Computational photography

MIT 6.098, 6.882

Bill Freeman, Fredo Durand

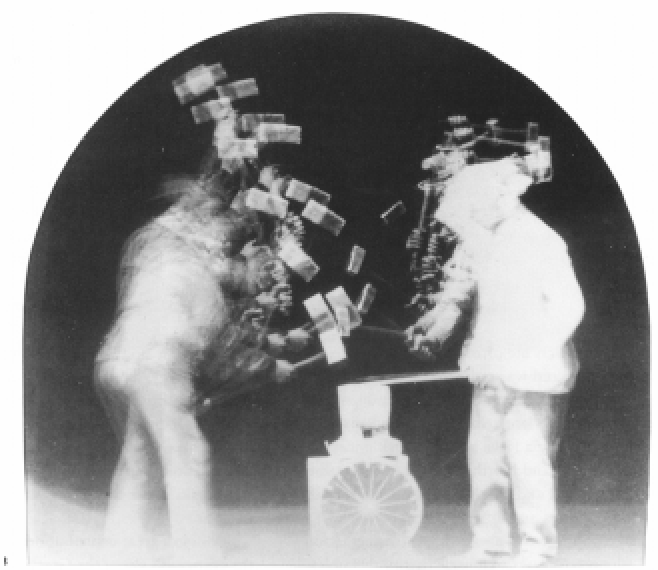
- Finish digital forensics
- Analyzing multiple images
  - Shapetime photography
  - Image stacks
- Analysing and synthesizing motion sequences
  - Motion without movement
  - Motion magnification

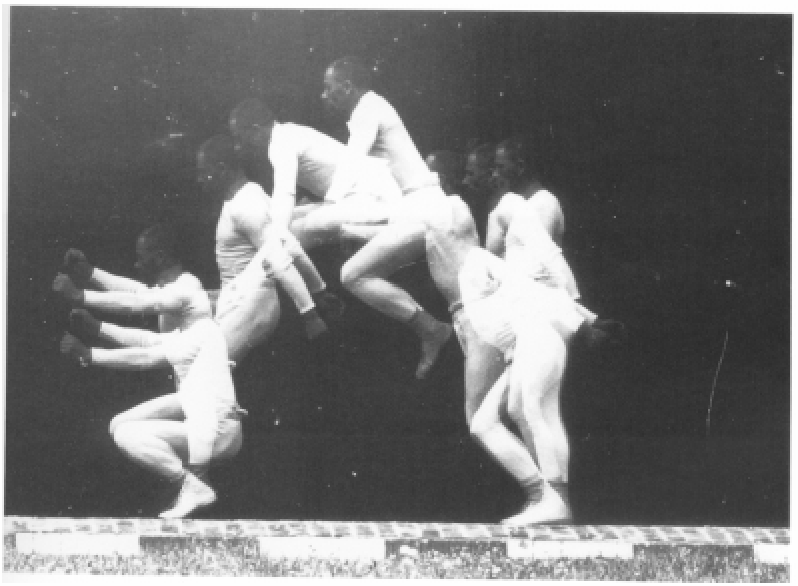
April 20, 2006

Analyzing multiple images

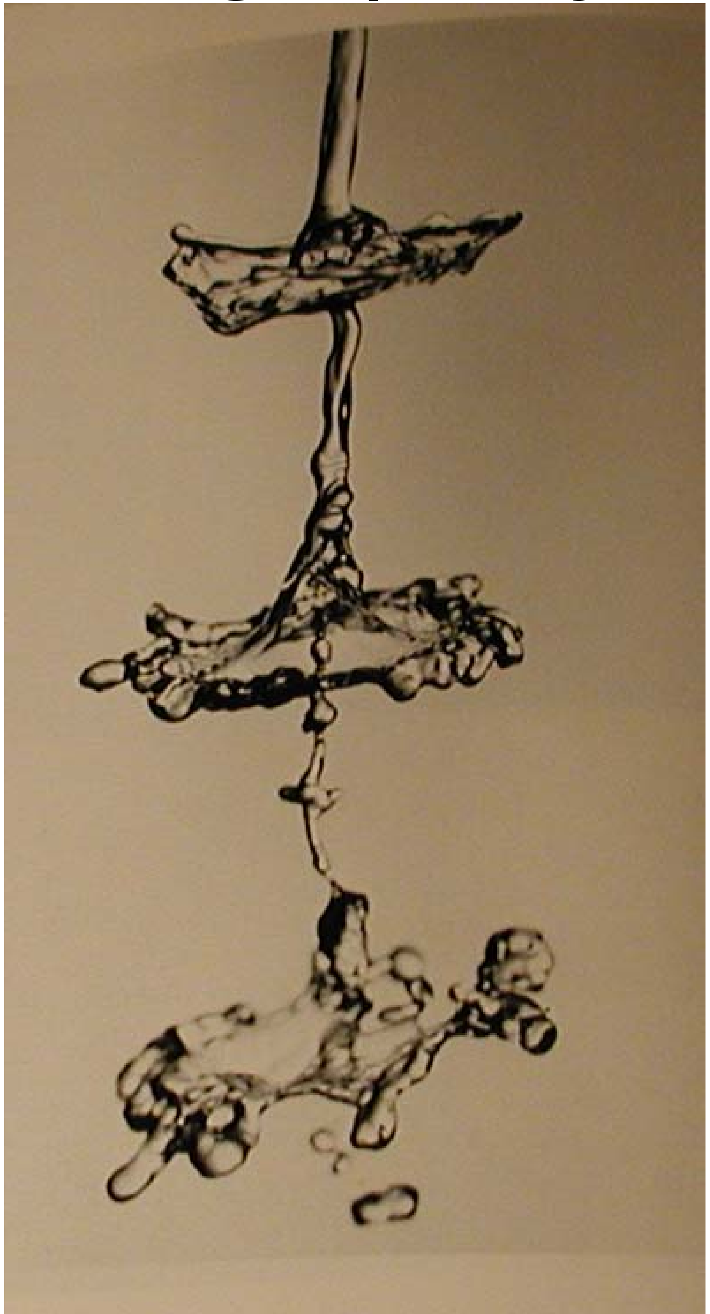
Bill Freeman
Fredo Durand
MIT Computational Photography, 6.882
April 20, 2006

Multiple-exposure images by Marey

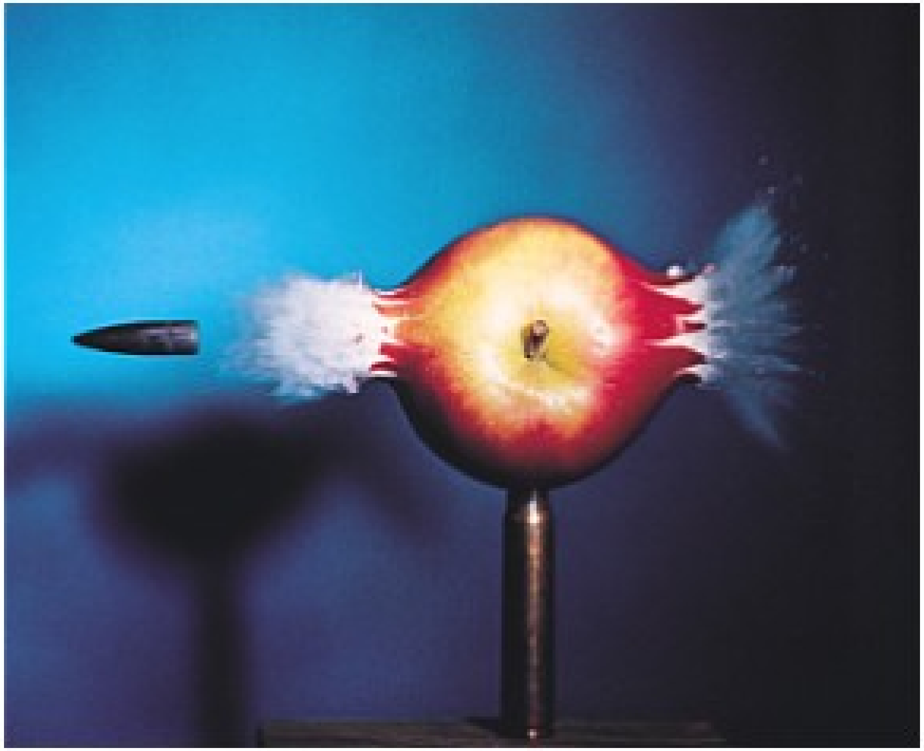


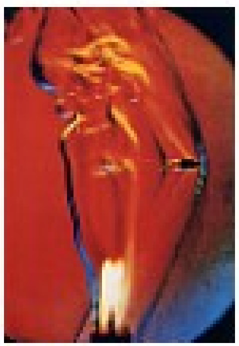


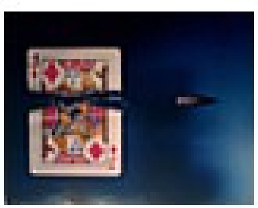
Strobe photograph by Edgerton

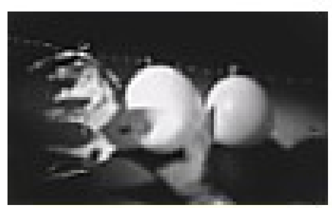


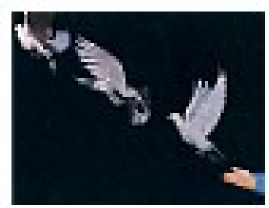
Other photographs by Doc Edgerton











What hardware was needed to make these photographs?

Strobe light, capacitor, thyristor…

Computational photography

Surely we can update those photographic techniques, adding the generality and flexibility of digital methods. Analyze and re-render the images.

