Image alignment and stabilization

Frédo Durand
MIT EECS & CSAIL

Some slides by Alyosha Efros,
Steve Seitz & Rick Szeliski

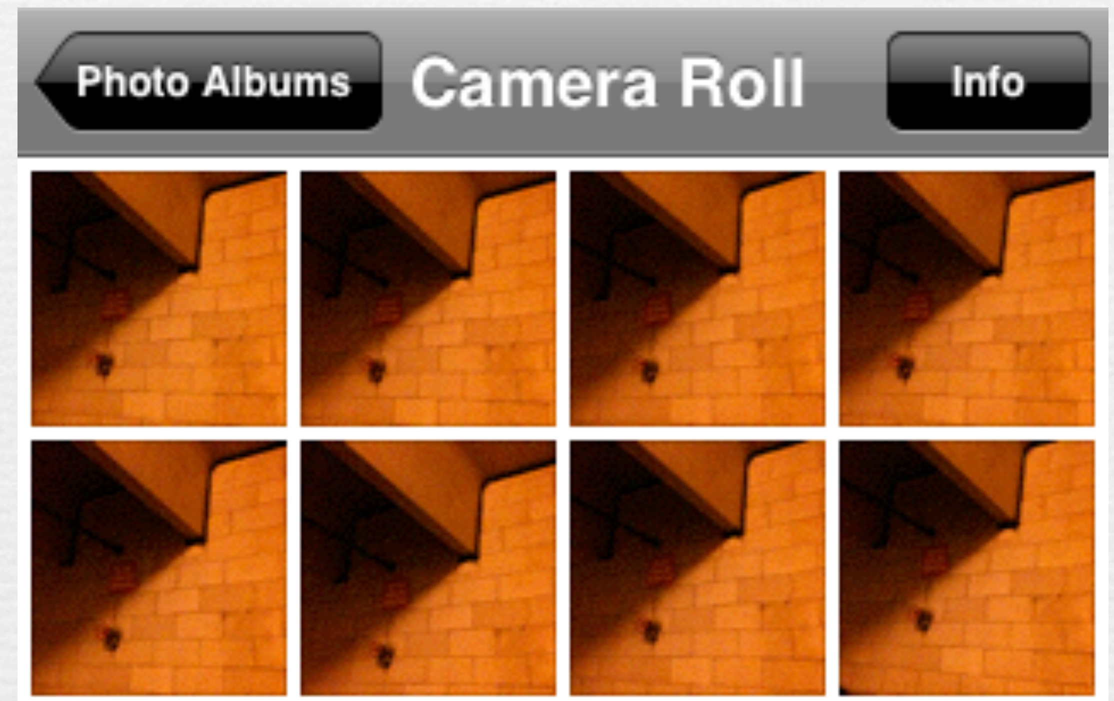
Align & combine multiple images

- High dynamic range imaging
- Flash no flash
- Denoising
- Lucky imaging
- Depth of field extension,
- Panoramas
- Photomontage
- Video stabilization
- 3D compositing

DENOISING

Tutorial by Eugene Hsu

- http://iphone.squicky.org/noise-84
  - Follow him at http://twitter.com/hsugene/
- Take multiple shots of a static scene
- Align
- Average to reduce noise.



Single frame

Average of 8 frames

EXTENDED DEPTH of FIELD

Focal stack DoF extensions

- **Capture N images focused at different distances**
- **For each output pixel, choose the sharpest image**
  - e.g. look at local variance, gradient.

From Agarwala et al.

Focal stack

Focal stack

Focal stack

Focal stack

Focal stack

Focal stack

Focal stack

Focal stack

Focal stack

Focal stack

Focal stack

Focal stack

Focal stack

Montage

Macro montage

- **55 images here**

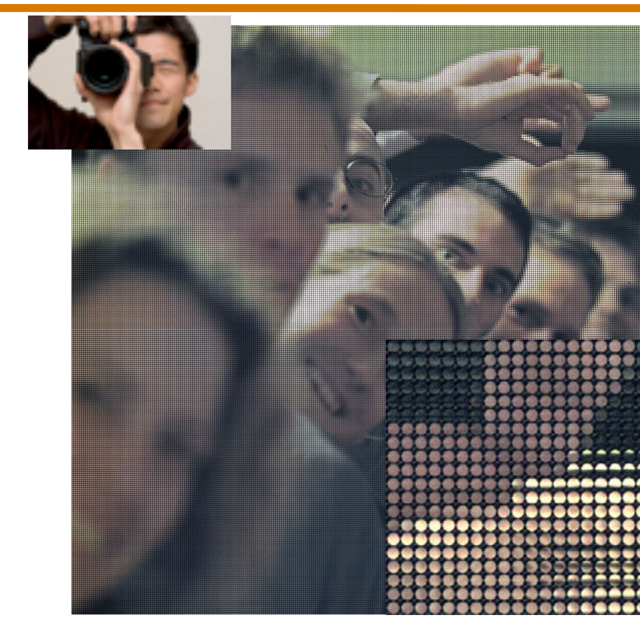
Software

- **Helicon focus**
- **http://www.heliconsoft.com/animation/Krebs_fly1/index.html**
- **http://www.krebsmicro.com/**

Focal stack & plenoptic camera

Light Field Photography with a Hand-Held Plenoptic Camera, Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, Pat Hanrahan

- **Capture light field**



- **Refocus to create focal stack**

- **Use photomontage to generate all-focus image**

Focal stack & plenoptic camera

Figure 15: *Left*: Extended depth of field computed from a stack of photographs focused at different depths. *Right:* A single sub-aperture image, which has equal depth of field but is noisier.

From Ng et al. http://graphics.stanford.edu/papers/lfcamera/

References

- http://www.janrik.net/ptools/ExtendedFocusPano12/index.html
- http://www.outbackphoto.com/workflow/wf_72/essay.html
- http://grail.cs.washington.edu/projects/photomontage/
- http://people.csail.mit.edu/hasinoff/timecon/

- http://graphics.stanford.edu/papers/lfcamera/

IMAGE STACKS, PHOTOMONTAGE

Interactive Digital Photomontage

- Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, Michael Cohen. Interactive Digital Photomontage. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004), 2004.
- Set of aligned images of same scene
- Combine in clever ways
  - automatic or user-specified
- More about the exact combination next time.

Family portrait challenge

Family portrait challenge

Family portrait challenge

Family portrait challenge

Family portrait challenge

Digital photomontage

Tourist removal

Tourist removal

Tourist removal

Tourist removal

Tourist removal

Tourist removal

Wire removal

Wire removal

Wire removal

Wire removal

Wire removal

Wire removal

Wire removal

Wire removal

Wire removal

Wire removal

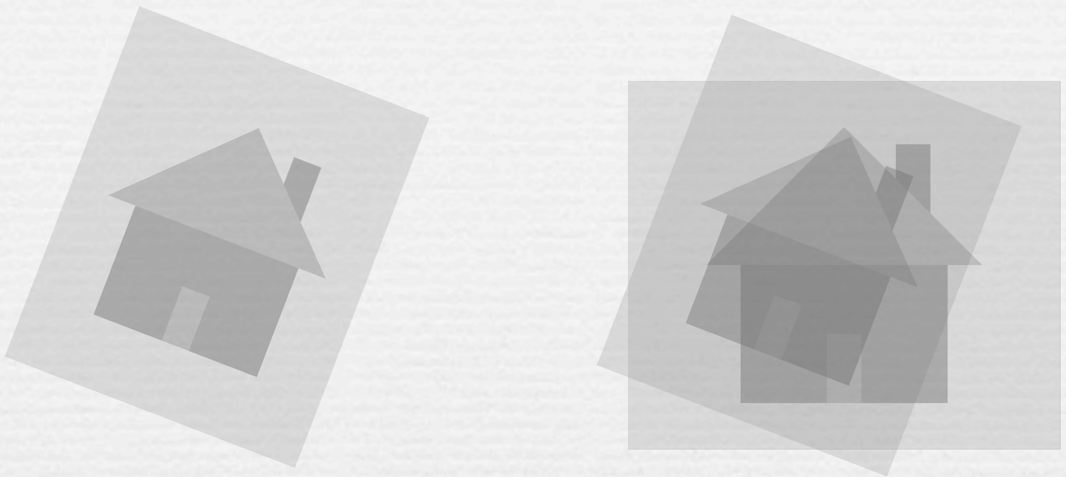
IMAGE ALIGNMENT

Image alignment goals

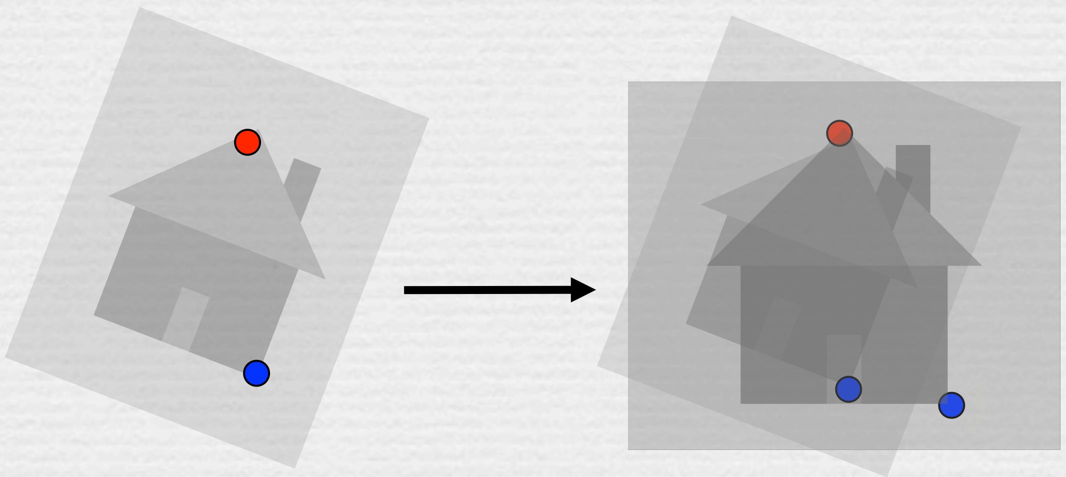
- Multiple-exposure photography
  - Denoising, depth of field extension, etc.
  - Lucky imaging
  - Flash no flash, Panoramas, HDR
- Photomontage
- Video stabilization
- Matchmove
  - Recover 3D camera path for computer graphics object compositing

Approaches: dense vs. sparse

- Pixel-based alignment
  - match all pixels
  - aka dense



- Feature-based alignment
  - match only special pixels such as corners
  - aka sparse



Approaches: model or not

- Model based : restricted range of motions
  - e.g. translation, affine, homography
- Non-parametric
  - motion could be anything

BRUTE FORCE

Brute force: dense & model-based

- Given low-order motion model
- Find parameters that minimize Sum of Square difference

