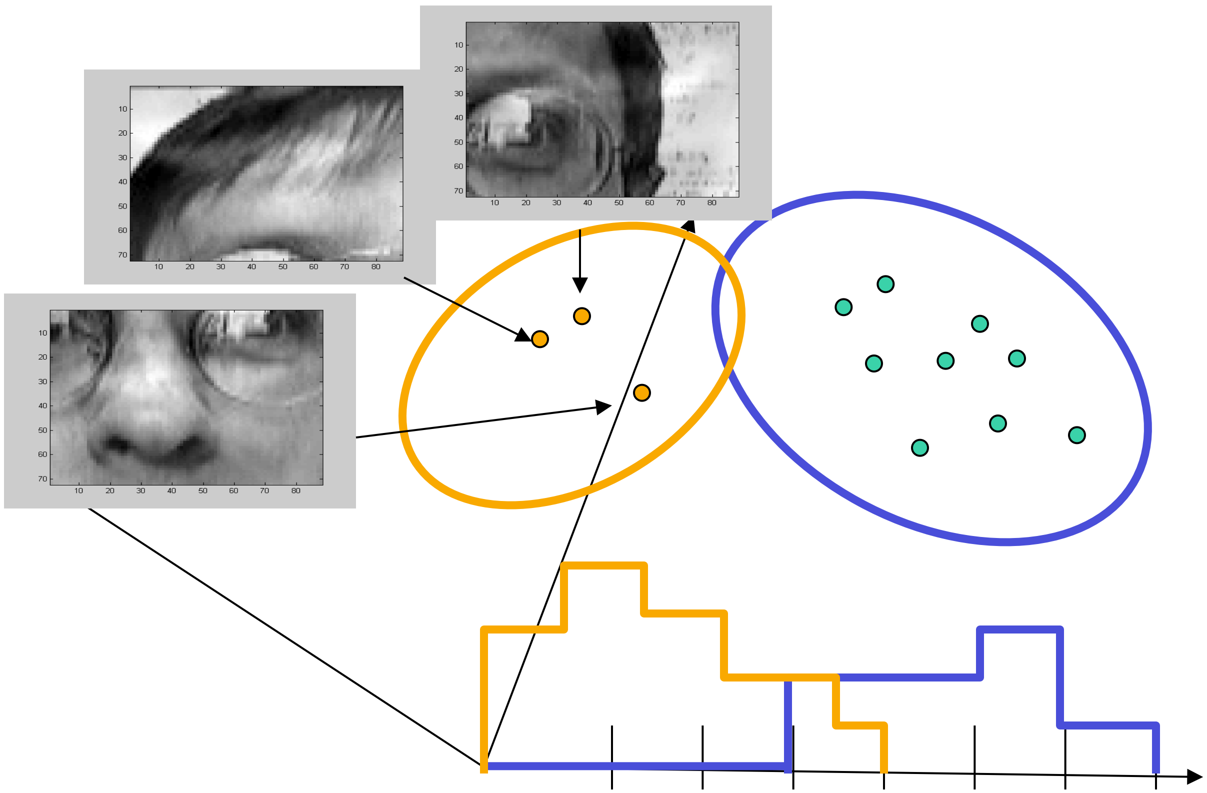
Mahalanobis distance

$$p(\vec{x}) = \frac{1}{(2\pi)^{D/2} | C |^{1/2}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu})^T C^{-1} (\vec{x} - \vec{\mu}) \right)$$

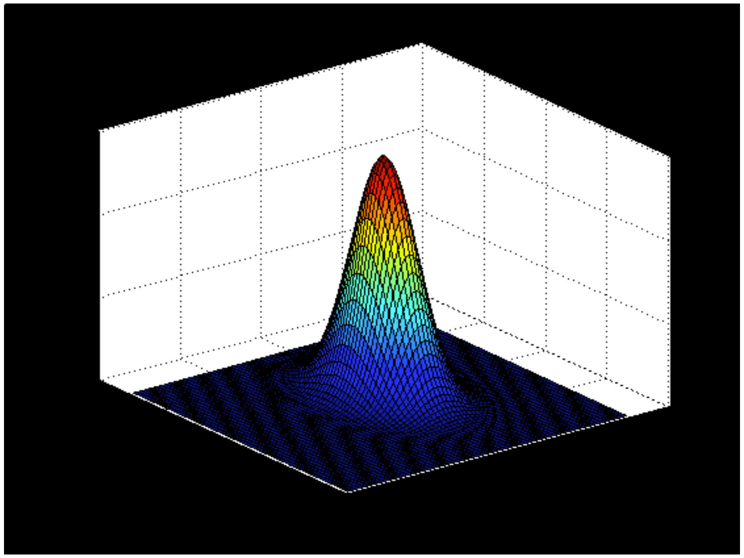
$$p(\vec{x}) = \frac{1}{(2\pi)^{D/2} | S |^{1/2}} \exp\left(-\frac{1}{2} y^T S^{-1} y \right)$$

$$p(\vec{x}) = \prod_{i=1}^{D} \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left(-\frac{1}{2} y_i^2 / \lambda_i \right)$$