Computational photography

Fredo and Bill describing computational photography:

- Fredo: using computation to make better quality photographs—**to enhance.**
- Bill: using computation to reveal things about the world that we otherwise couldn’t see—**to reveal**.

How display a single-frame summary of multiple frames?


im_9353.JPG

im_9352.JPG

im_9351.JPG

im_9350.JPG

im_9349.JPG

im_9348.JPG

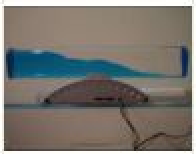
im_9347.JPG

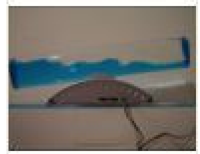
im_9346.JPG

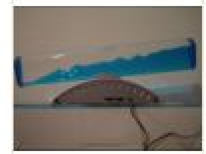
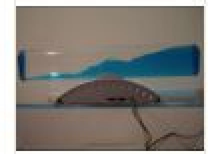
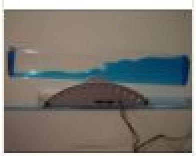
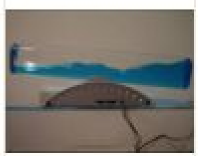
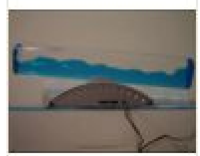
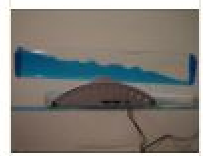
im_9345.JPG

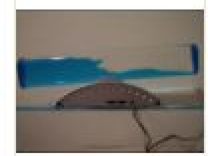
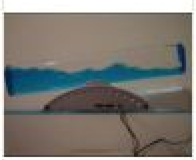
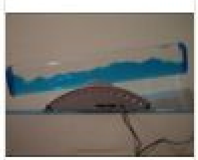
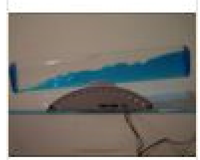
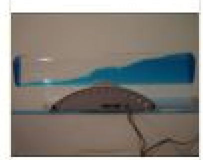
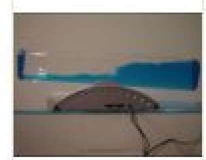
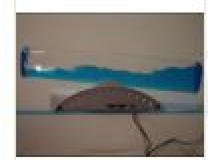
im_9344.JPG

im_9343.JPG

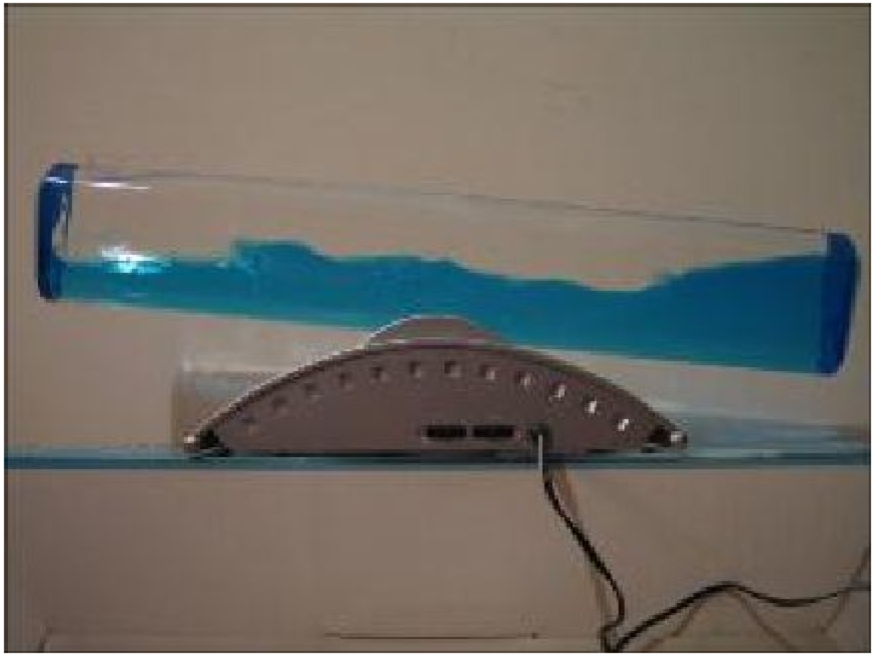
im_9342.JPG

im_9341.JPG

im_9340.JPG

im_9339.JPG

im_9338.JPG

im_9337.JPG

im_9336.JPG

im_9335.JPG

im_9334.JPG

im_9333.JPG

im_9332.JPG

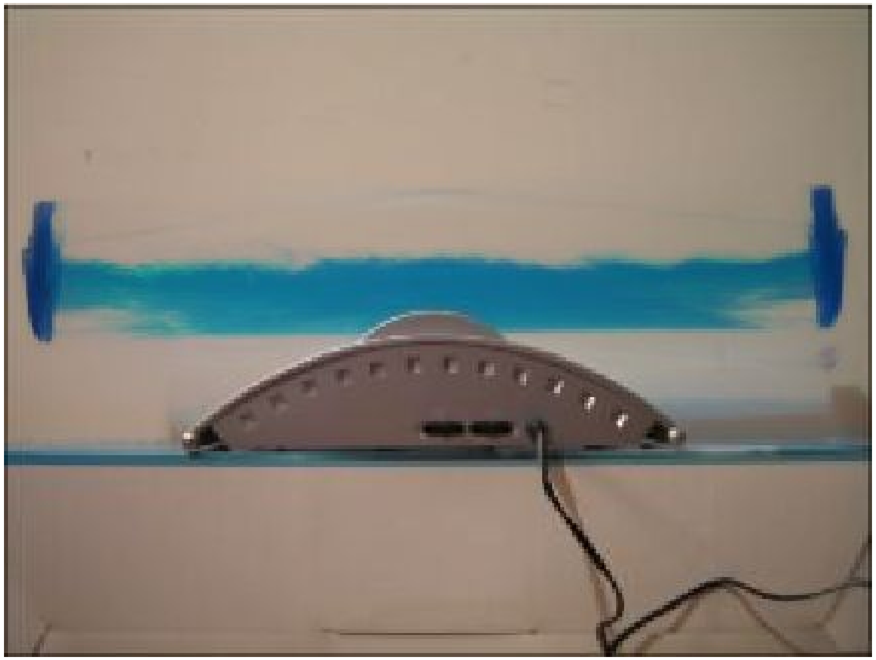
im_9331.JPG

im_9330.JPG

im_9329.JPG

im_9328.JPG

im_9327.JPG

im_9326.JPG

im_9325.JPG

im_9324.JPG

Typical frame

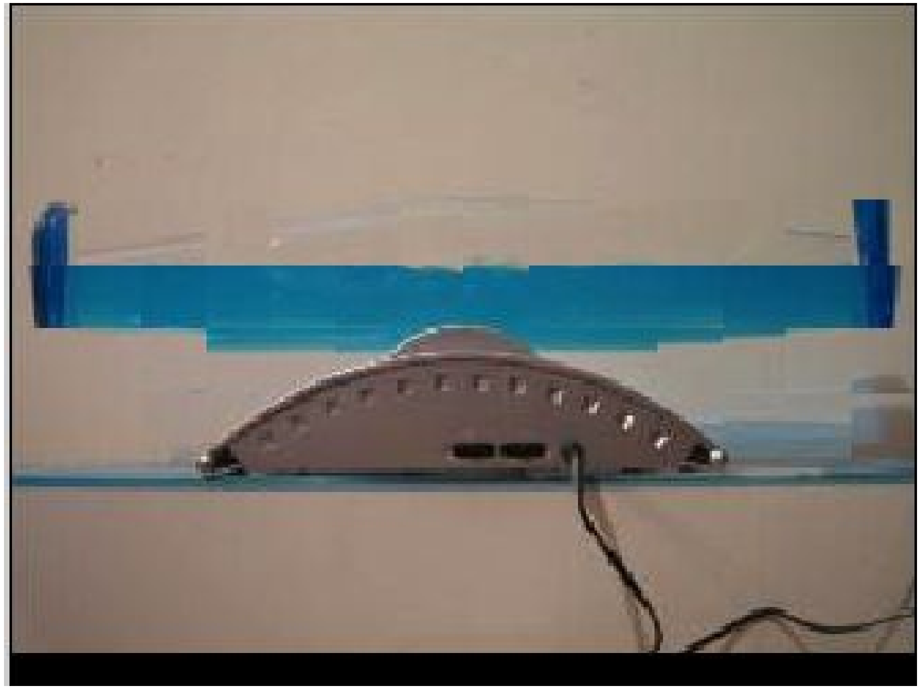


Average over 50 frames

Median filter over time



Vector median filter (20x20 patchs)