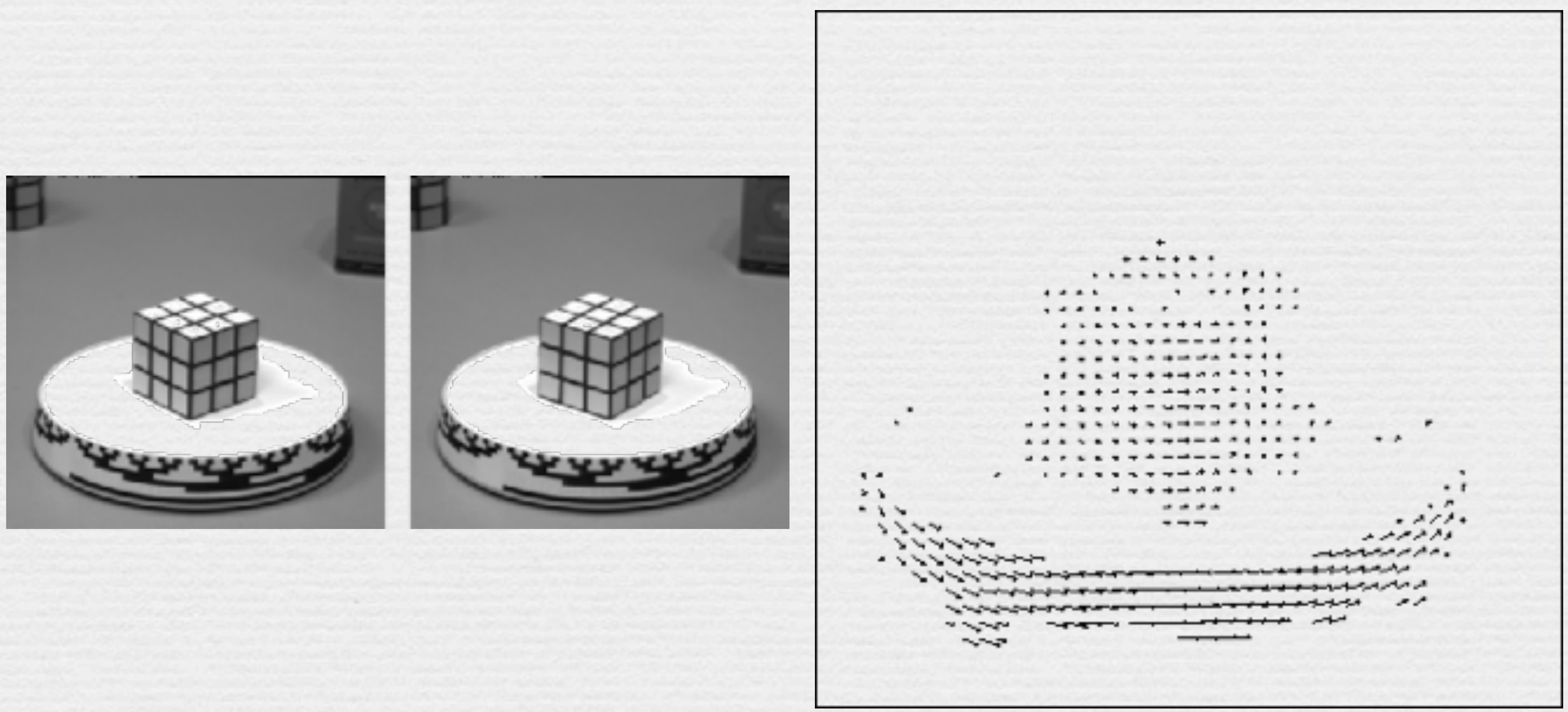
- e.g. for translation:

```
 for tx=x0:step:x1,
  for ty=y0:step:y1,
     compare image1(x,y) to image2(x+tx,y+ty)
  end;
end;
```

OPTICAL FLOW

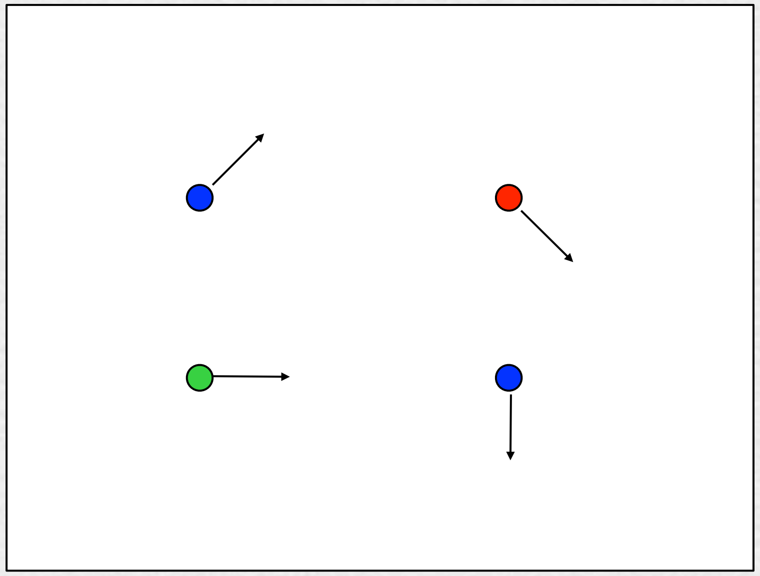
Optical flow: dense, non-parametr.

- Estimate motion of each pixel separately

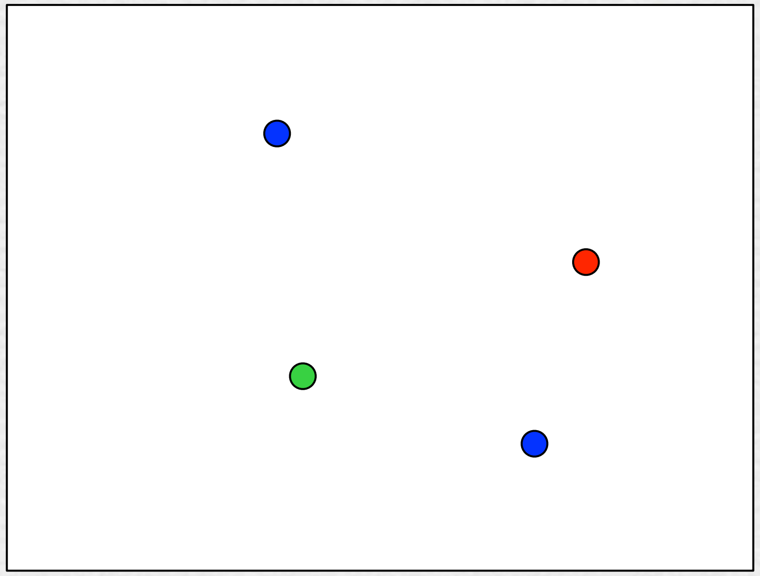


Problem statement

- Motion from image H to image I
- Given pixel in H, find nearby pixel in I with same color
- Assumptions:
  - small motion
  - color (or brightness) constancy



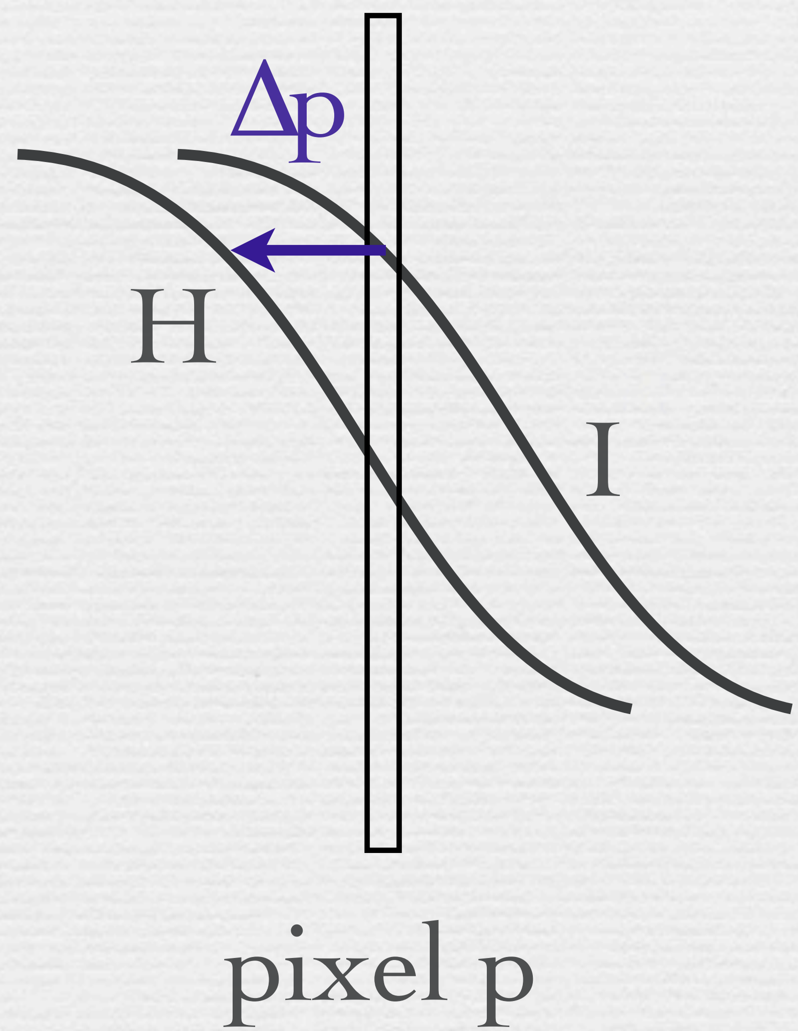
$H(x, y)$



$I(x, y)$

1D brightness constancy

- Goal: Estimate motion by observing a single pixel just look at brightness variation between H(p) and I(p)
- Solution: use first-order model