Classification



Covariance

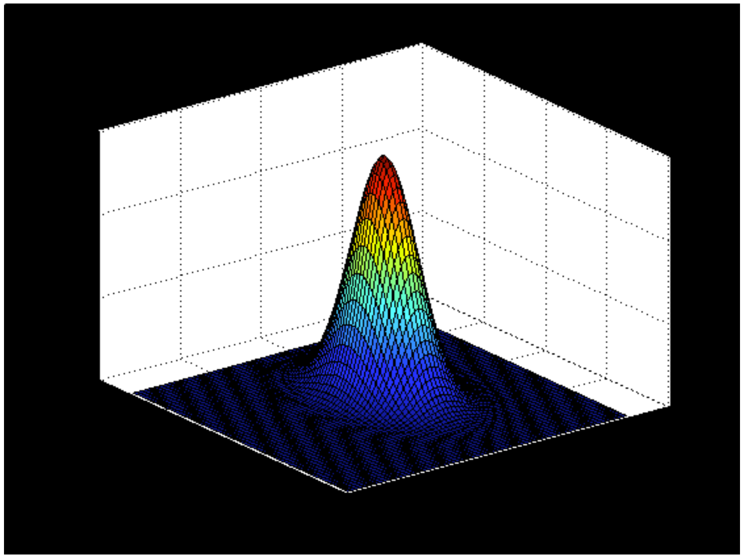


Multivariate Gaussian
(Normal)

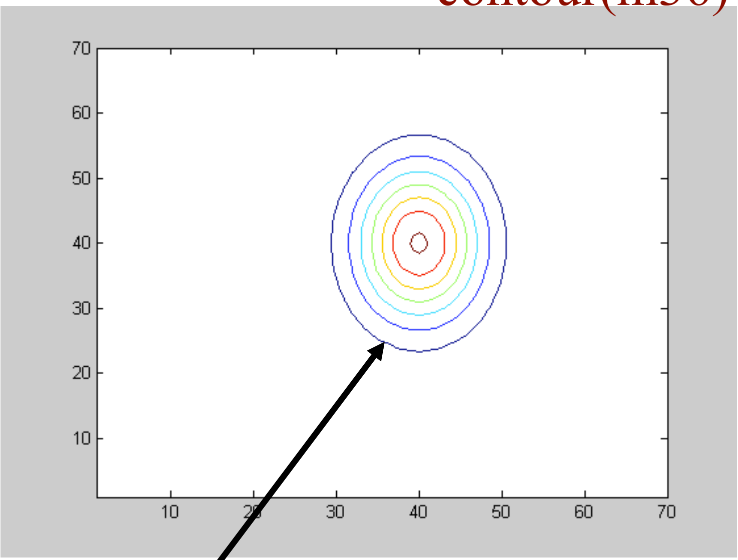
Mahalanobis distance Δ^2

$$p(\vec{x}) = \frac{1}{(2\pi)^{D/2} \, | C |^{1/2}} \exp\left(-\frac{1}{2} \underbrace{(\vec{x} - \vec{\mu})^T C^{-1} (\vec{x} - \vec{\mu})}_{} \right)$$

Covariance Ellipse



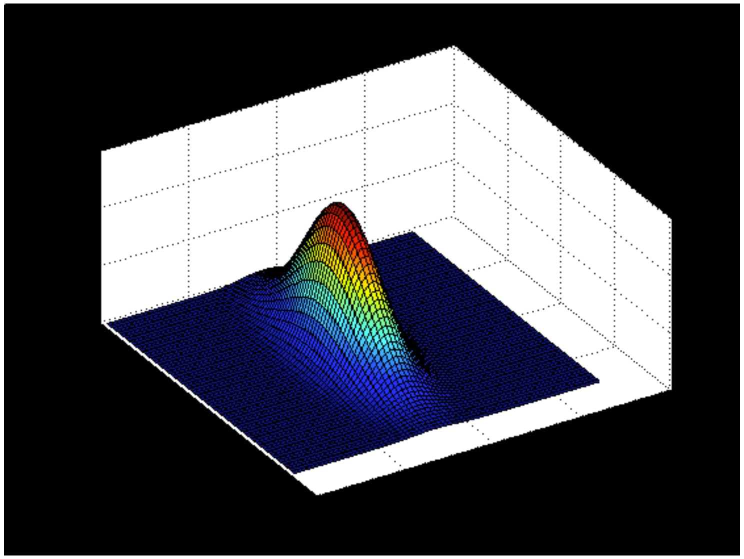
contour(m36)

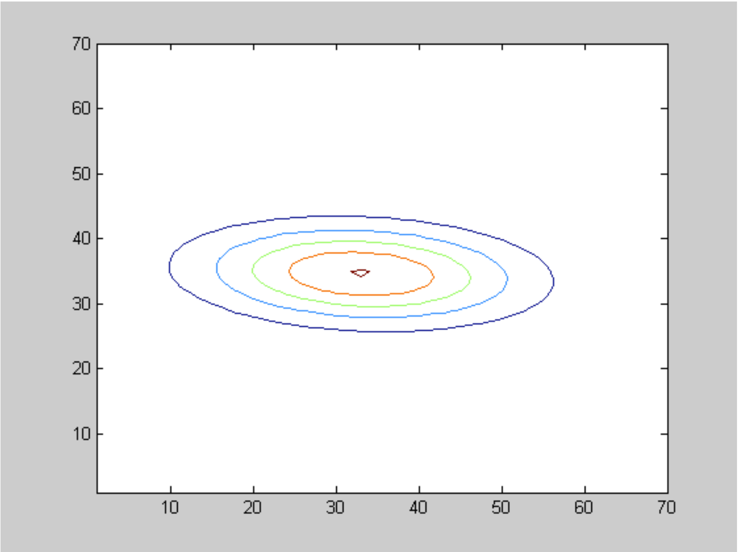


hyperellipsoids of constant Mahalanobis distance Δ^2

Note the ellipse is axis-aligned. Why?