2x2 vector median

2x2 vector least median

Shapetime photography

Joint work with Hao Zhang, U.C. Berkeley

2002

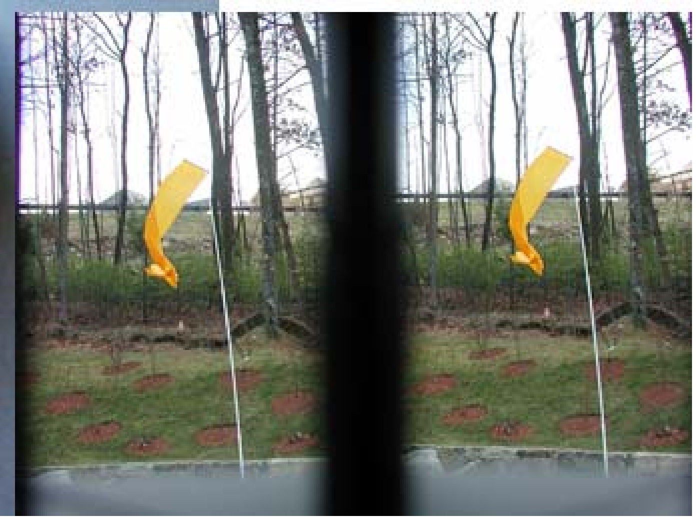
Video frames

Multiple-exposure

Layer-By-Time

Shape-Time


PENTAX
STEREO ADAPTER
OLYMPUS



Frame index of each displayed pixel

Resulting composite image

Frame index of each displayed pixel

Resulting composite image

With edge-preserving regularization

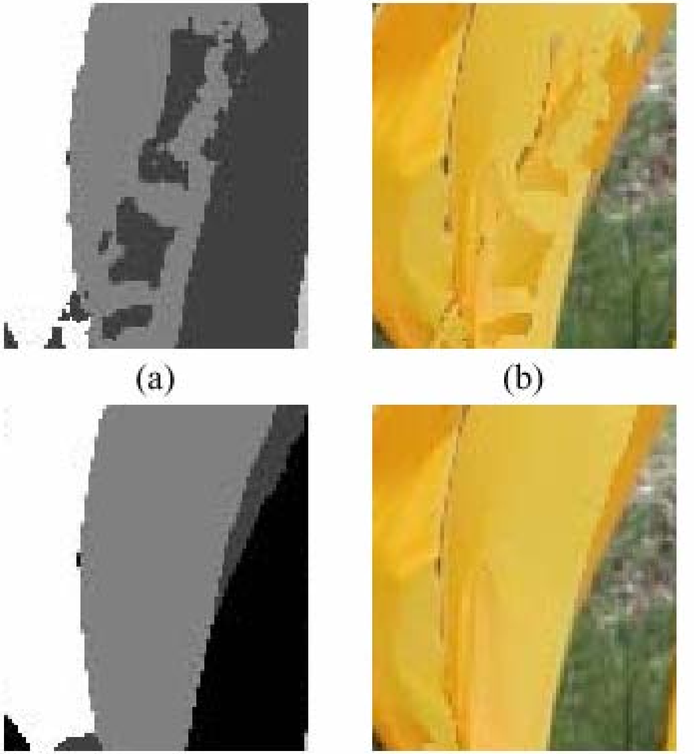


Figure 4: **(a) Layer assignments, without MRF processing. (b) Shape-time image based on those assignments. (c) Most probable layer assignments, computed by MRF. (d) Resulting shape-time image.**

number of pixels in the image.

$$P(\vec{t}) = \frac{1}{Z} \prod_{(jk)} \phi_{jk}(t_k, t_j) \prod_{k} \psi_k(t_k), \qquad (2)$$

(a)

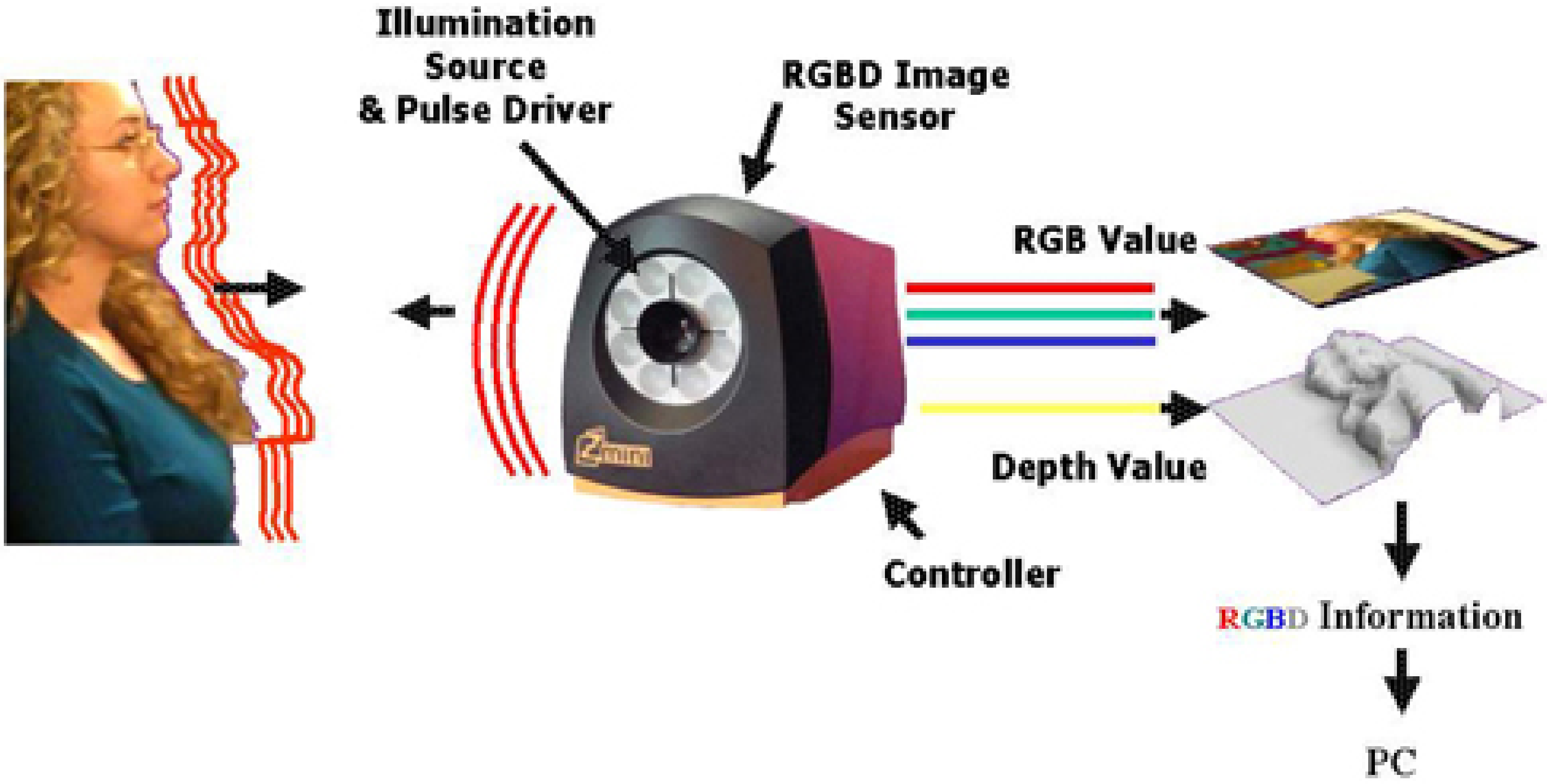
“how to sew”

Input sequence

Shape-Time composite

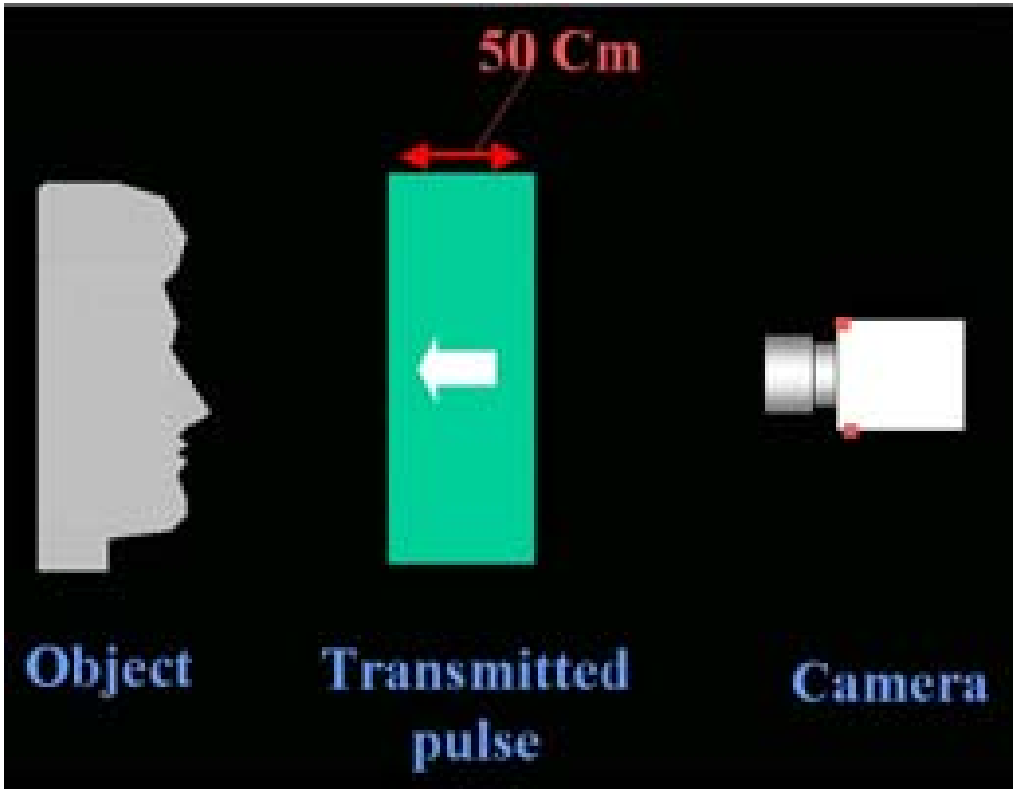
“inside-out”