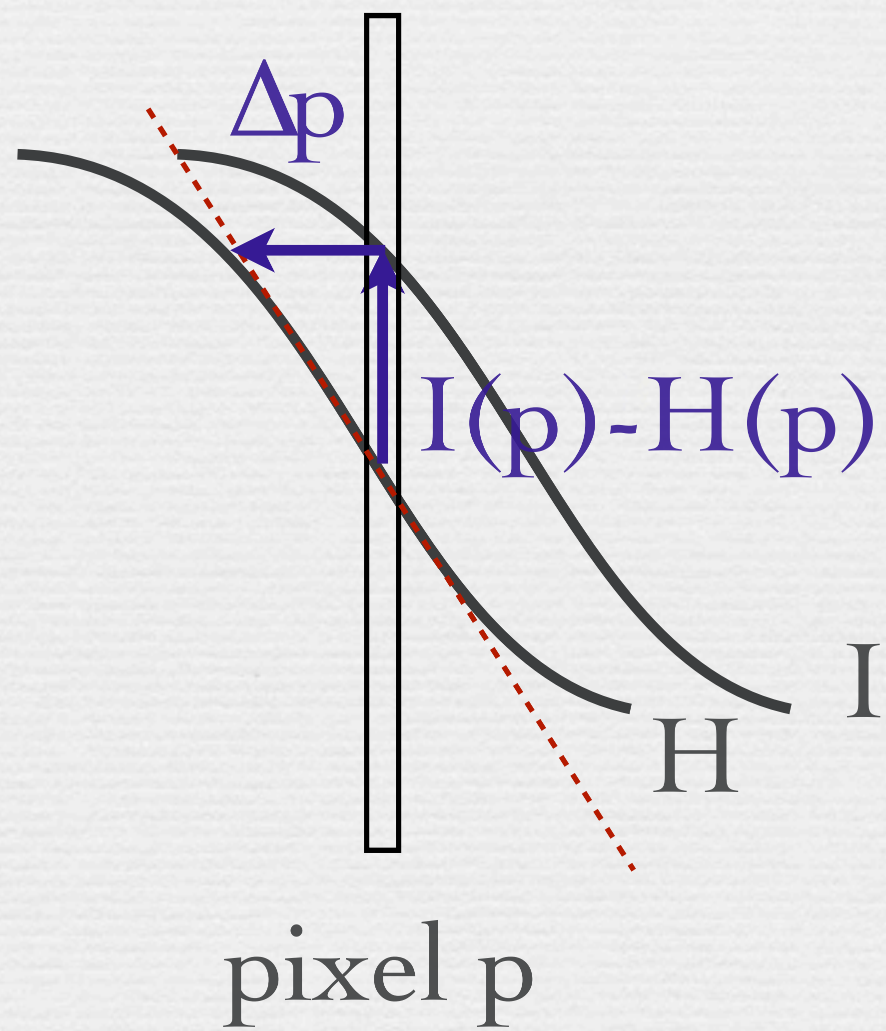


1D brightness constancy

- We observe a given brightness variation at p
- We know the local image derivative

$I(p) = H(p + \Delta p)$
$I(p) \approx H(p) + H'(p)\Delta p$
$I(p) - H(p) \approx H'(p)\Delta p$

$\mathbf{\Delta p \approx [I(p) - H(p)] / H'(p)}$



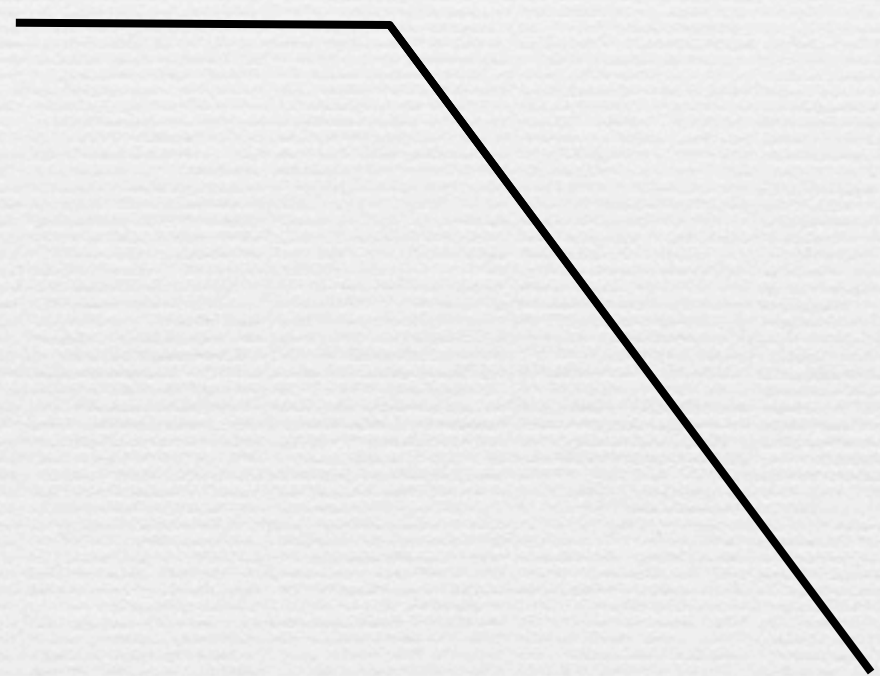
Same in 2D

- If It is the time derivative and [u v] is the flow:

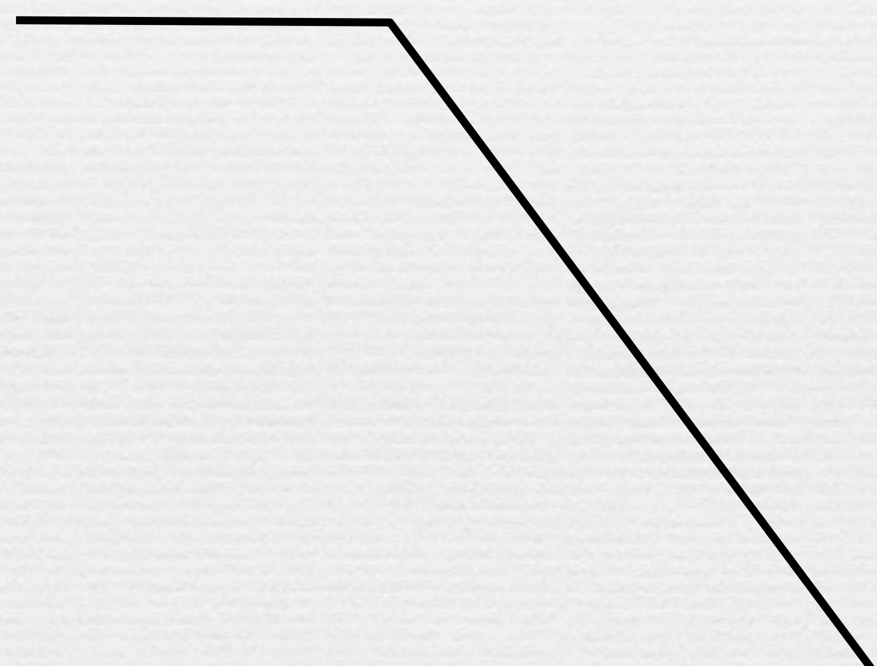
$$0 = I_t + \nabla I \cdot [u \;\; v]$$

- bean counting:
  - 2 unknowns per pixel : [u,v]
  - only one equation
  - Only the component along the gradient is known (aperture problem)
  - Explains the barberpole illusion : http://www.sandlotscience.com/Ambiguous/barberpole.htm

