Introduction to Computer Vision

Michael J. Black

Oct 2009

Motion estimation

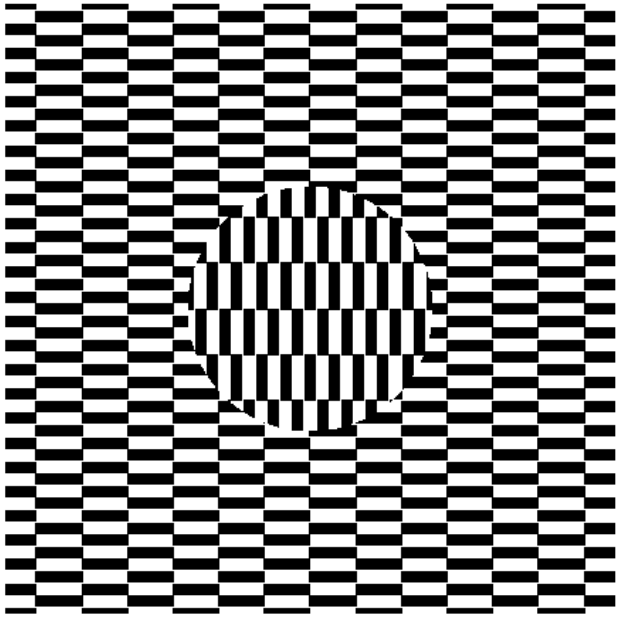
Goals

- Motion estimation
  - Beyond translation
  - Optimization
  - Large motions
- Friday
  - Motion under perspective.
  - dense, smooth motion and regularization. Robust statistics
- Monday – discuss projects.

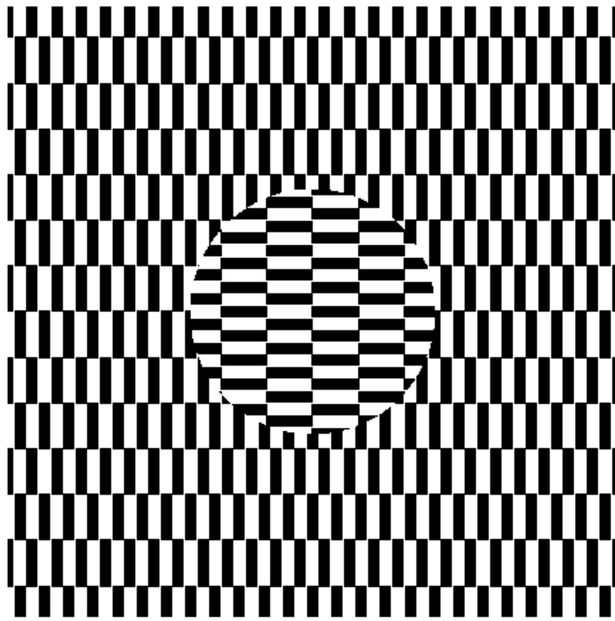
Assignment 3

- Out right after class.