Z-cam, made by 3DV



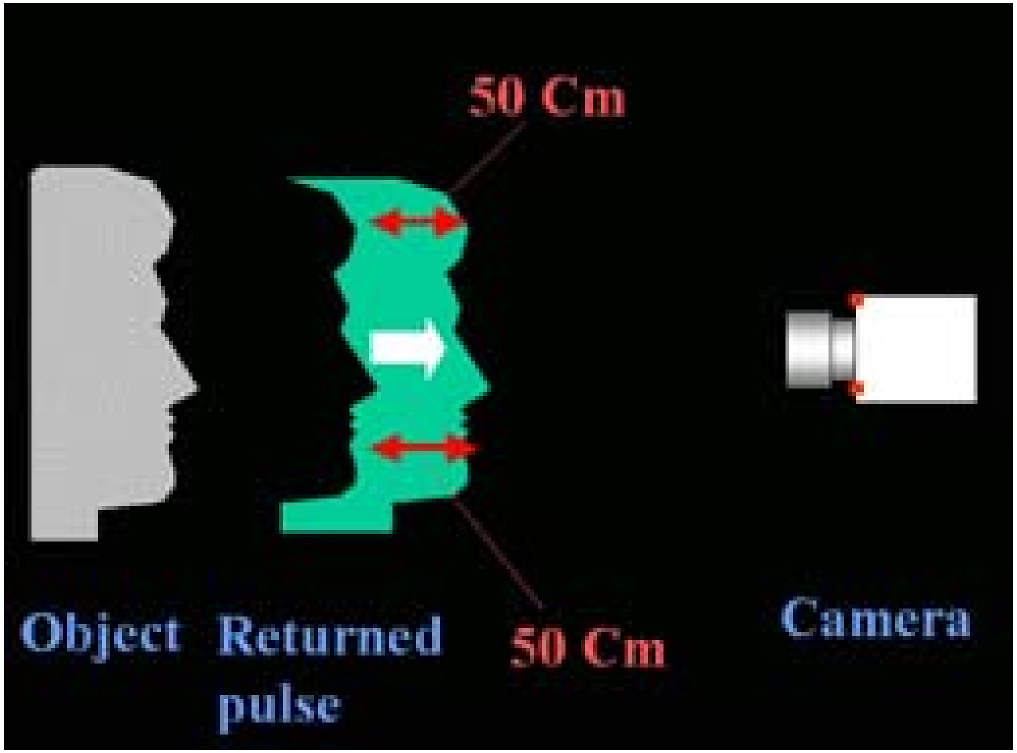
http://www.3dvsystems.com

3DV camera operation



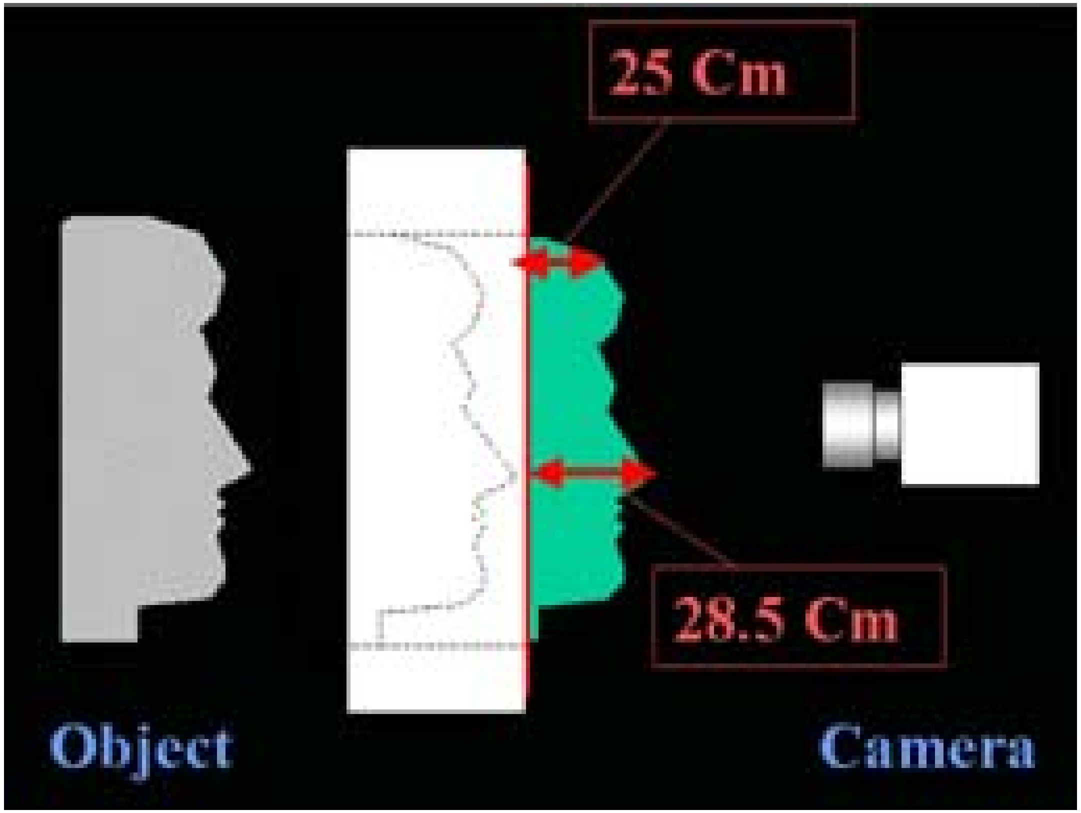
http://www.3dvsystems.com

3DV camera operation



http://www.3dvsystems.comv

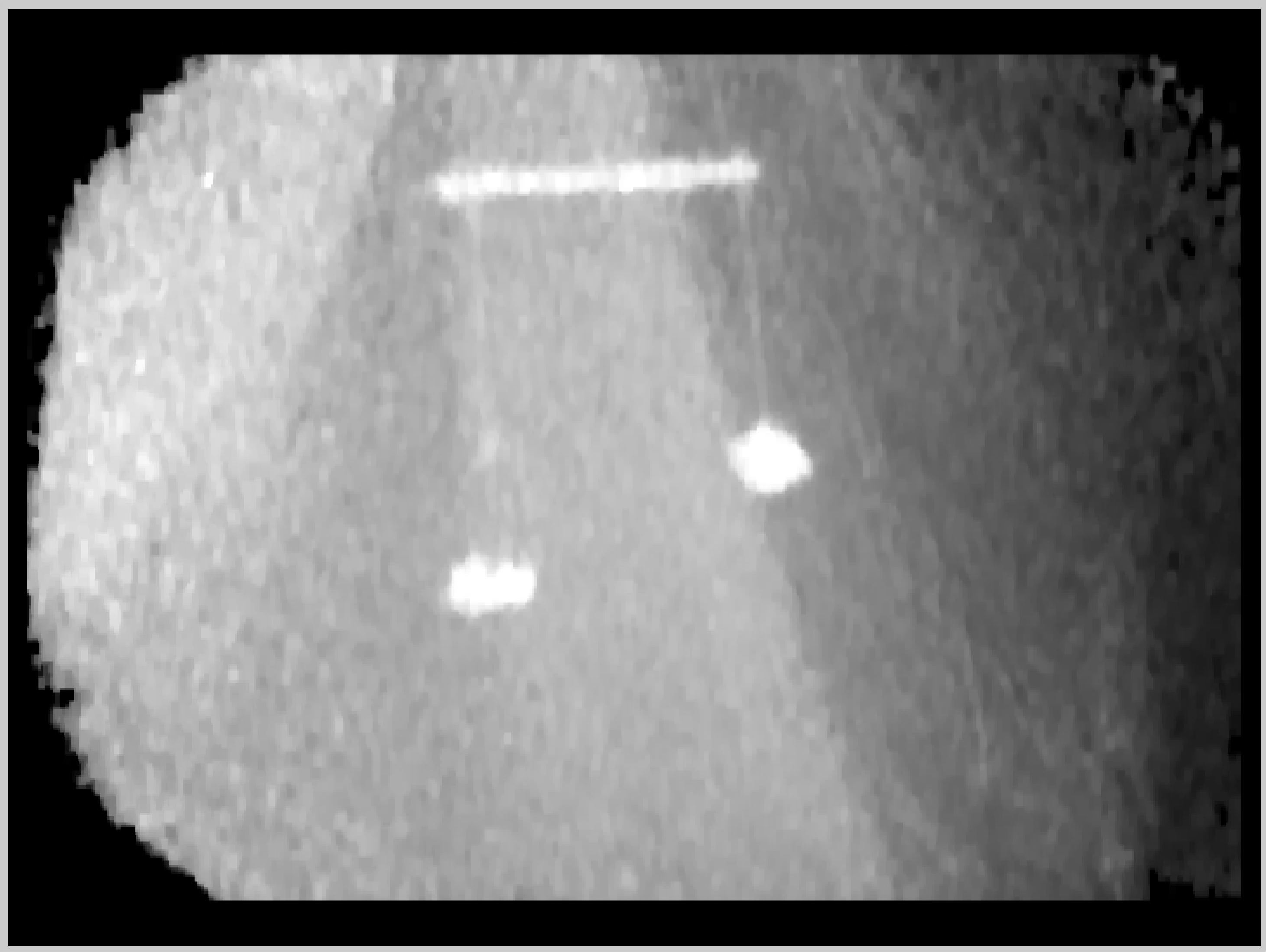
3DV camera operation



http://www.3dvsystems.com

RGB image

Z image



shapetime video image

Zitnick et al, Siggraph 2004

Show Michael Cohen slides, a selection from:

http://research.microsoft.com/~cohen/FindingMagicInAnImageStack.pdf

To appear in the ACM SIGGRAPH '04 conference proceedings

Interactive Digital Photomontage

Aseem Agarwala[1] Mira Dontcheva[1] Maneesh Agrawala[2] Steven Drucker[2] Alex Colburn[2]
Brian Curless[1] David Salesin[1,2] Michael Cohen[2]

[1]University of Washington [2]Microsoft Research

Figure 1 From a set of five source images (of which four are shown on the left), we quickly create a composite family portrait in which everyone is smiling and looking at the camera (right). We simply flip through the stack and coarsely draw strokes using the *designated source* image objective over the people we wish to add to the composite. The user-applied strokes and computed regions are color-coded by the borders of the source images on the left (middle).