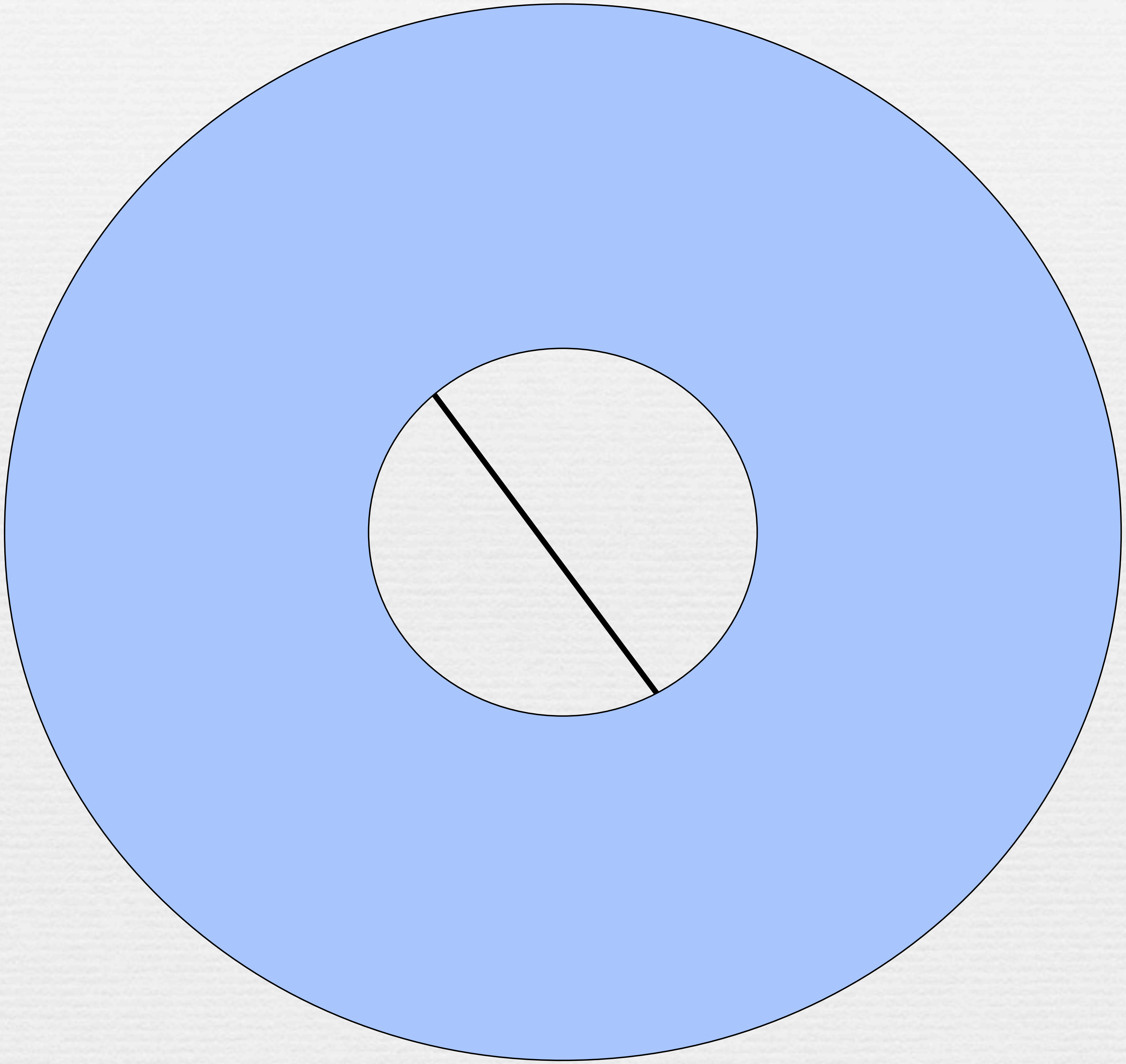
Aperture problem



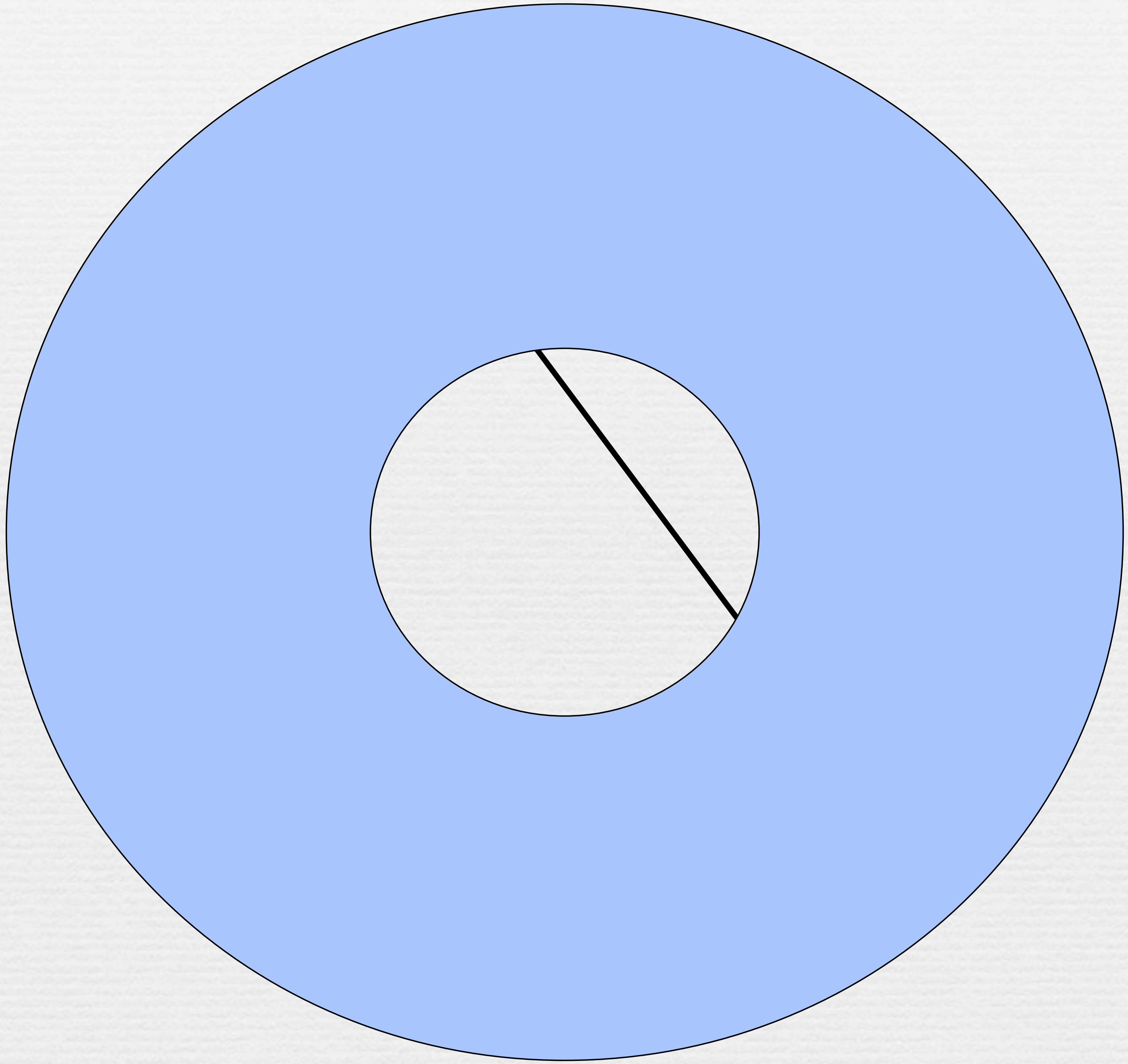
Aperture problem



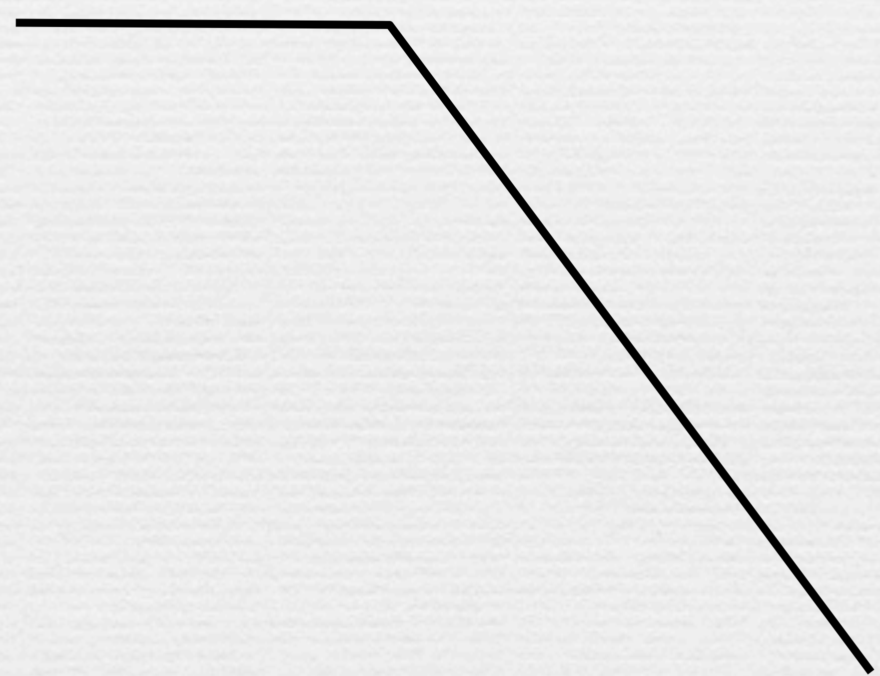
Aperture problem



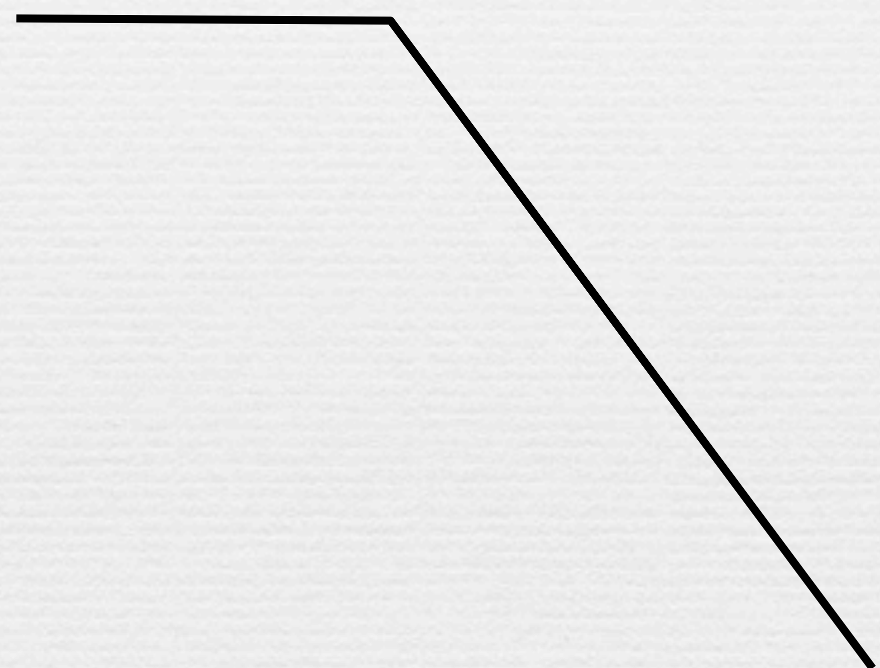
Aperture problem



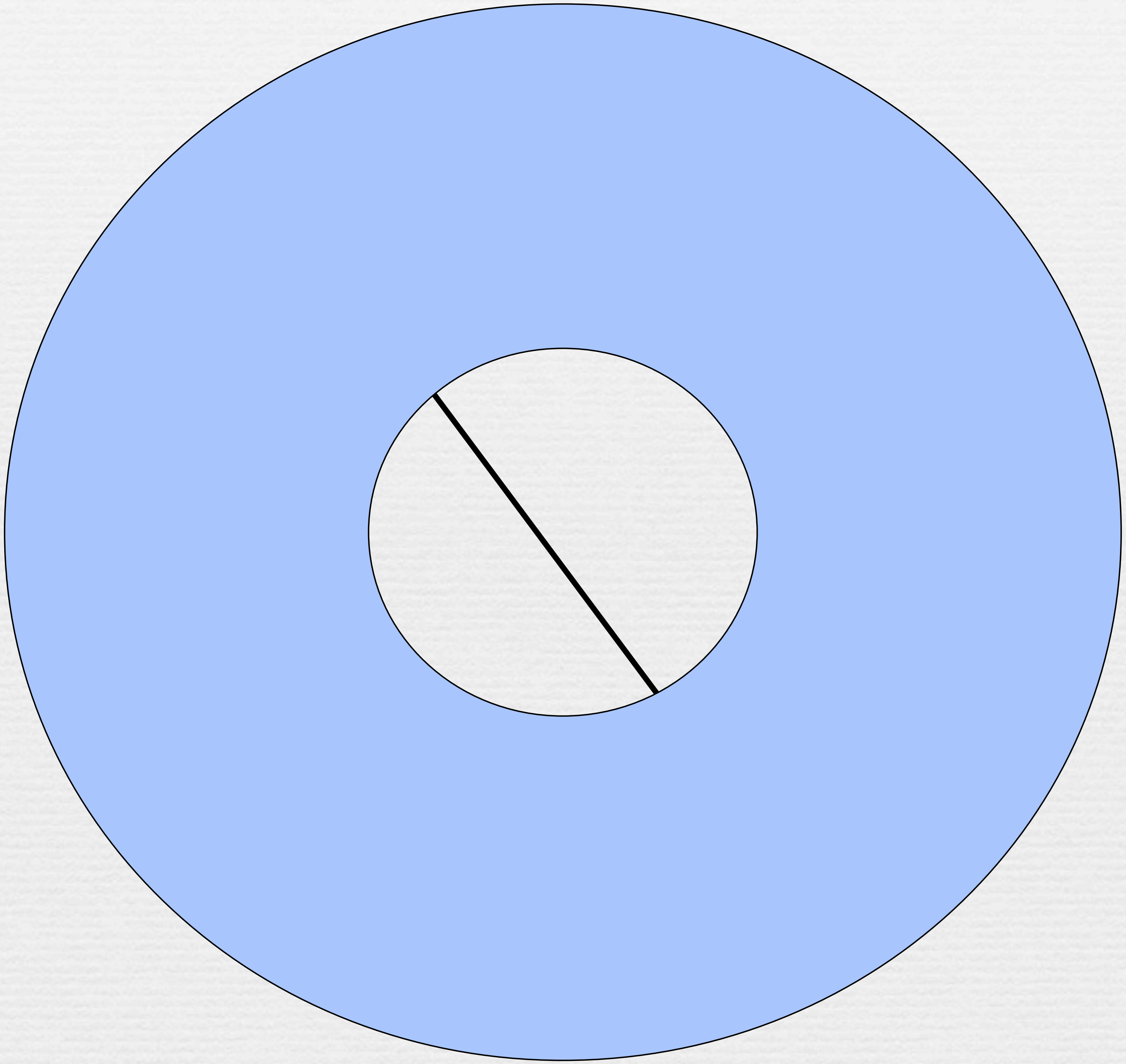
Aperture problem



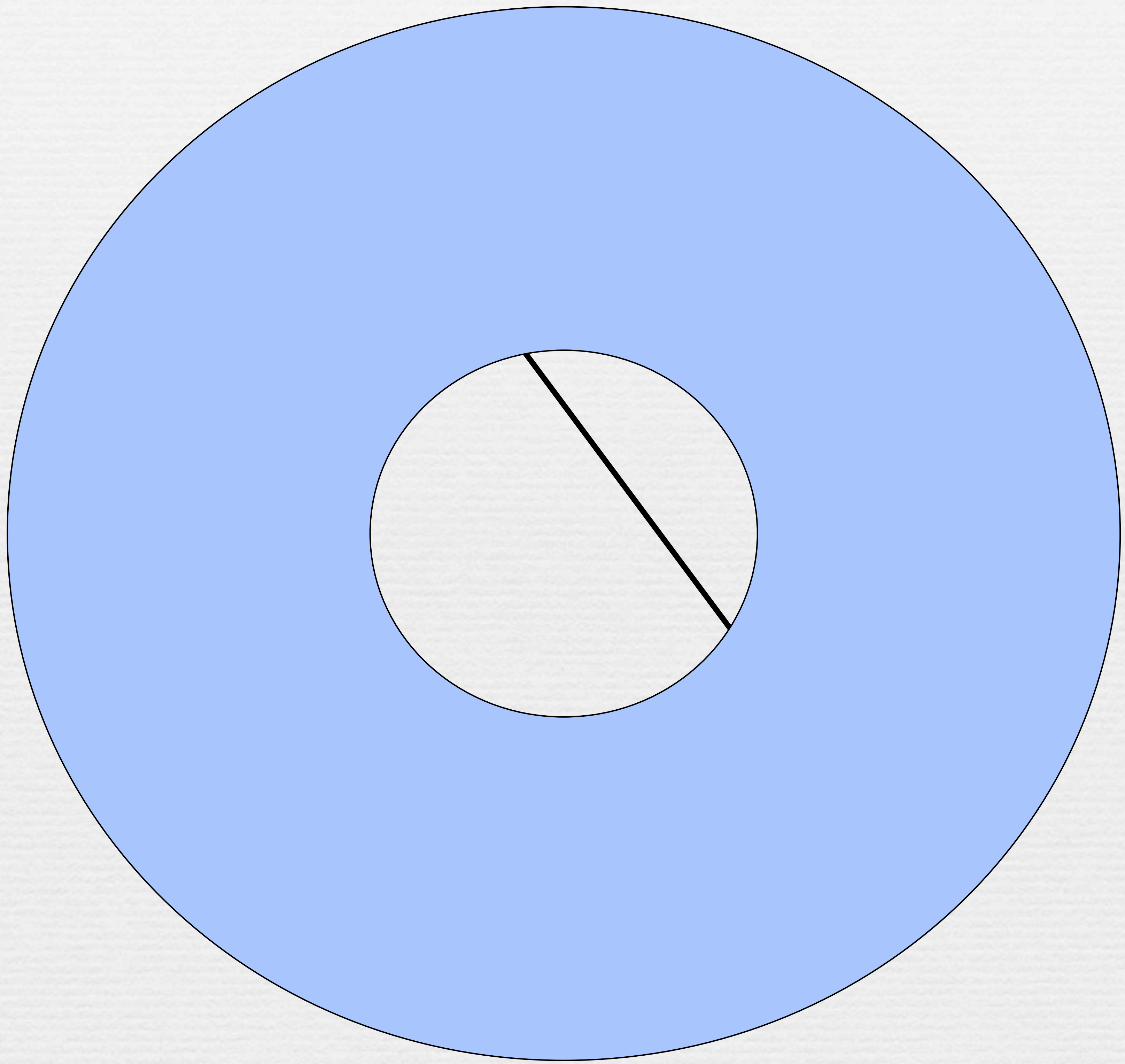
Aperture problem



Aperture problem



Aperture problem



Solving the aperture problem

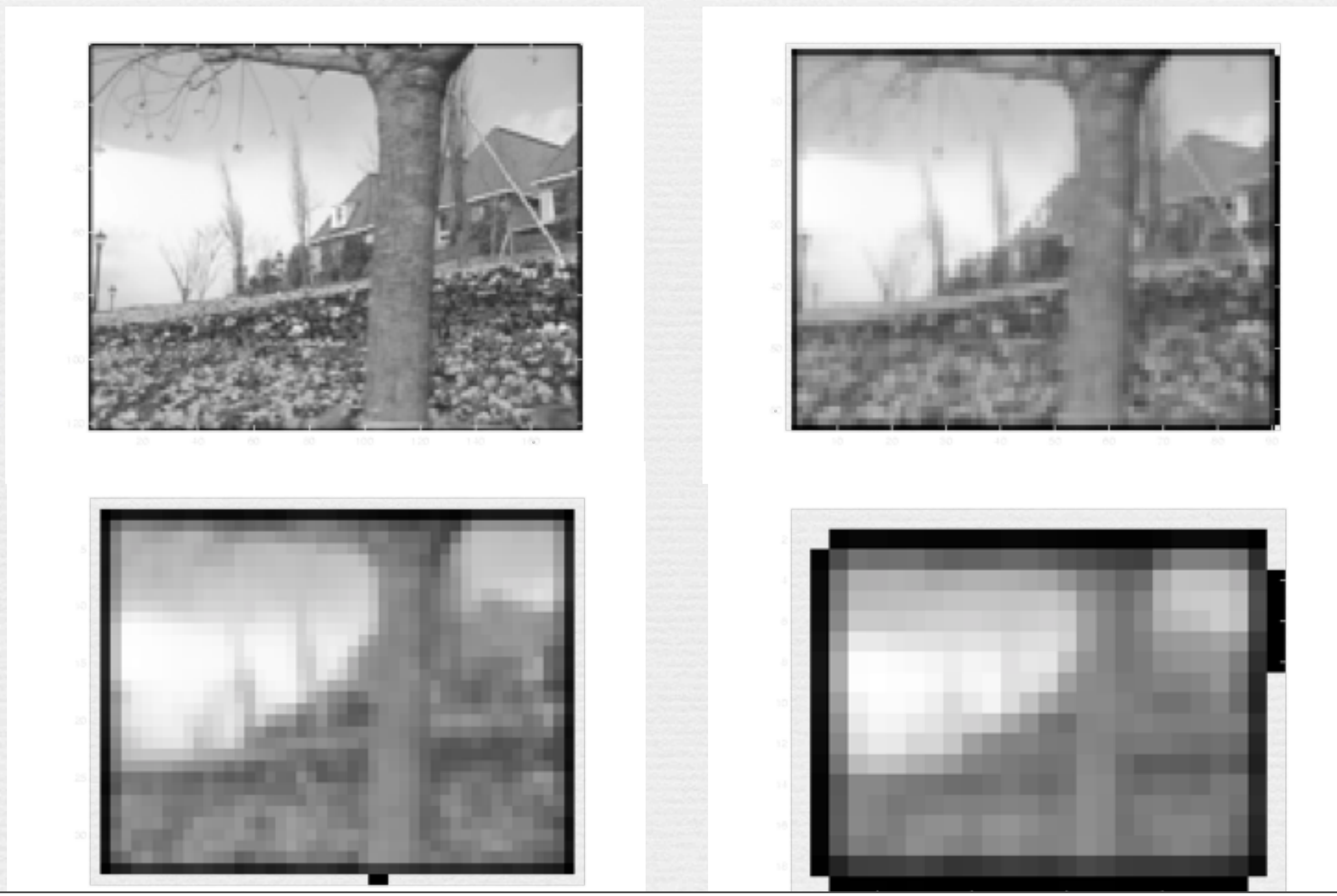
- Idea: use multiple pixels, assume flow is smooth