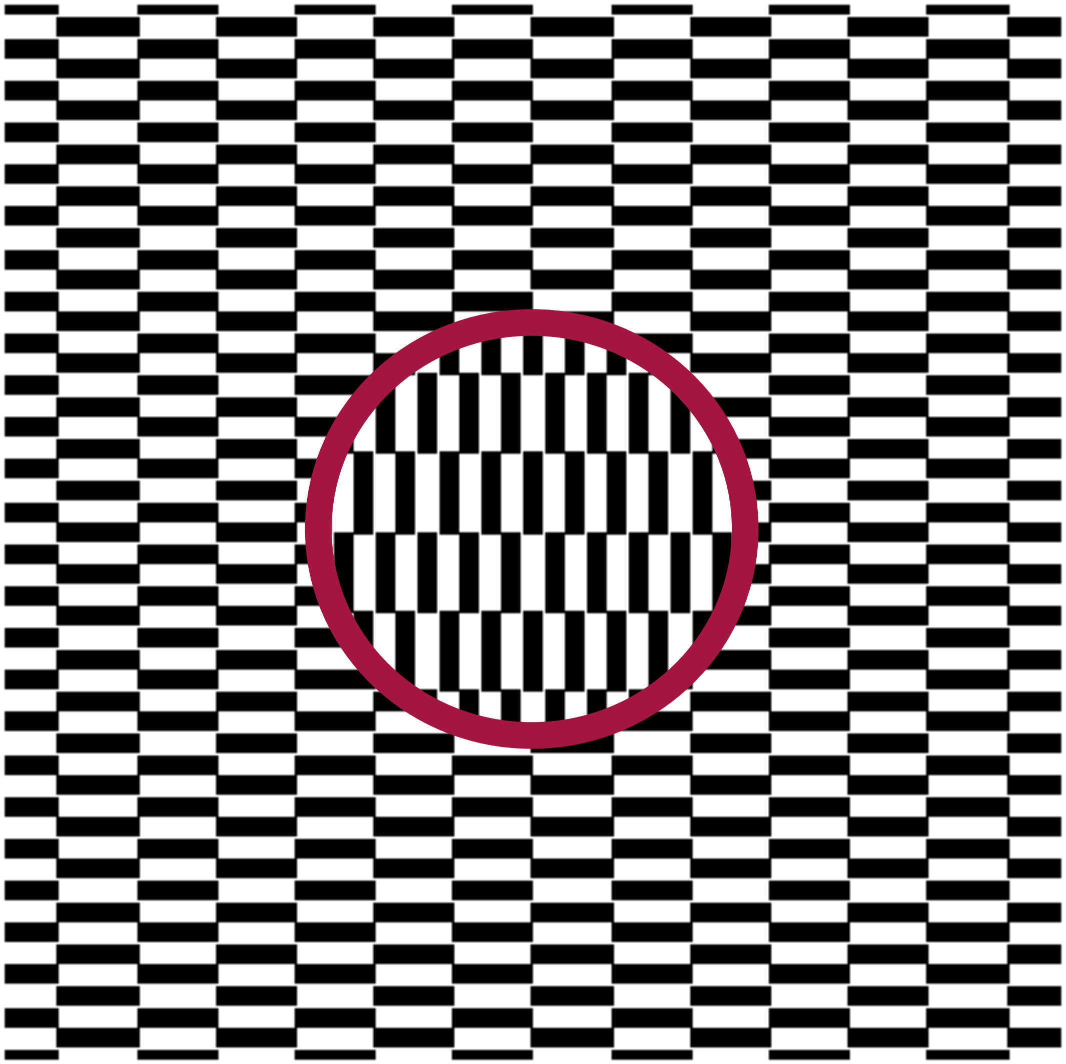
Ouchi Illusion

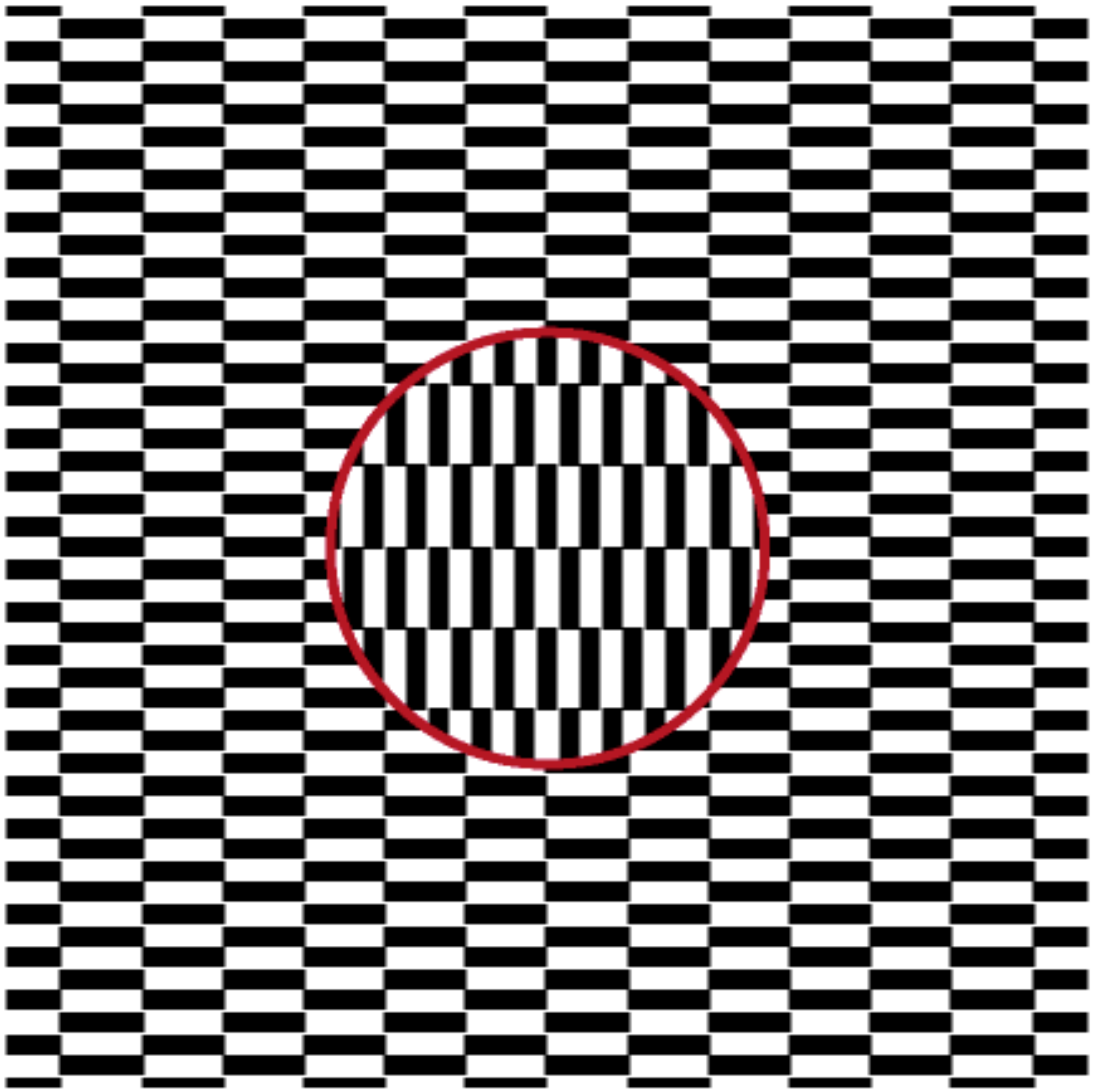


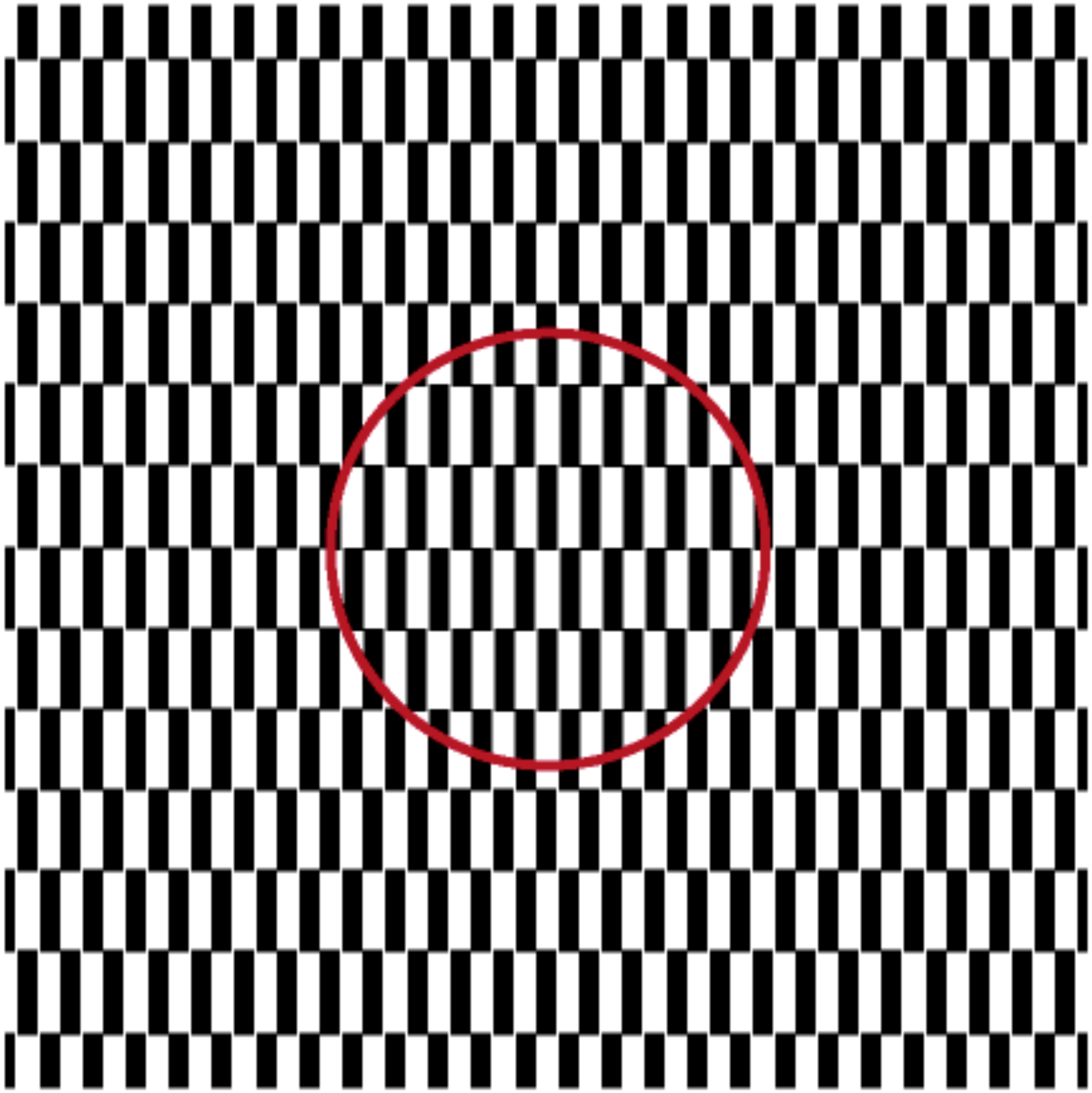
Ouchi Illusion

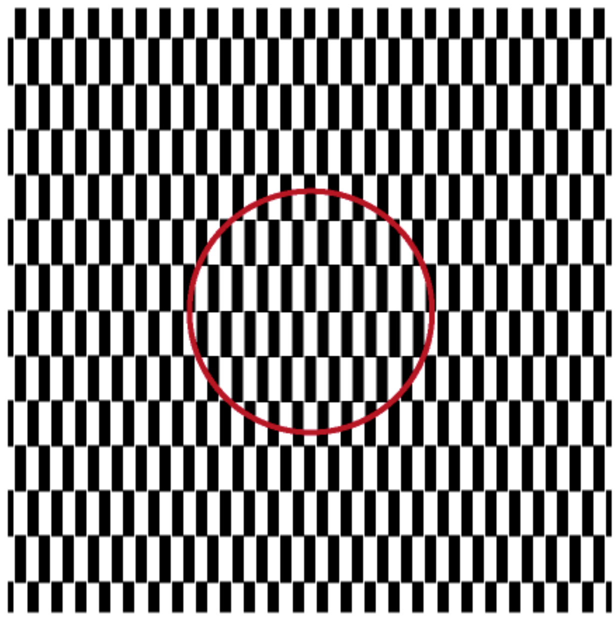


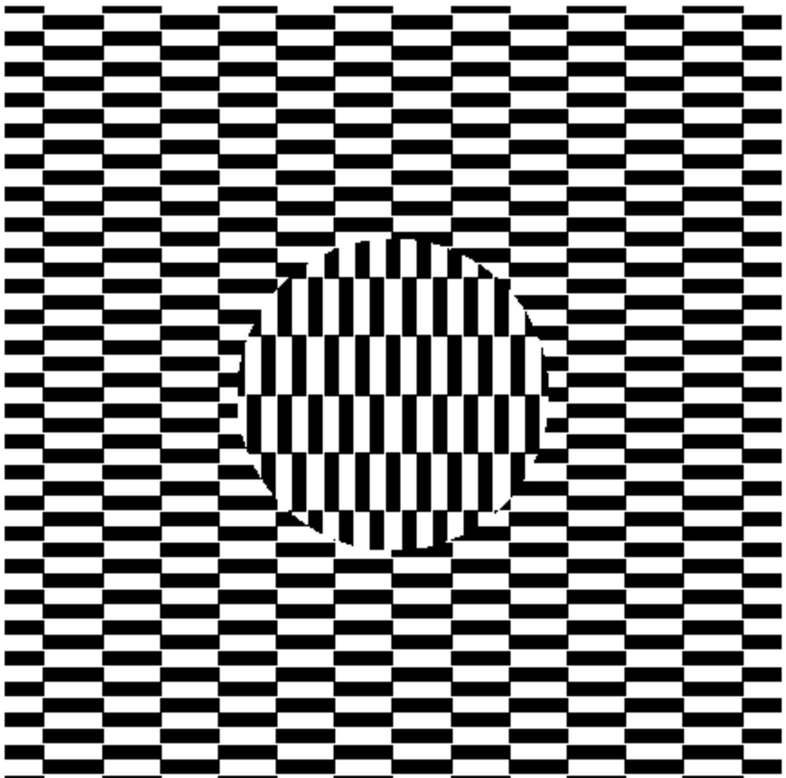