In our case, we define the *cost function* C of a pixel labeling L as the sum of two terms: a *data penalty* C_d over all pixels p and an *interaction penalty* C_i over all pairs of neighboring pixels p, q:

$$C(L) \quad = \quad \sum_{p} C_d(p, L(p)) + \sum_{p,q} C_i(p, q, L(p), L(q)) \qquad (1)$$

For our application, the data penalty is defined by the distance to the image objective, whereas the interaction penalty is defined by the distance to the seam objective.

Designated color (most or least similar): the Euclidean distance in *RGB* space of the source image pixel $S_{L(p)}(p)$ from a user-specified target color. We supply a user interface for the selection of a pixel in the span that is used as the color target.

Minimum (maximum) luminance: the distance in luminance from the minimum (maximum) luminance pixel in a pixels span.

Minimum (maximum) likelihood: the probability (or one minus the probability) of the color at $S_{L(p)}(p)$, given a probability distribution function formed from the color histogram of all pixels in the span (the three color channels are histogrammed separately, using 20 bins, and treated as independent random variables).

Eraser: the Euclidean distance in *RGB* space of the source image pixel $S_{L(p)}(p)$ from the current composite color.

Minimum (maximum) difference: the Euclidean distance in *RGB* space of the source image pixel $S_{L(p)}(p)$ from $S_u(p)$, where S_u is a user-specified source image.

Designated image: 0 if $L(p) = u$, where S_u is a user-specified source image, and a large penalty otherwise.

Contrast: a measure created by subtracting the convolution of two Gaussian blur kernels computed at different scales [Reinhard et al. 2002].

We define the seam objective to be 0 if $L(p) = L(q)$. Otherwise, we define the objective as:

$$C_i(p,q,L(p),L(q)) = \begin{cases} X & \text{if matching "colors"} \\ Y & \text{if matching "gradients"} \\ X+Y & \text{if matching "colors \& gradients"} \\ X/Z & \text{if matching "colors \& edges"} \end{cases}$$

where

$$\begin{aligned} X &= \|S_{L(p)}(p) - S_{L(q)}(p)\| + \|S_{L(p)}(q) - S_{L(q)}(q)\| \\ Y &= \|\nabla S_{L(p)}(p) - \nabla S_{L(q)}(p)\| + \|\nabla S_{L(p)}(q) - \nabla S_{L(q)}(q)\| \\ Z &= E_{L(p)}(p,q) + E_{L(q)}(p,q)) \end{aligned}$$

and $\nabla S_z(p)$ is a 6-component color gradient (in R, G, and B) of image z at pixel p, and $E_z(p,q)$ is the scalar edge potential between two neighboring pixels p and q of image z, computed using a Sobel filter.

Figure 2 A set of macro photographs of an ant (three of eleven used shown on the left) taken at different focal lengths. We use a global *maximum contrast* image objective to compute the graph-cut composite automatically (top left, with an inset to show detail, and the labeling shown directly below). A small number of remaining artifacts disappear after gradient-domain fusion (top, middle). For comparison we show composites made by Auto-Montage (top, right), by Haeberli's method (bottom, middle), and by Laplacian pyramids (bottom, right). All of these other approaches have artifacts; Haeberli's method creates excessive noise, Auto-Montage fails to attach some hairs to the body, and Laplacian pyramids create halos around some of the hairs.

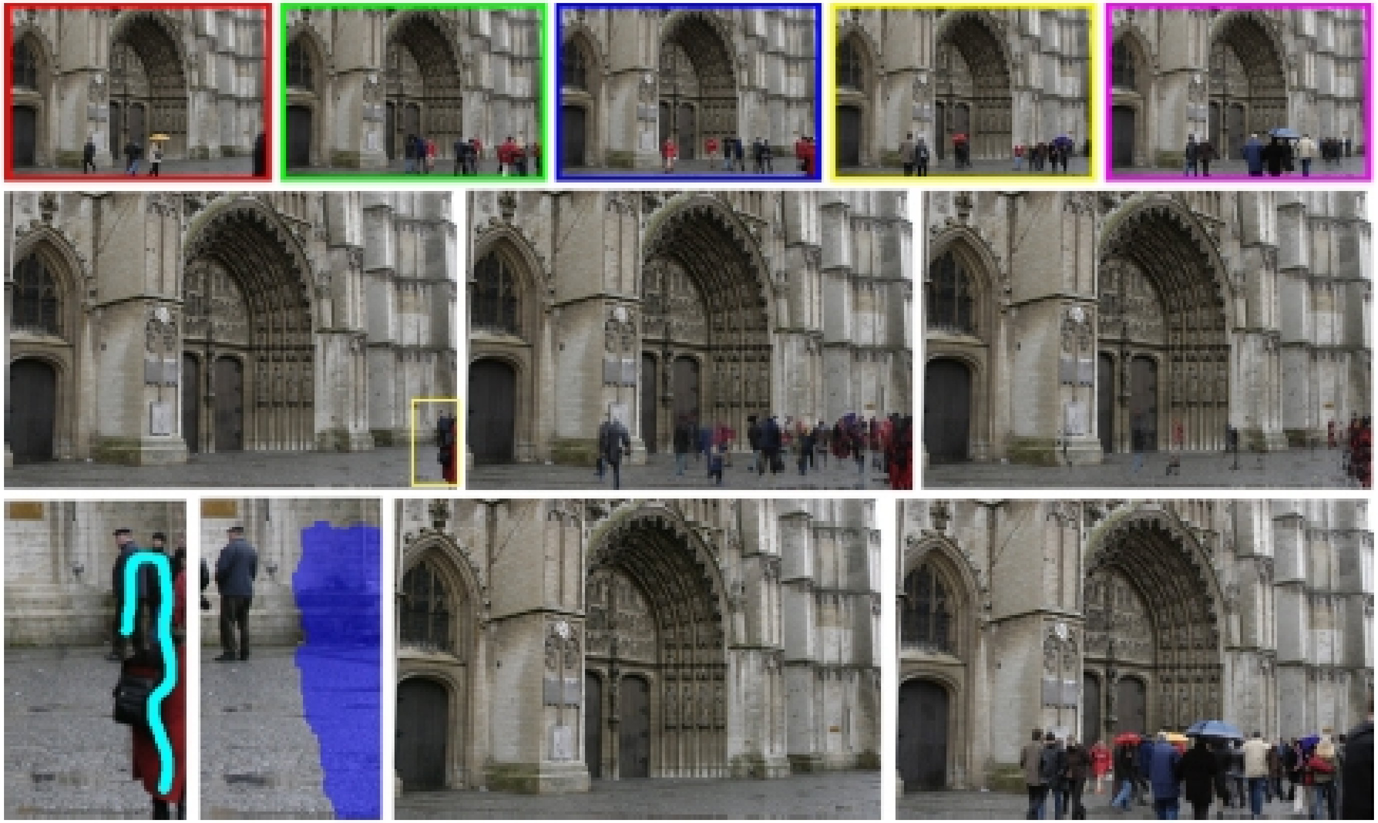


Figure 12 From a set of five images (top row) we create a relatively clean background plate using the *maximum likelihood* objective (middle row, left). The next two images to the right show that our result compares favorably to a per-pixel median filter, and a per-pixel maximum likelihood objective, respectively. An inset of our result (bottom row, left) shows several remaining people. The user paints over them with the *eraser* objective, and the system offers to replace them with a region, highlighted in blue, of the fourth input image. The user accepts this edit, and then applies gradient-domain fusion to create a final result (bottom row, middle). Finally, using a *minimum likelihood* image objective allows us to quickly create a large crowd (bottom right).

Figure 5 To capture the progression of time in a single image we generate this stroboscopic image from a video sequence. Several video frames are shown in the first column. We first create a background image using the *maximum likelihood* objective (second column, top) and then add it to the stack. Then, we use the *maximum difference* objective to compute a composite that is maximally different from the background (second column, bottom). A lower weight for the image objective results in fewer visible seams but also fewer instances of the girl (third column, top). Beginning with the first result, the user removes the other girls by brushing in parts of the background and one of the sources using the *designated source* objective (third column, bottom) to create a final result (right).

Figure 6 We use a set of portraits (first row) to mix and match facial features, to either improve a portrait, or create entirely new people. The faces are first hand-aligned, for example, to place all the noses in the same location. In the first two images in the second row, we replace the closed eyes of a portrait with the open eyes of another. The user paints strokes with the *designated source* objective to specify desired features. Next, we create a fictional person by combining three source portraits. Gradient-domain fusion is used to smooth out skin tone differences. Finally, we show two additional mixed portraits.