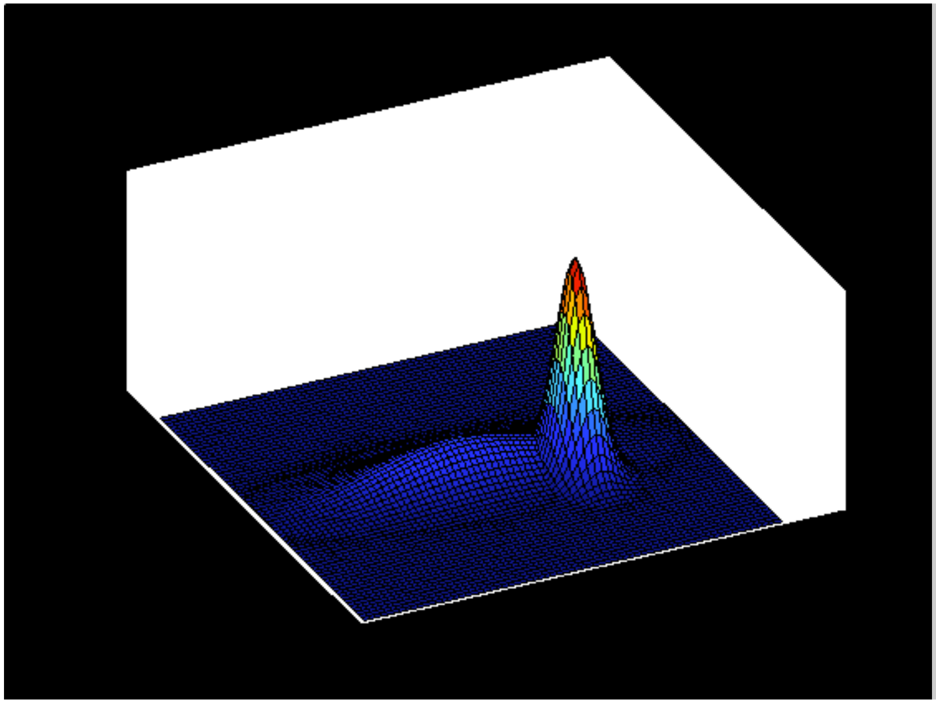
What about Not Mouths

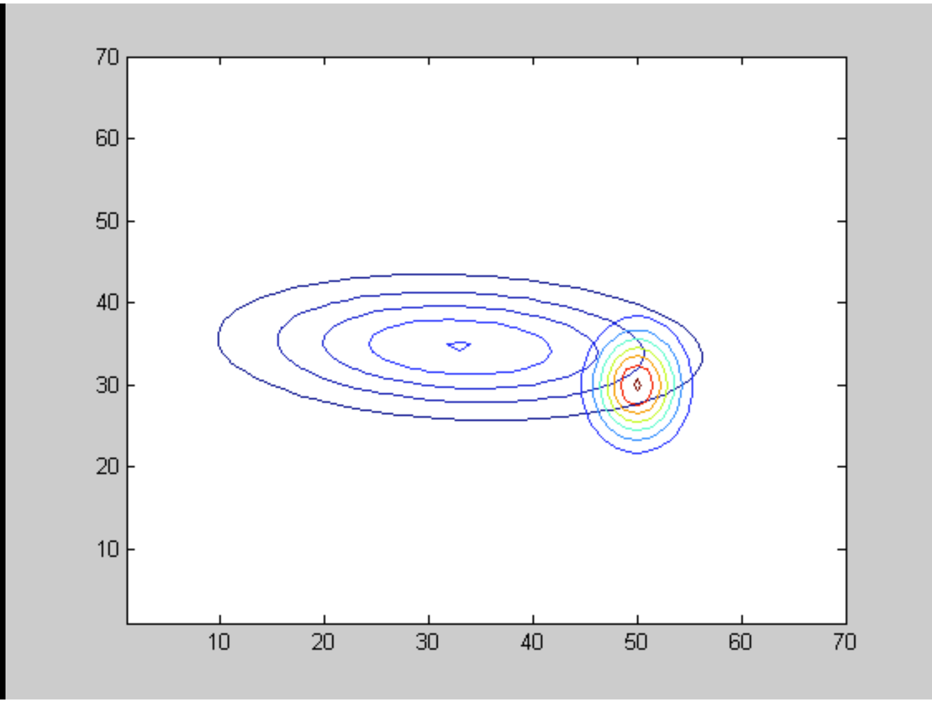




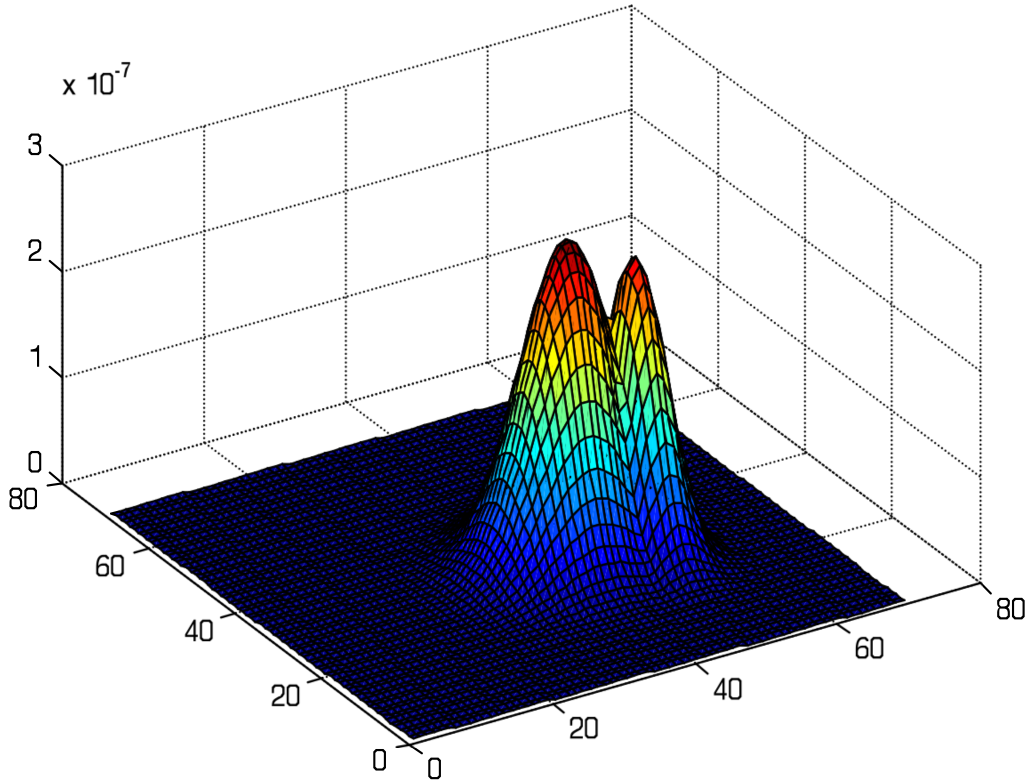
Note ellipse is not axis aligned. Why?

Plot them together





Posterior



Discriminant Function

Classify feature vector as a mouth if:

$$g_{\text{mouth}}(\vec{x}) > g_{\neg\text{mouth}}(\vec{x})$$

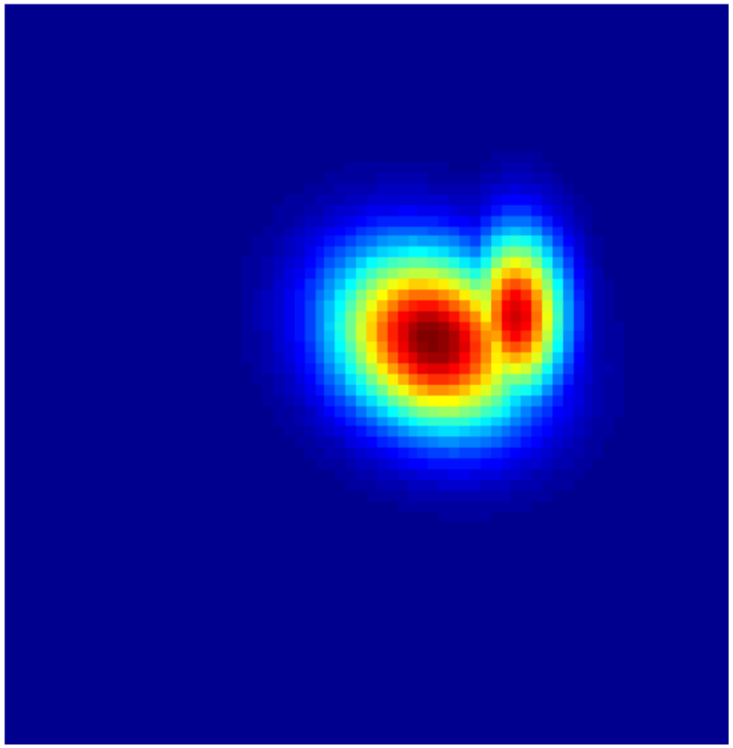
Take g to be the posterior

$$g_{\text{mouth}}(\vec{x}) = p(\text{mouth} \mid \vec{x}) \propto p(\vec{x} \mid \text{mouth})\, p(\text{mouth})$$

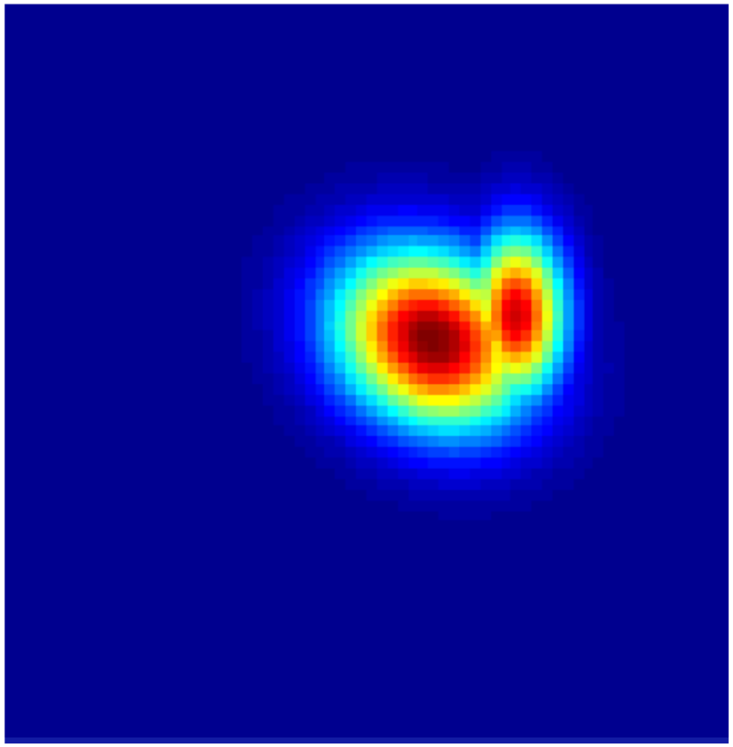
Holds true for monotonic functions of g (e.g. log). Define