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \ v]$$

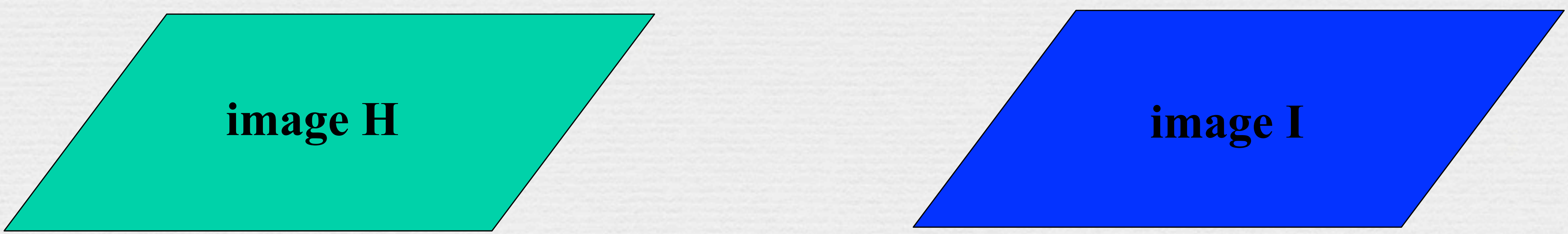
$$\underbrace{\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix}}_{\substack{A \\ 25\times 2}} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\substack{d \\ 2\times 1}} = - \underbrace{\begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}}_{\substack{b \\ 25\times 1}}$$

Small motion problem

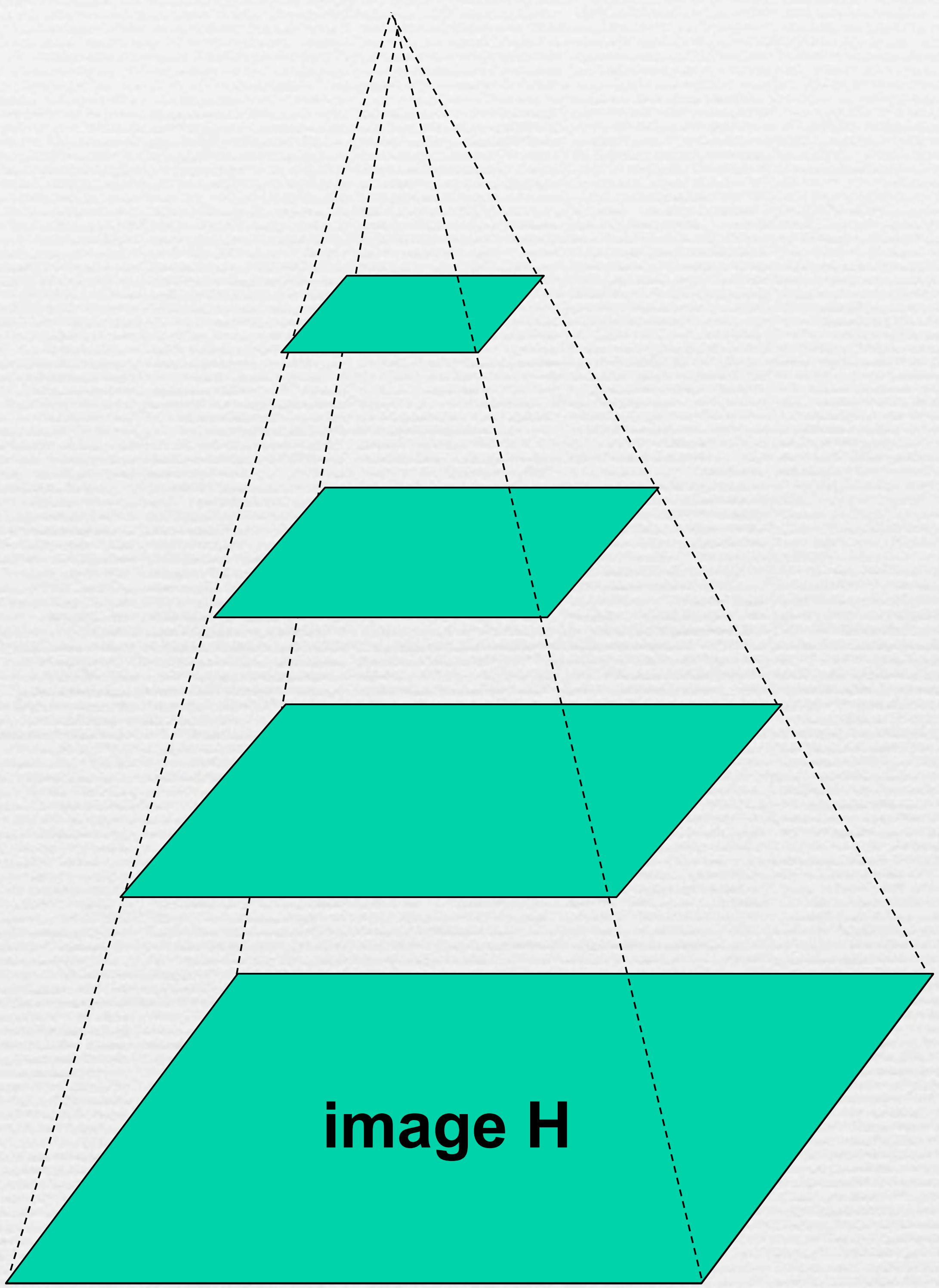
- The first order model breaks quickly
- Solution: reduce image resolution!