Demonstrate MSR group shot program, downloadable from

http://research.microsoft.com/~cohen/

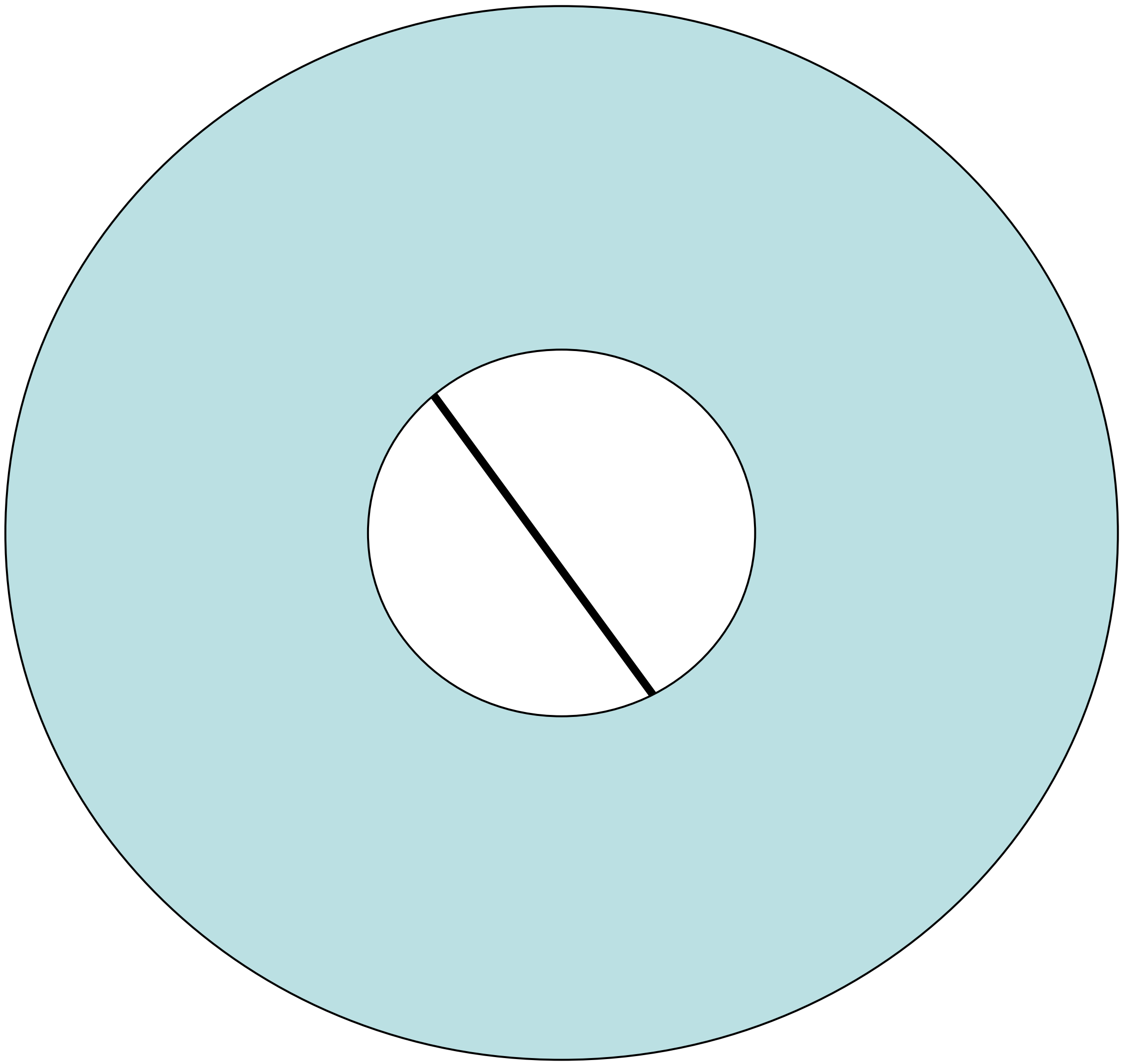
or

http://research.microsoft.com/projects/GroupShot/

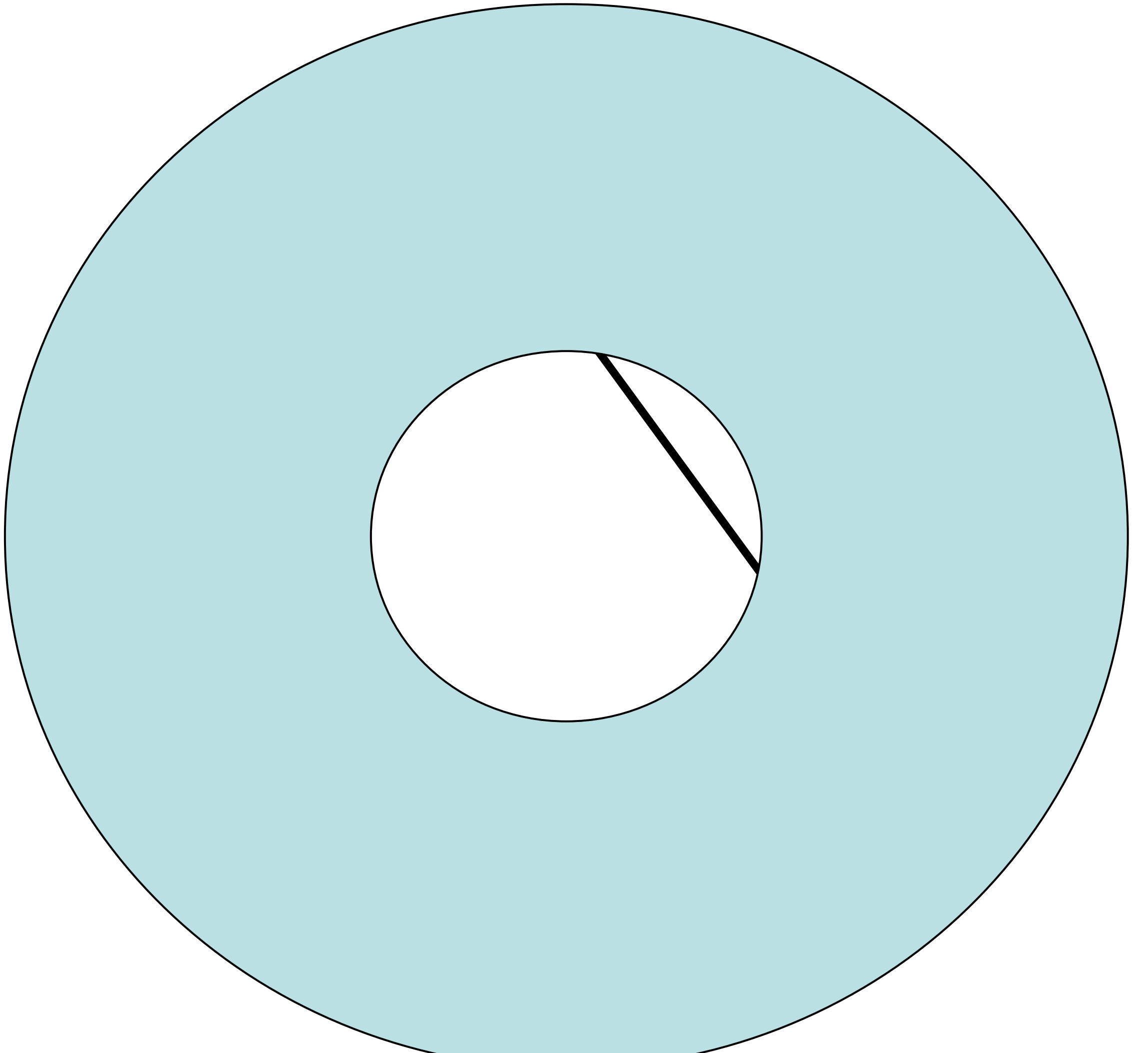
Analyzing and synthesizing motion

Bill Freeman
Fredo Durand
MIT Computational Photography, 6.882
April 20, 2006

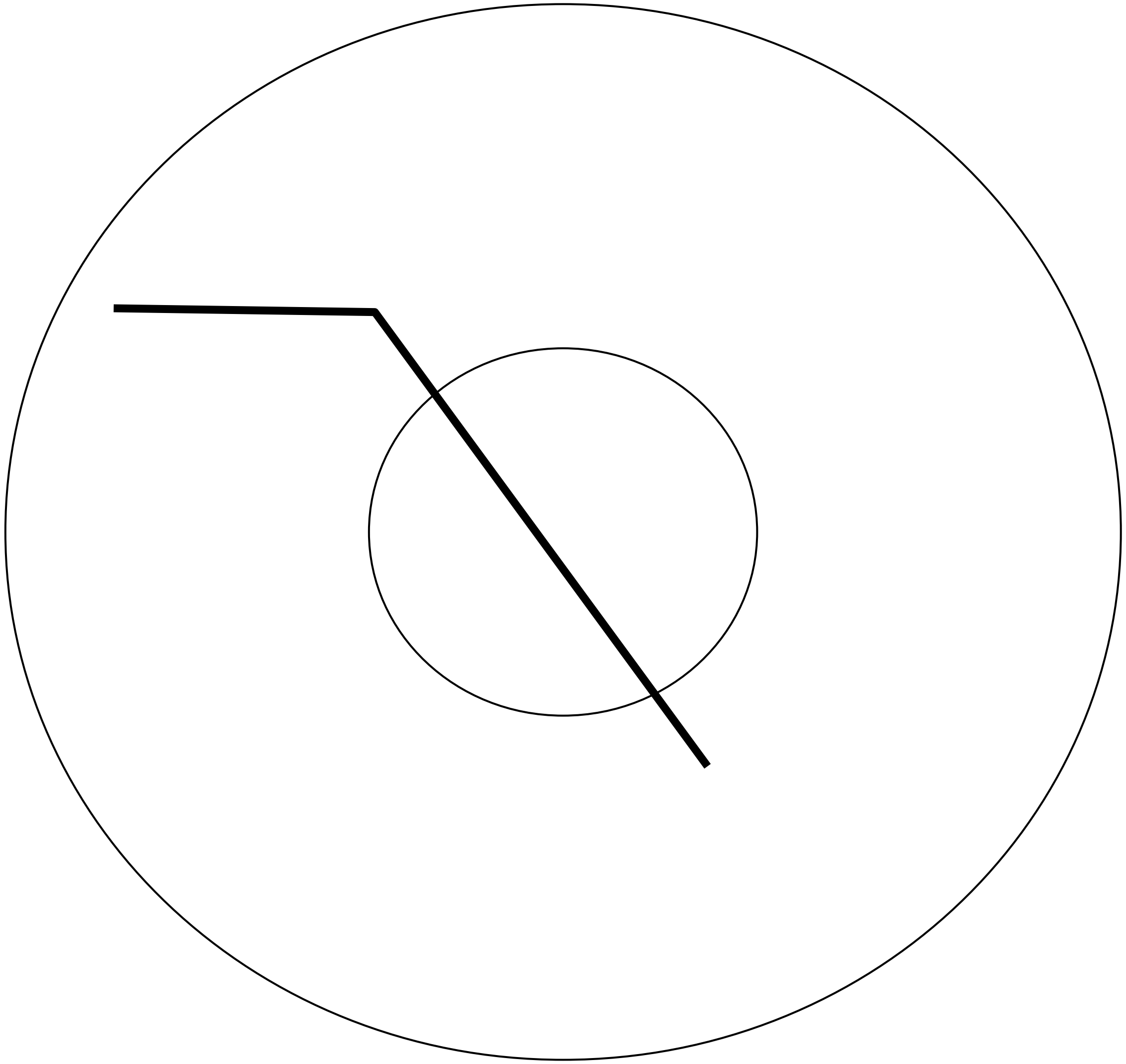
Aperture Problem and Normal Flow



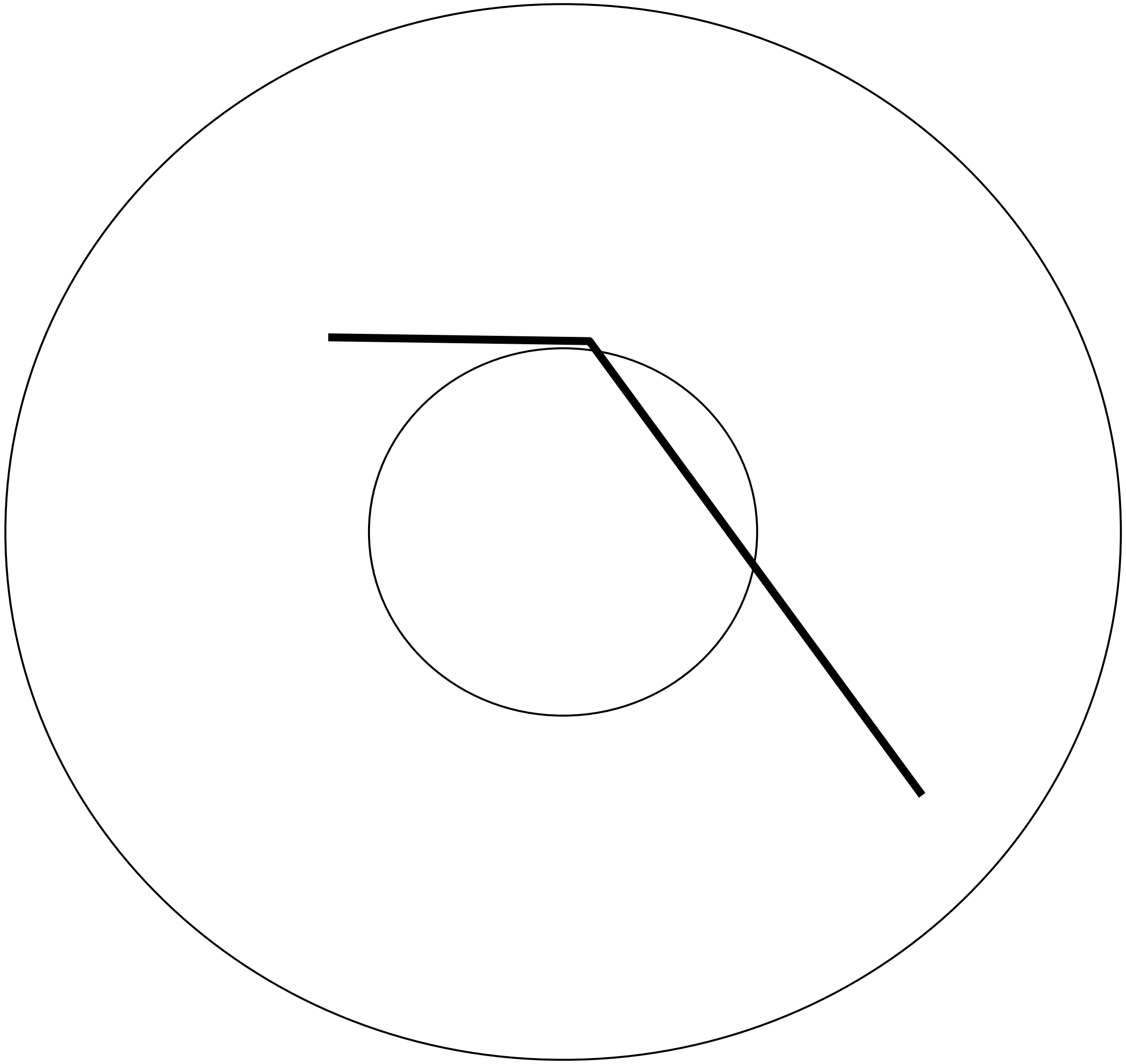
Aperture Problem and Normal Flow