$$g_{\text{mouth}}(\vec{x}) = \log(p(\vec{x} \mid \text{mouth})) + \log(p(\text{mouth}))$$

Decision boundary

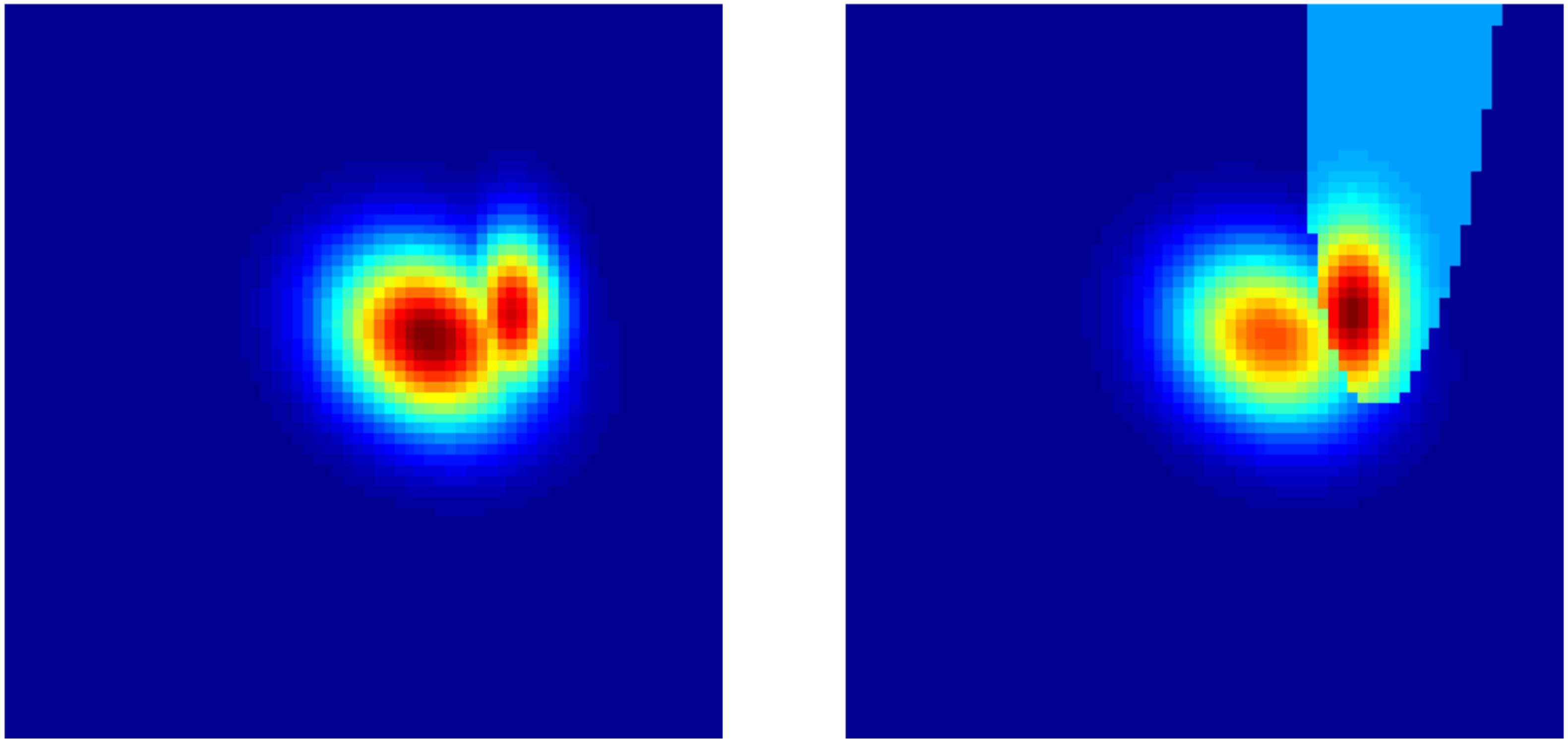


Decision boundary



In higher dimensions we should do even better.

Decision boundary



In higher dimensions we should do even better.