Ouichi Illusion

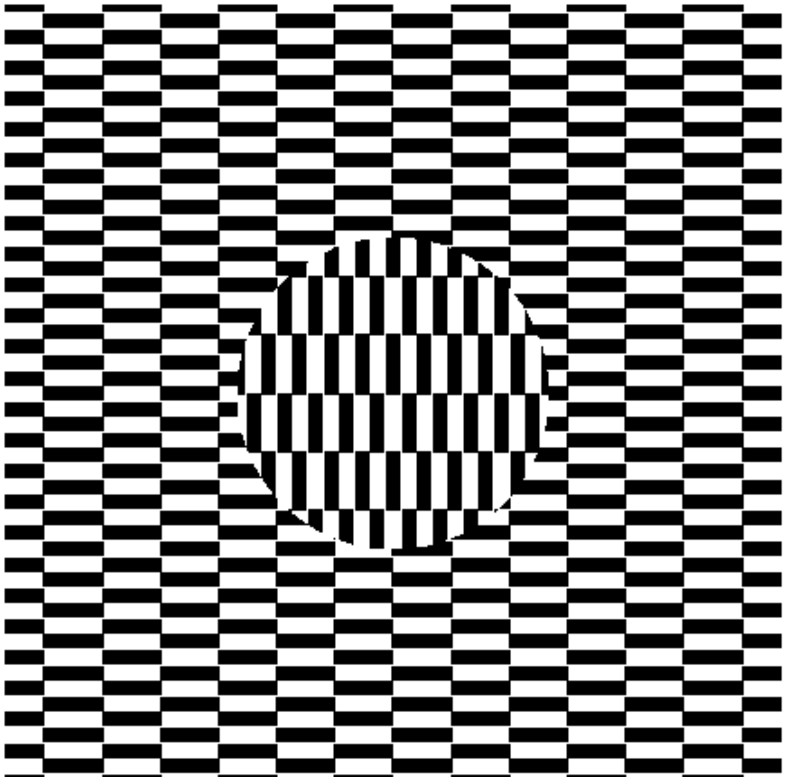












Optimization

$$E(u,v) = \sum_{x,y \in R} (I_x(x,y,t)u + I_y(x,y,t)v + I_t(x,y,t))^2$$

Differentiate with respect to *u* and *v* and set this to zero, rearrange.