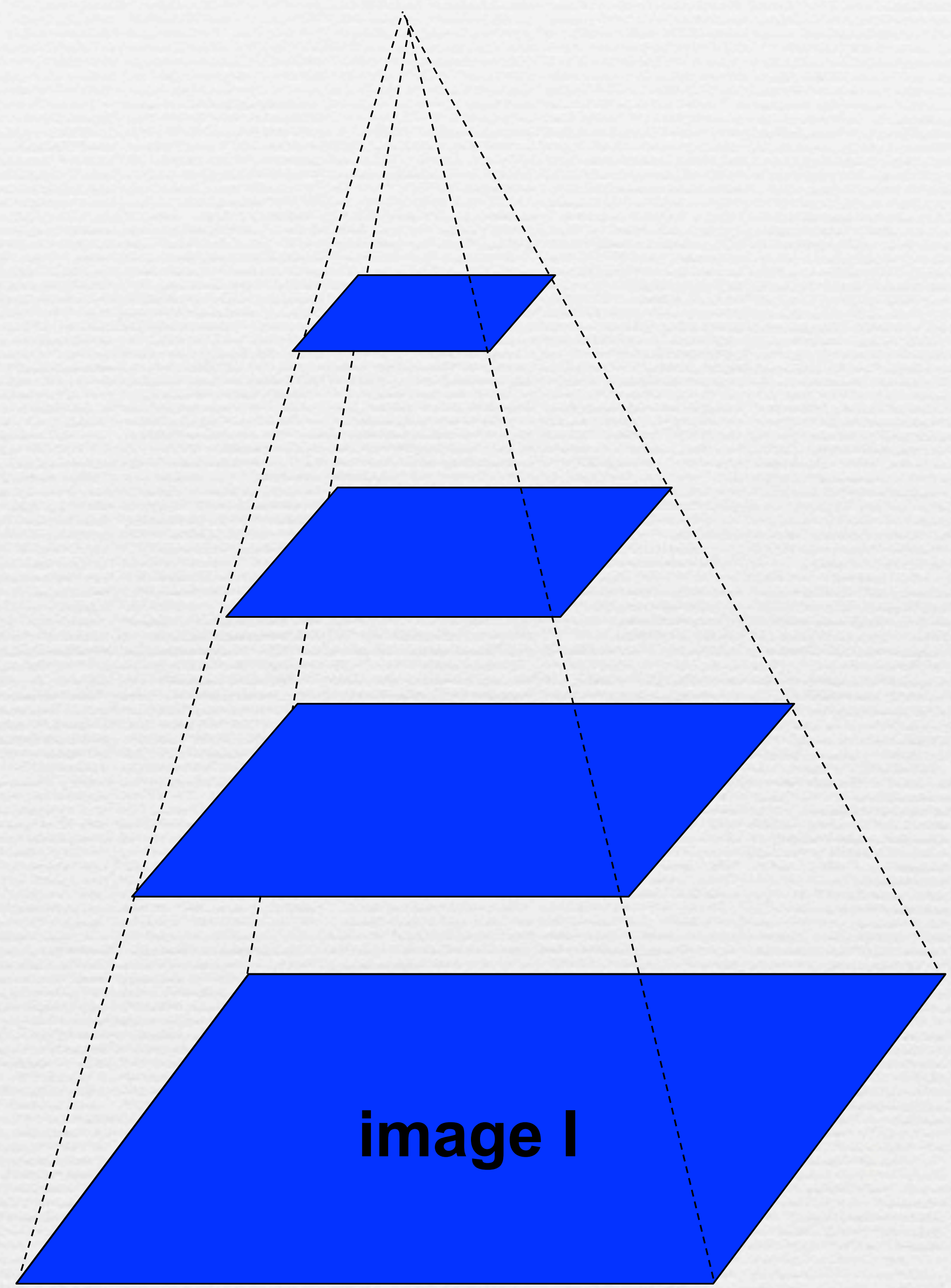
Coarse-to-fine optical flow



Coarse-to-fine optical flow

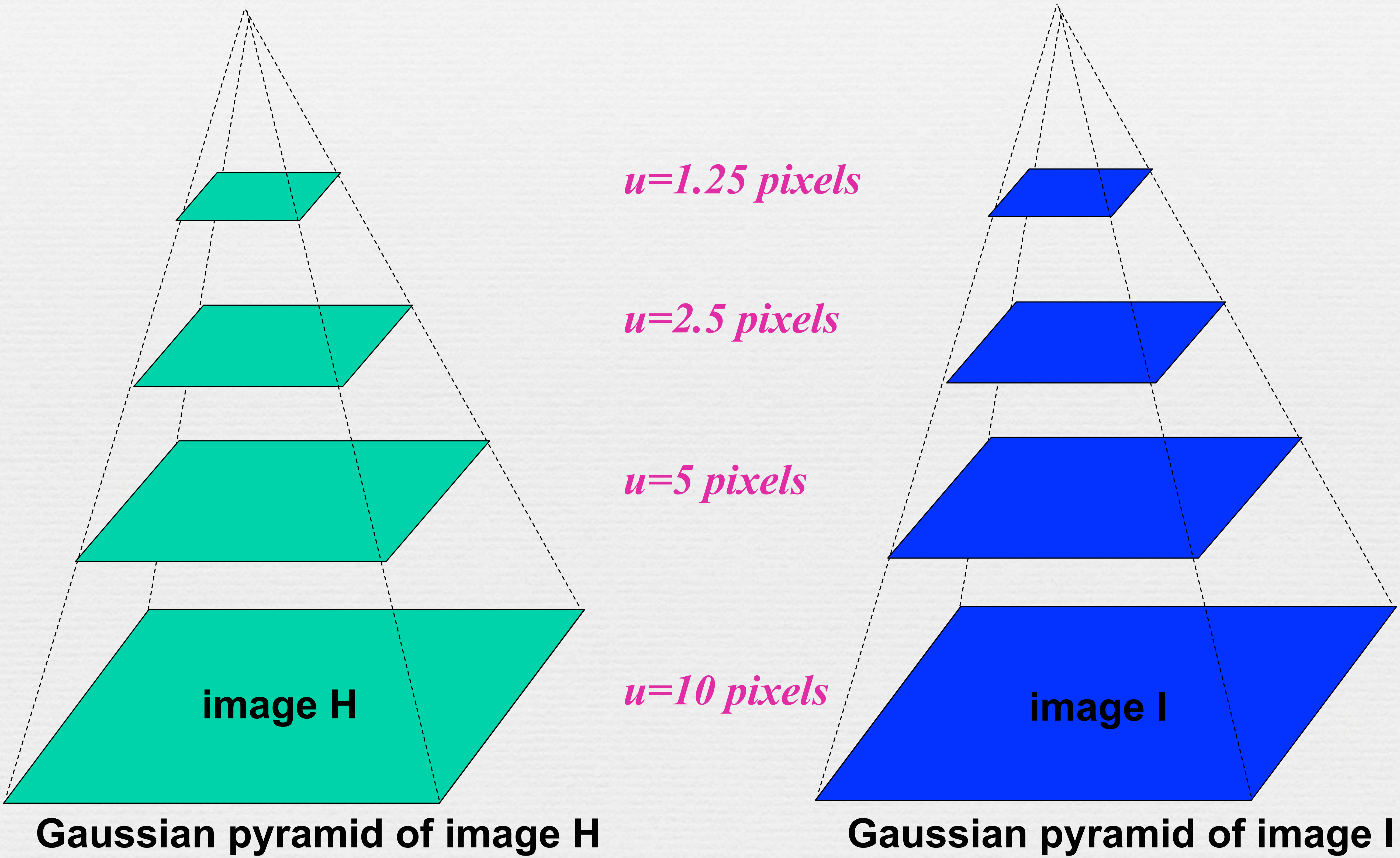


Gaussian pyramid of image H

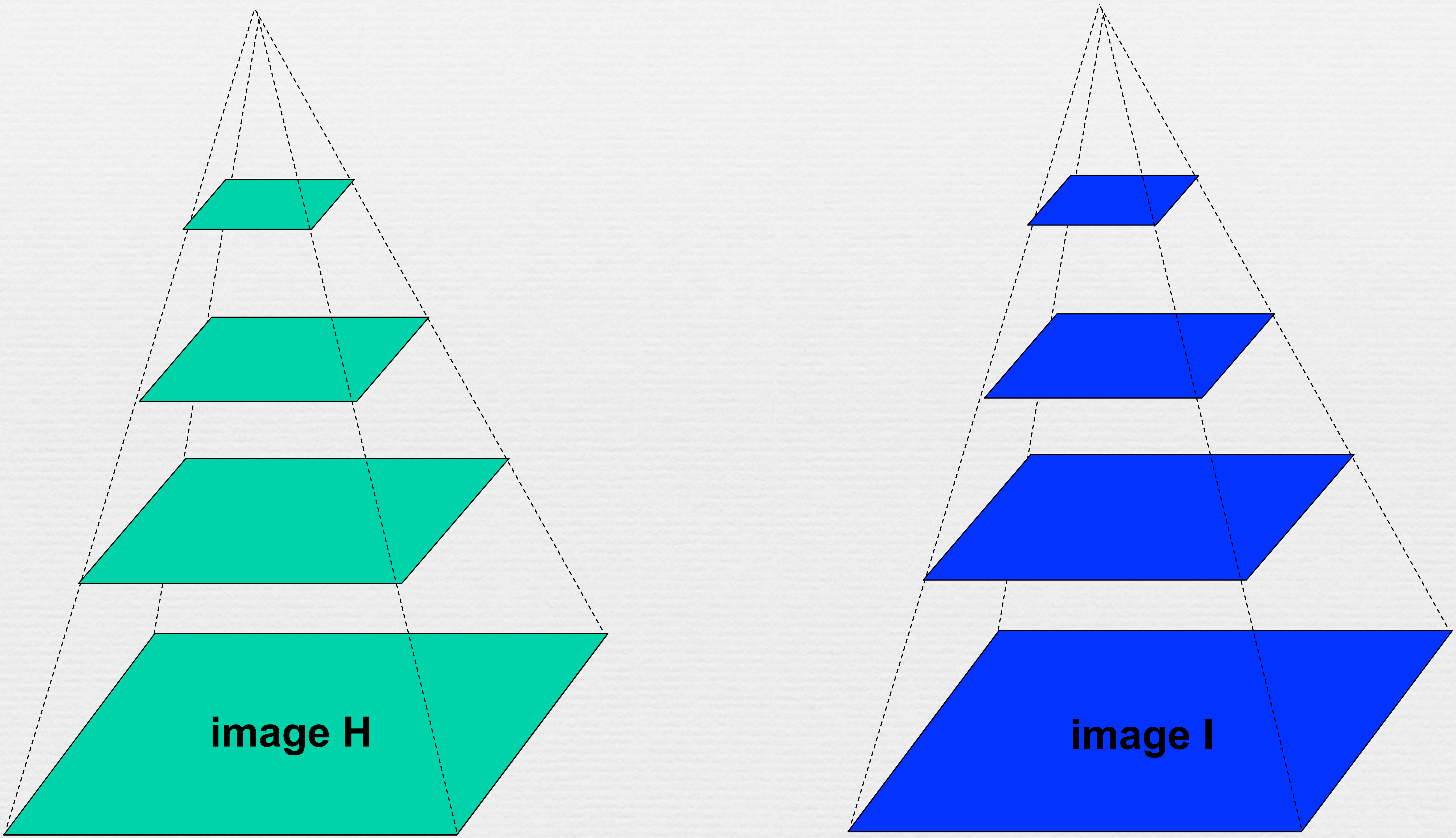


Gaussian pyramid of image I

Coarse-to-fine optical flow