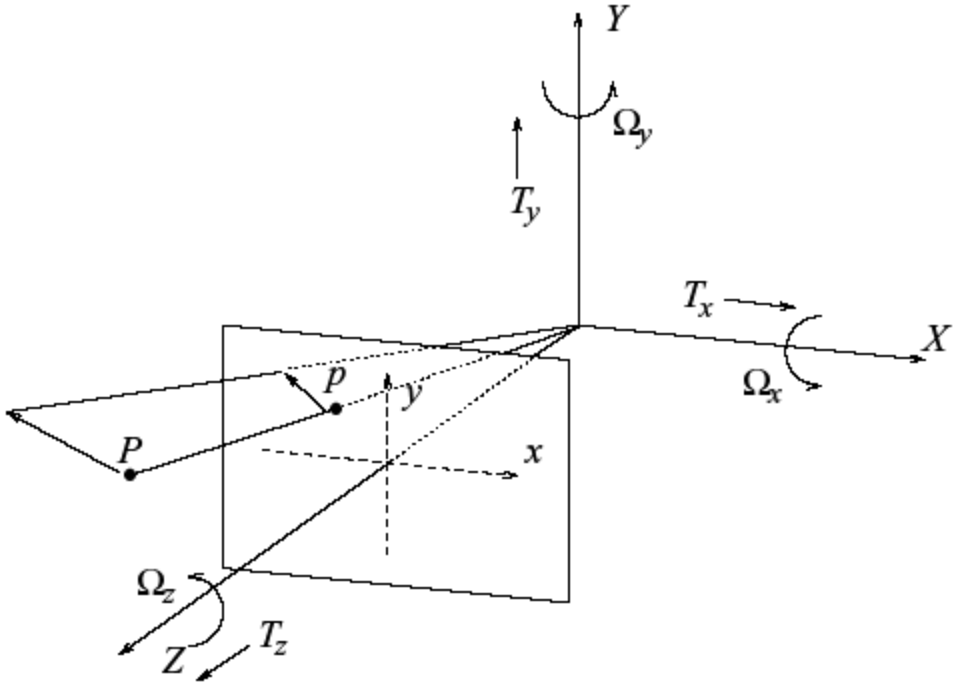
Goal: Introduce Motion

- So far we've looked at static images.
- The world is more complex and interesting.
- We need to understand movment.

Optical Flow

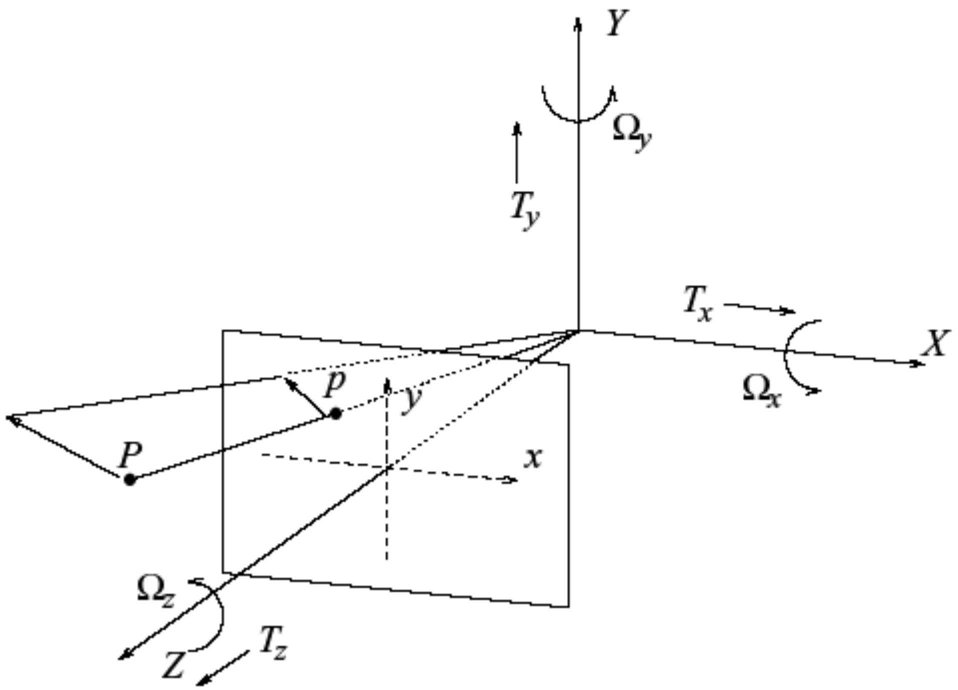
J. J. Gibson, The Ecological Approach to Visual Perception

Motion Field



Motion field = 2D motion field representing the projection of the 3D motion of points in the scene onto the image plane.