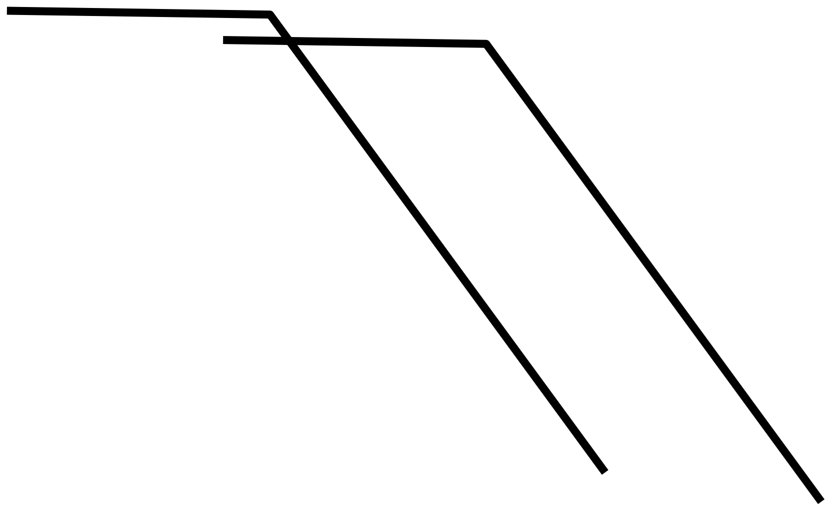
Aperture Problem and Normal Flow



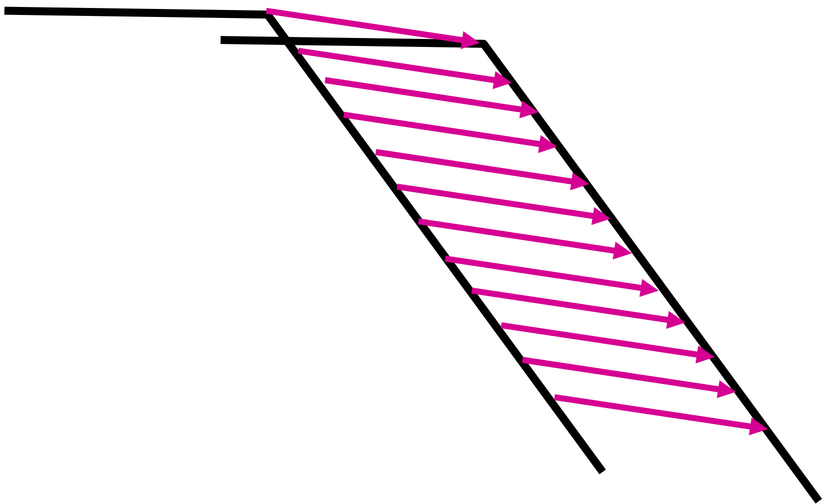
Aperture Problem and Normal Flow



Aperture Problem and Normal Flow



Aperture Problem and Normal Flow



Optical flow constraint equation

Brightness should stay constant as you track motion

$$I(x+u\delta t, y+v\delta t, t+\delta t) = I(x,y,t)$$

1st order Taylor series, valid for small δt

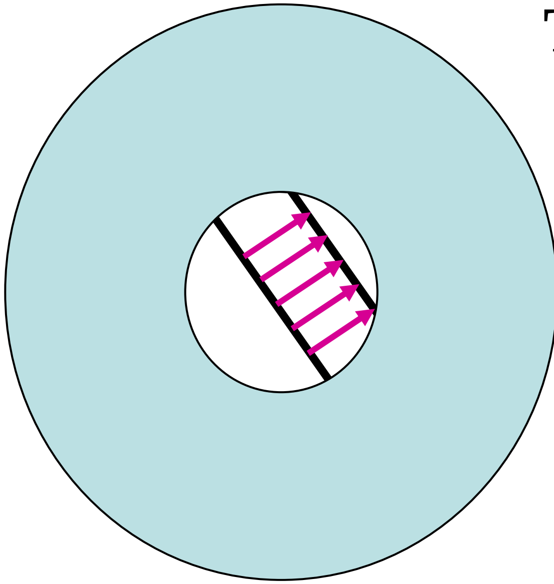
$$I(x,y,t) + u\delta t I_x + v\delta t I_y + \delta t I_t = I(x,y,t)$$