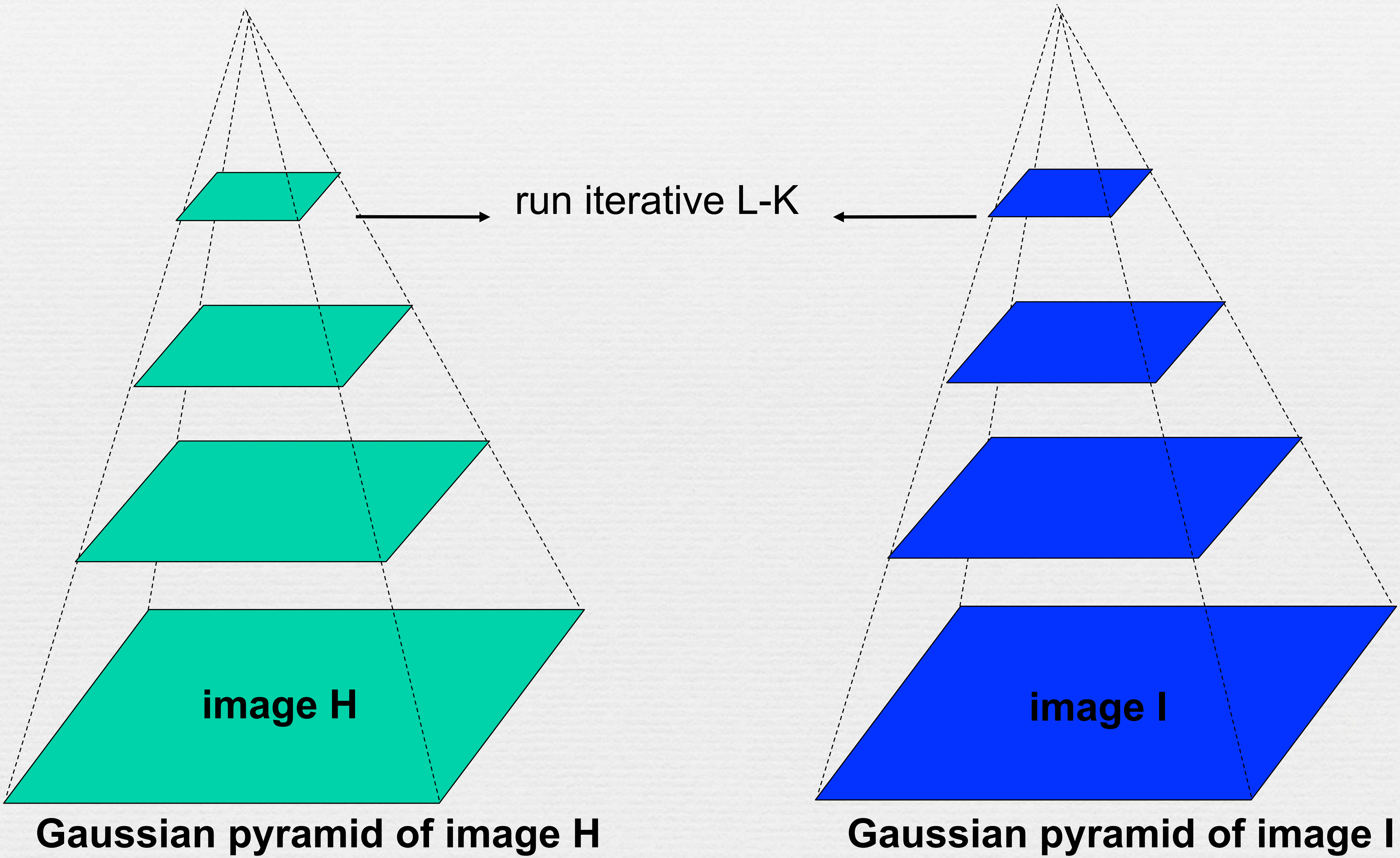


Coarse-to-fine optical flow

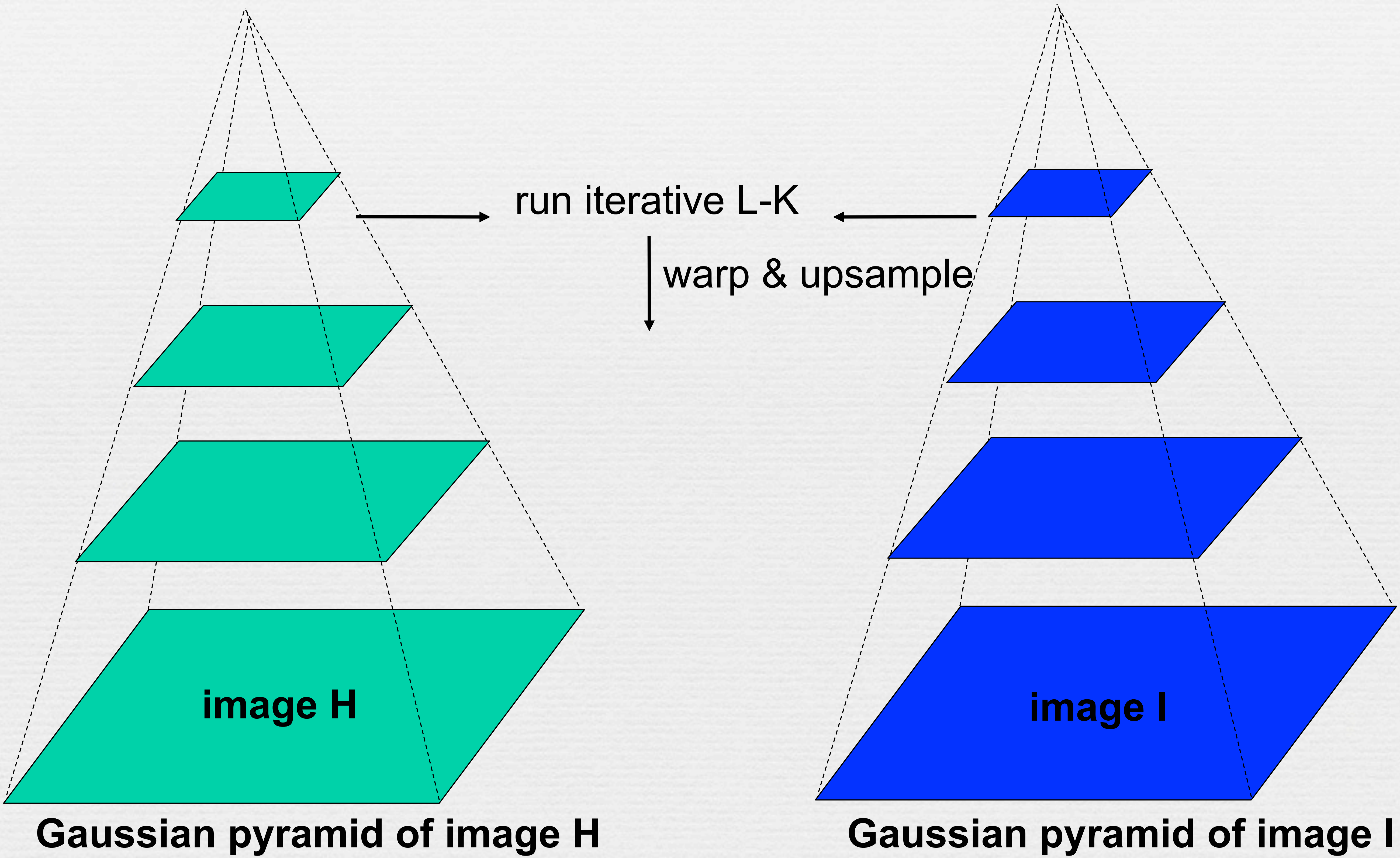


Gaussian pyramid of image H | Gaussian pyramid of image I

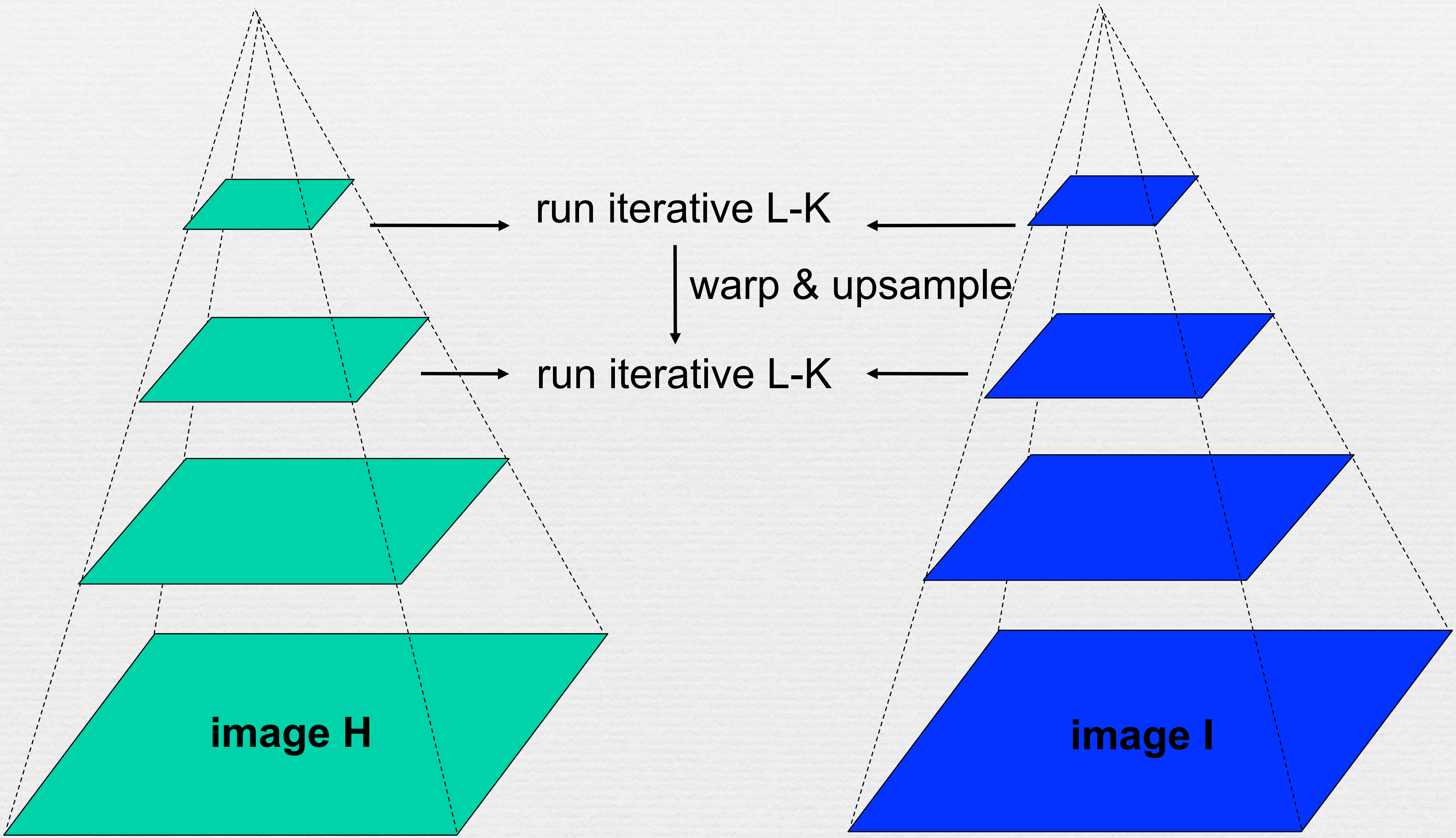
Coarse-to-fine optical flow