Optical Flow

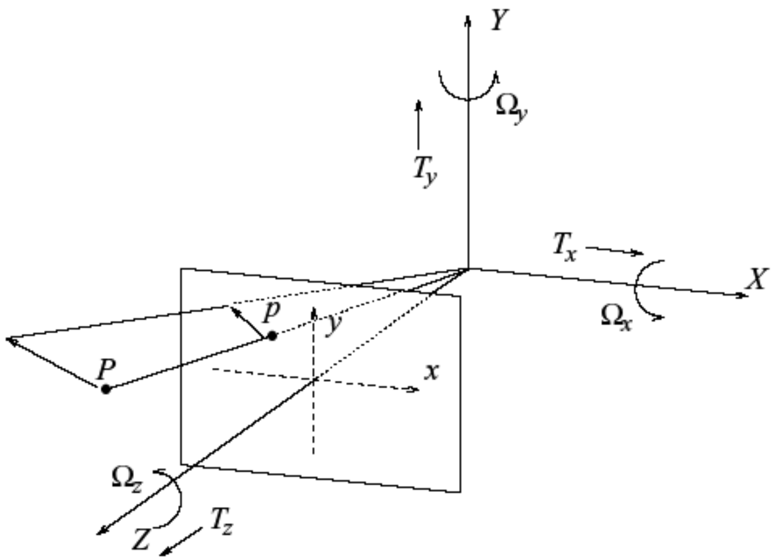


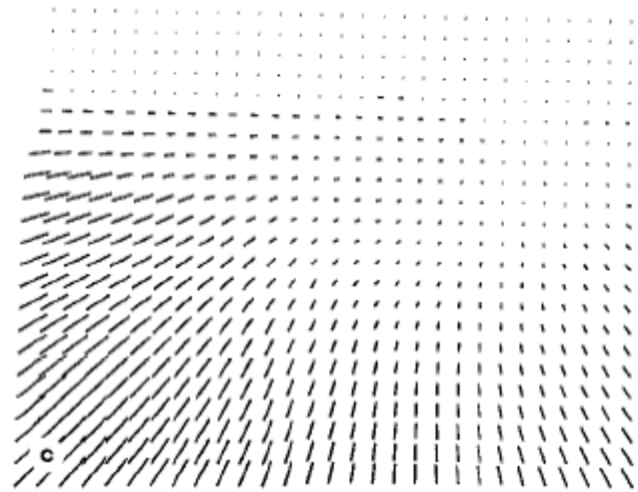
Optical flow = 2D velocity field describing the apparent motion in the images.

Optical Flow Field

Image irradiance at time *t* and location **x**=(*x*, *y*)

$$I(x, y, t)$$

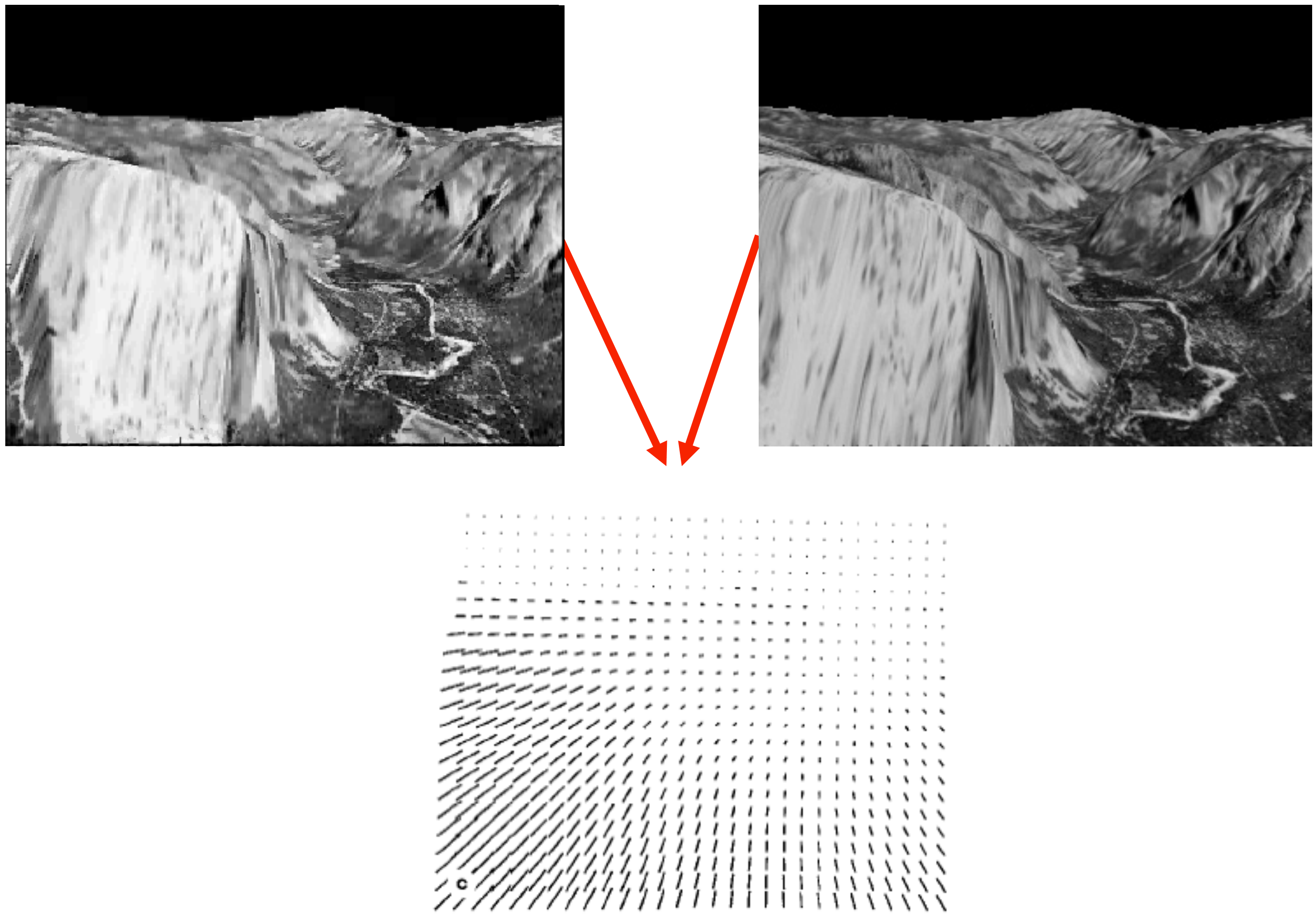




$u(x, y)$ Horizontal component

$v(x, y)$ Vertical component

Problem



Thought Experiment 1

Lambertian (matte) ball rotating in 3D

What does the 2D motion field look like?

What does the 2D optical flow field look like?