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

Optimization: solve for u,v

System of 2 equations in 2 unknowns:

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

$$\mathbf{A}\mathbf{u} = \mathbf{b}$$
$$\mathbf{A}^T\mathbf{A}\mathbf{u} = \mathbf{A}^T\mathbf{b}$$
$$\mathbf{u} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

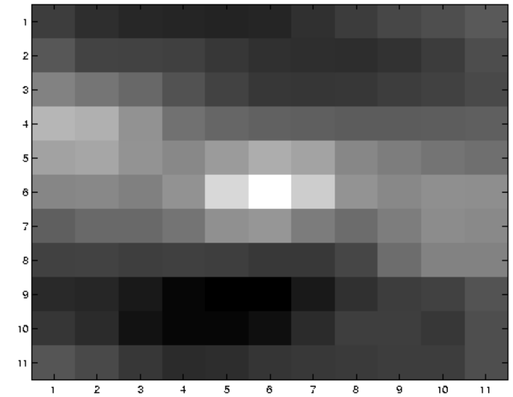
Assumptions?

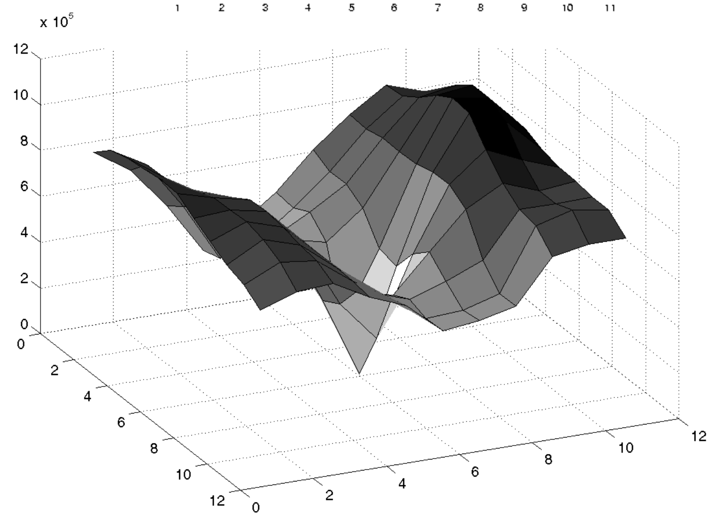
What assumptions are we making for this to work?

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

$$\mathbf{Au} = \mathbf{b}$$
$$\mathbf{A}^T\mathbf{Au} = \mathbf{A}^T\mathbf{b}$$
$$\mathbf{u} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