Constraint equation

$$\boxed{uI_x + vI_y + I_t = 0}$$

"BCCE" - Brightness Change Constraint Equation

Aperture Problem and Normal Flow

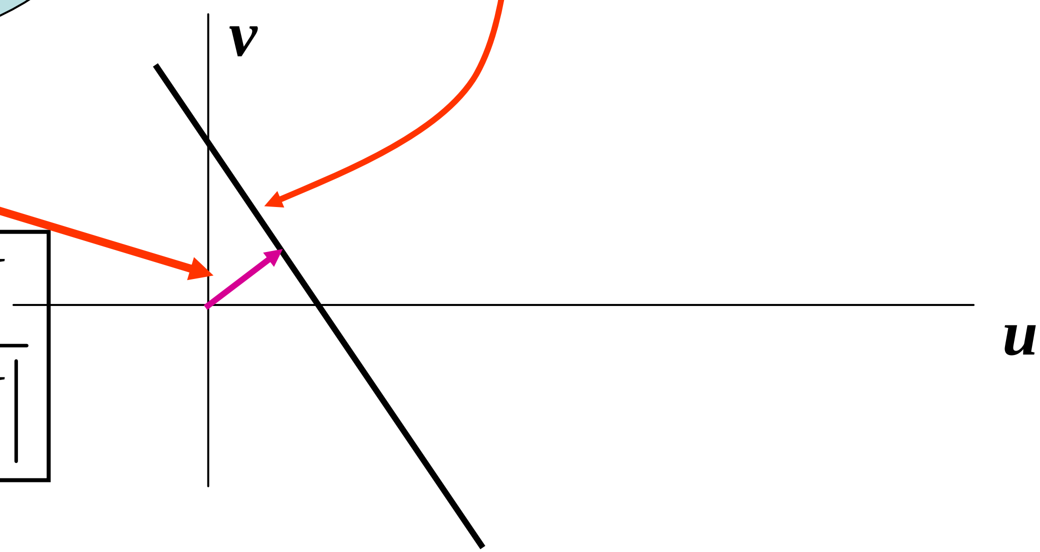


The gradient constraint:

$$I_x u + I_y v + I_t = 0$$

$$\nabla I \bullet \vec{U} = 0$$

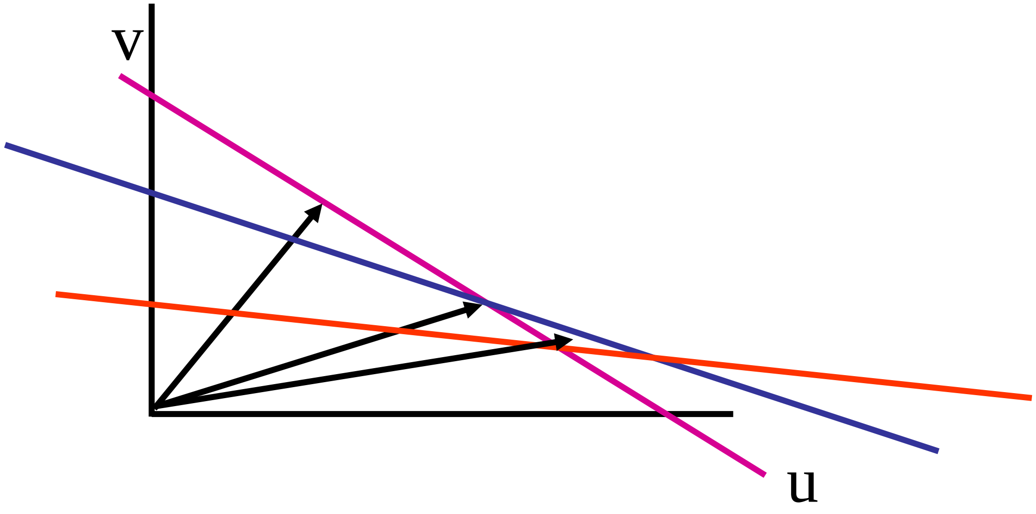
Defines a line in the *(u,v)* space



Normal Flow:

$$u_\perp = -\frac{I_t}{|\nabla I|}\frac{\nabla I}{|\nabla I|}$$

Combining Local Constraints



$$\nabla I^1 \bullet U = -I_t^1$$

$$\nabla I^2 \bullet U = -I_t^2$$

$$\nabla I^3 \bullet U = -I_t^3$$

etc.

Lucas-Kanade

(a good, generic motion analysis method): Integrate gradients over a patch

Assume a single velocity, u, v, for all pixels within an image patch. Find the (u, v) that minimizes the BCCE squared residual over the patch:

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

Setting derivative w.r.t. (u, v) equal to zero gives:

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

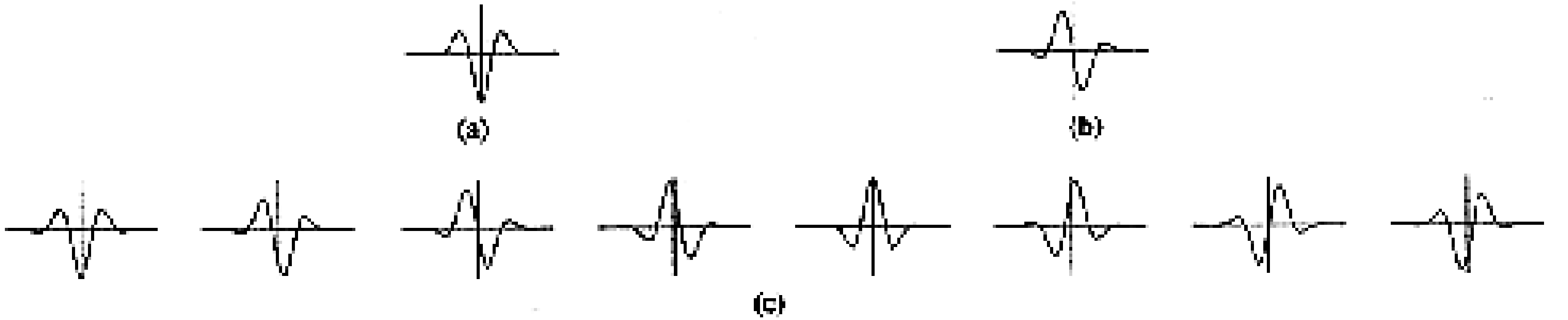
Note similarity of LHS matrix to Harris corner detector. When full-rank (corner-like), specifies a unique (u, v).

Motion without movement

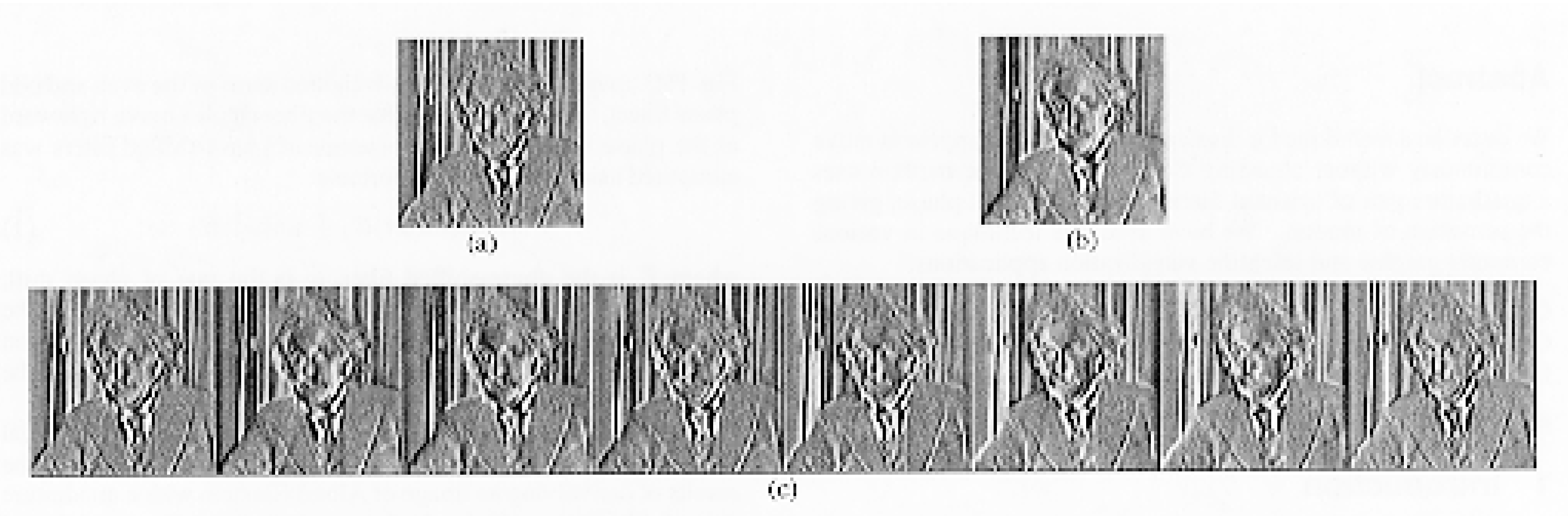
Joint work with Ted Adelson and David Heeger, MIT

1991

A linear combination of quadrature-phase filters can advance the local phase



Convolved with an image, the image data now modulates the local amplitude. People mis-attribute the phase advance to translation.



(Steerable filters allow synthesizing motion in arbitrary directions.)

Motion without movement video

http://www.cs.yorku.ca/~kosta/Motion_Without_Movement/Motion_Without_Movement.html

http://www.cs.yorku.ca/~kosta/Motion_Without_Movement/Motion_Without_Movement.html

Konstantinos G. Derpanis

Motion Magnification

(go to other slides…)