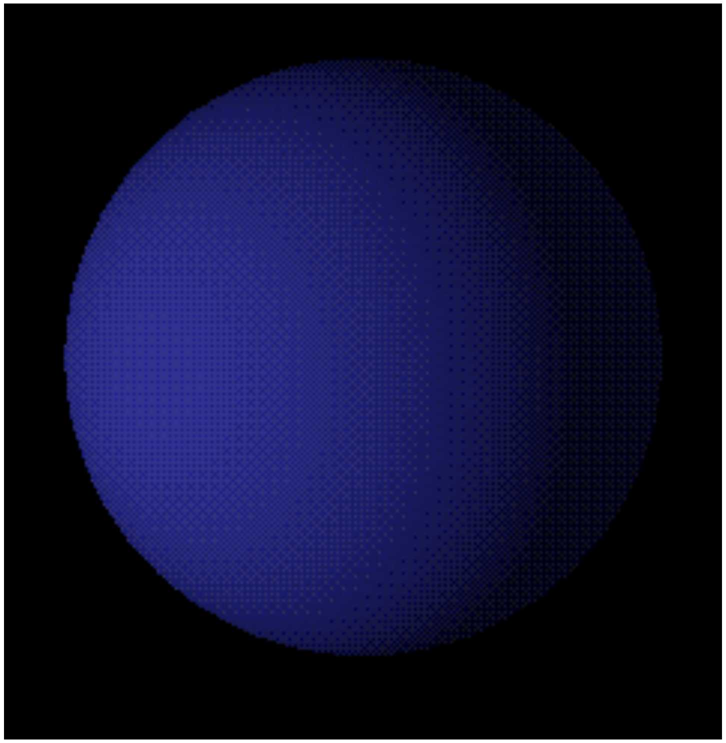
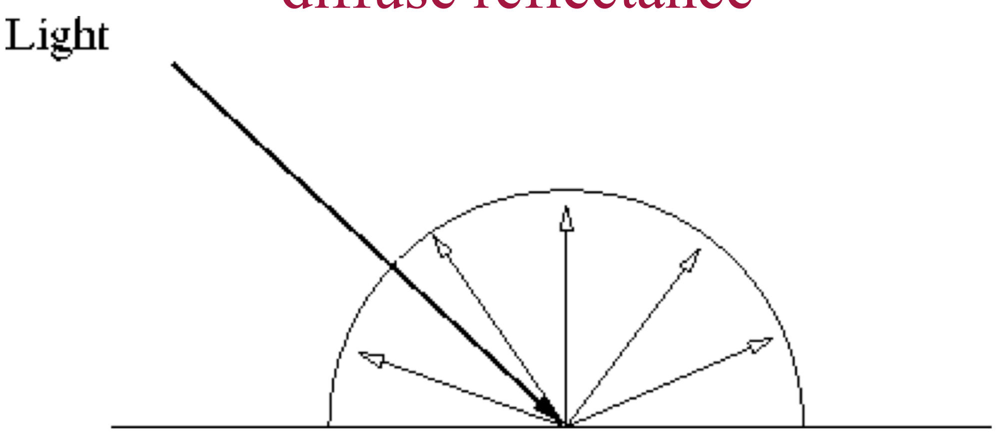


Image source: http://www.evl.uic.edu/aej/488/lecture12.html

Special Cases: Lambertian

Perfect matte surface

* reflects all light
* reflects equally in all directions
* patch appears equally bright from all viewing directions
* diffuse reflectance



Thought Experiment 2

Stationary Lambertian (matte) ball, moving light source.

What does the 2D motion field look like?

What does the 2D optical flow field look like?

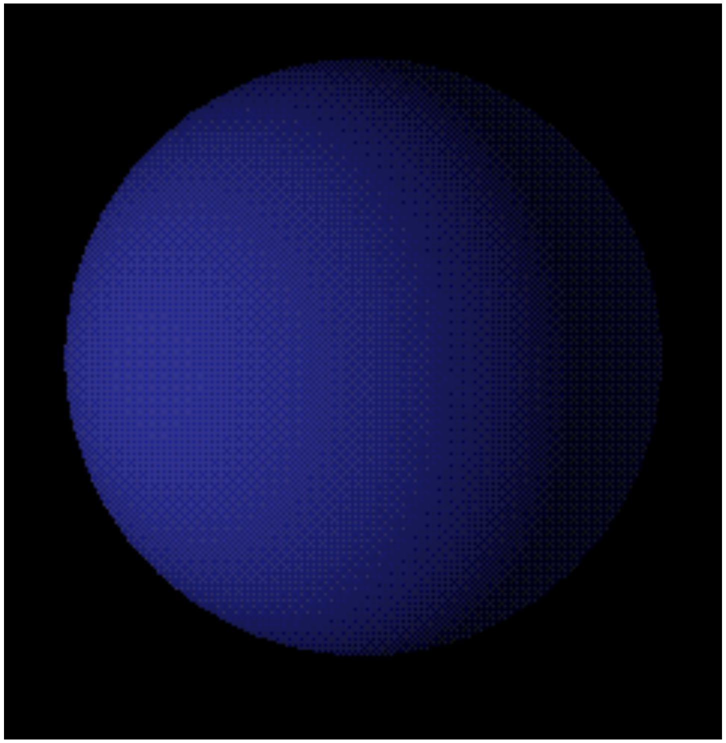


Image source: http://www.evl.uic.edu/aej/488/lecture12.html