SSD Surface – Textured area

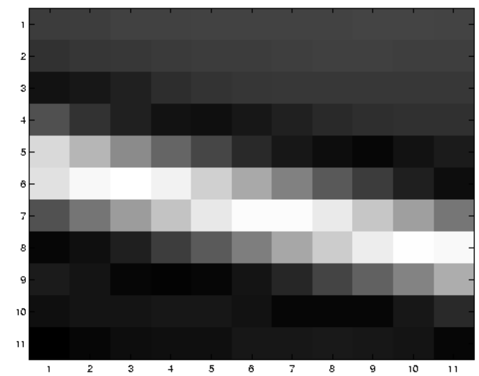


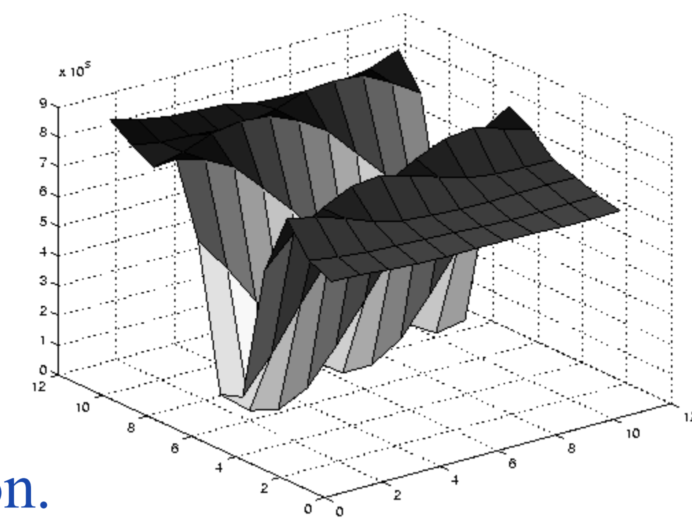


$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix}$$

Gradients in *x* and *y*.

SSD Surface -- Edge

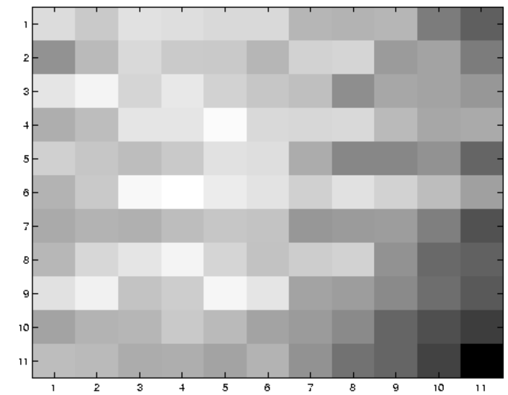


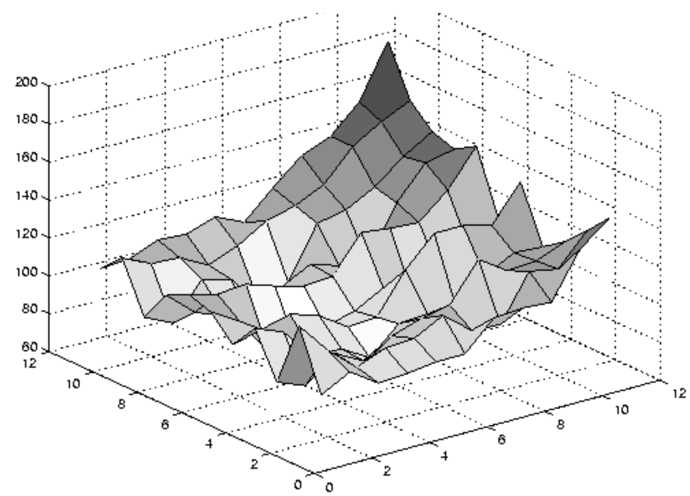


$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix}$$

Gradients oriented in one direction.

SSD Surface – homogeneous area





$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix}$$

Weak gradients everywhere.

SSD Surface – Surface Boundary

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix}$$

Gradients in x and y.

?

Translational Model

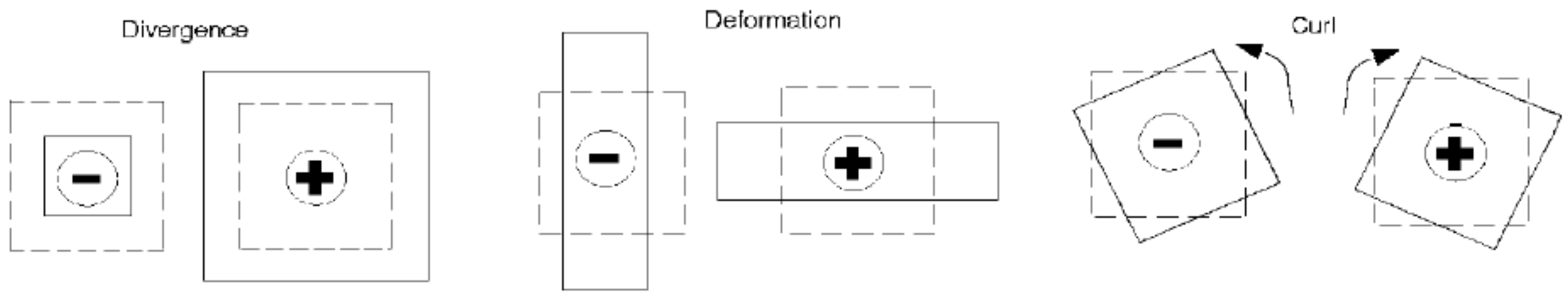
What's wrong with the translational assumption (ie constant motion within a region R)?

How can we generalize it?

Affine Flow

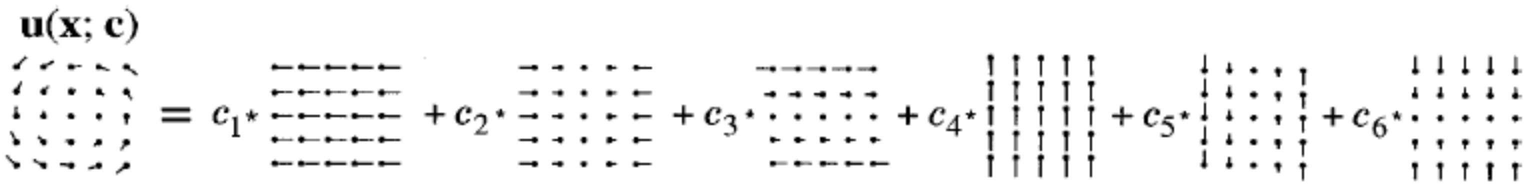
$$E(\mathbf{a}) = \sum_{x,y \in R} (\nabla I^T \mathbf{u}(\mathbf{x};\mathbf{a}) + I_t)^2$$

$$\mathbf{u}(\mathbf{x};\ \mathbf{a}) = \begin{bmatrix} u(\mathbf{x};\mathbf{a}) \\ v(\mathbf{x};\mathbf{a}) \end{bmatrix} = \begin{bmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{bmatrix}$$



Linear Basis

You can think of this as just another set of linear basis functions!



$$\mathbf{u}(\mathbf{x};\, \mathbf{c}) = \sum_{j=1}^{n} a_j \mathbf{b}_j(\mathbf{x})$$

Aside: we can learn these with PCA from examples.
See Hager and Belhemeu

Affine Transformation

$$\begin{bmatrix} x \\ y \end{bmatrix}^* = \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_3 \\ a_6 \end{bmatrix}$$

Linear transformation

Translation

Affine Transformation

$$\begin{bmatrix} x \\ y \end{bmatrix}^* = \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_3 \\ a_6 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}^* = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous coordinates

Linear transformation

Translation

Affine flow

Motion (flow) between frames (matrix form)

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a1 & a2 & a3 \\ a4 & a5 & a6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Affine flow

Motion (flow) between frames

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a1 & a2 & a3 \\ a4 & a5 & a6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Transformation of pixels

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} + \begin{bmatrix} a1 & a2 & a3 \\ a4 & a5 & a6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Use when transforming pixels

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a1+1 & a2 & a3 \\ a4 & a5+1 & a6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$