Coarse-to-fine optical flow



Coarse-to-fine optical flow



Gaussian pyramid of image H

Gaussian pyramid of image I

Coarse-to-fine optical flow

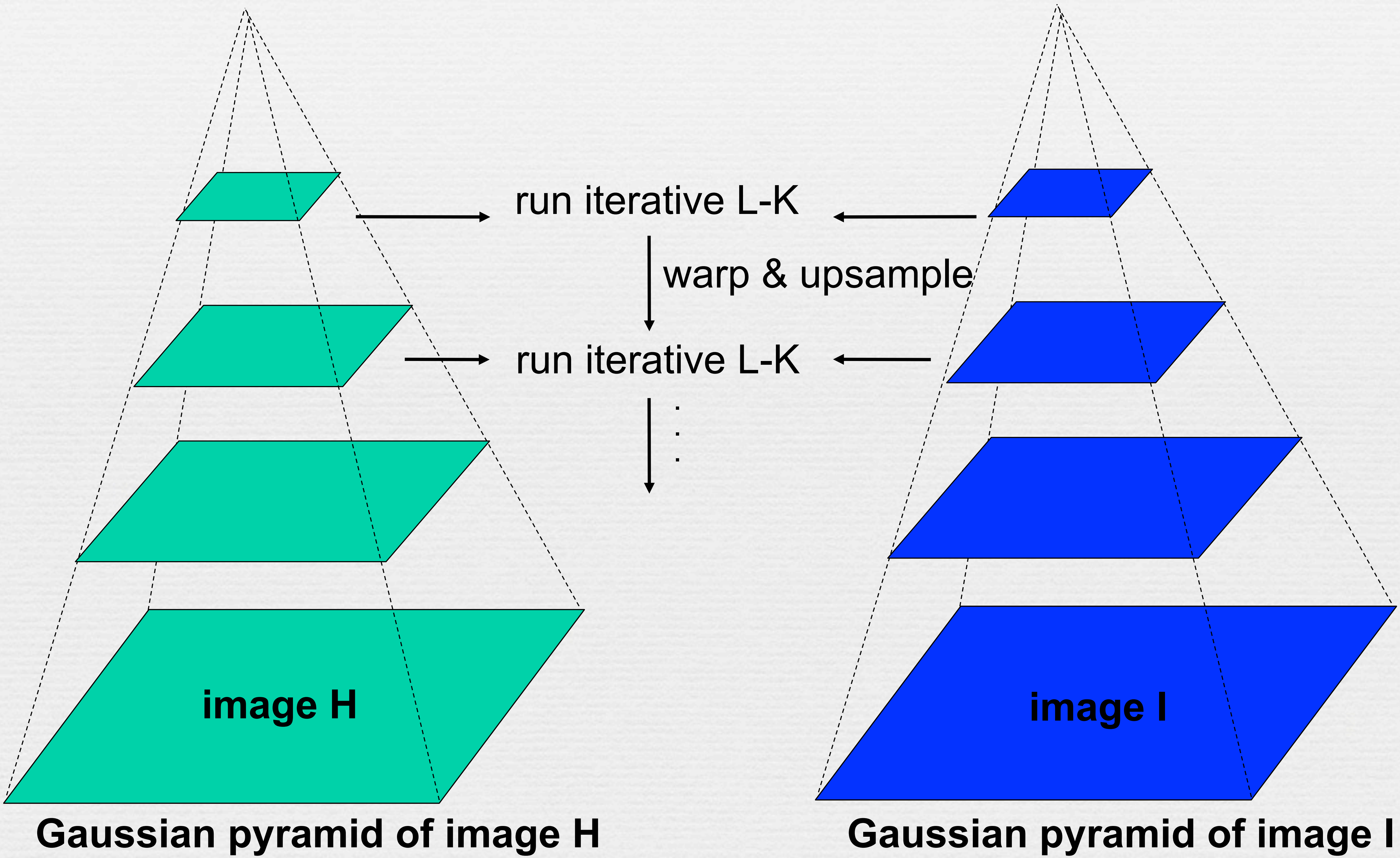


Image alignment: translation

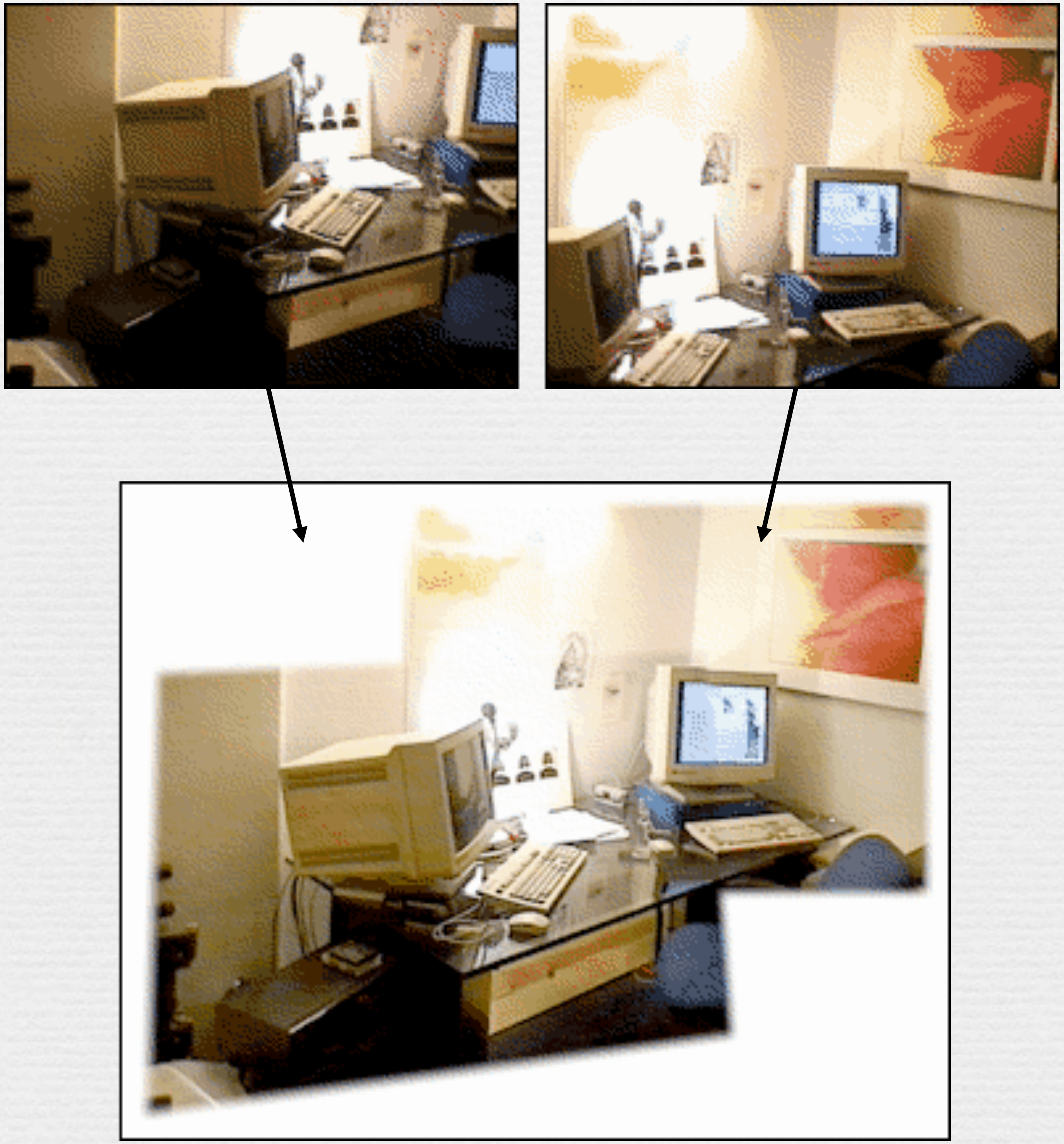
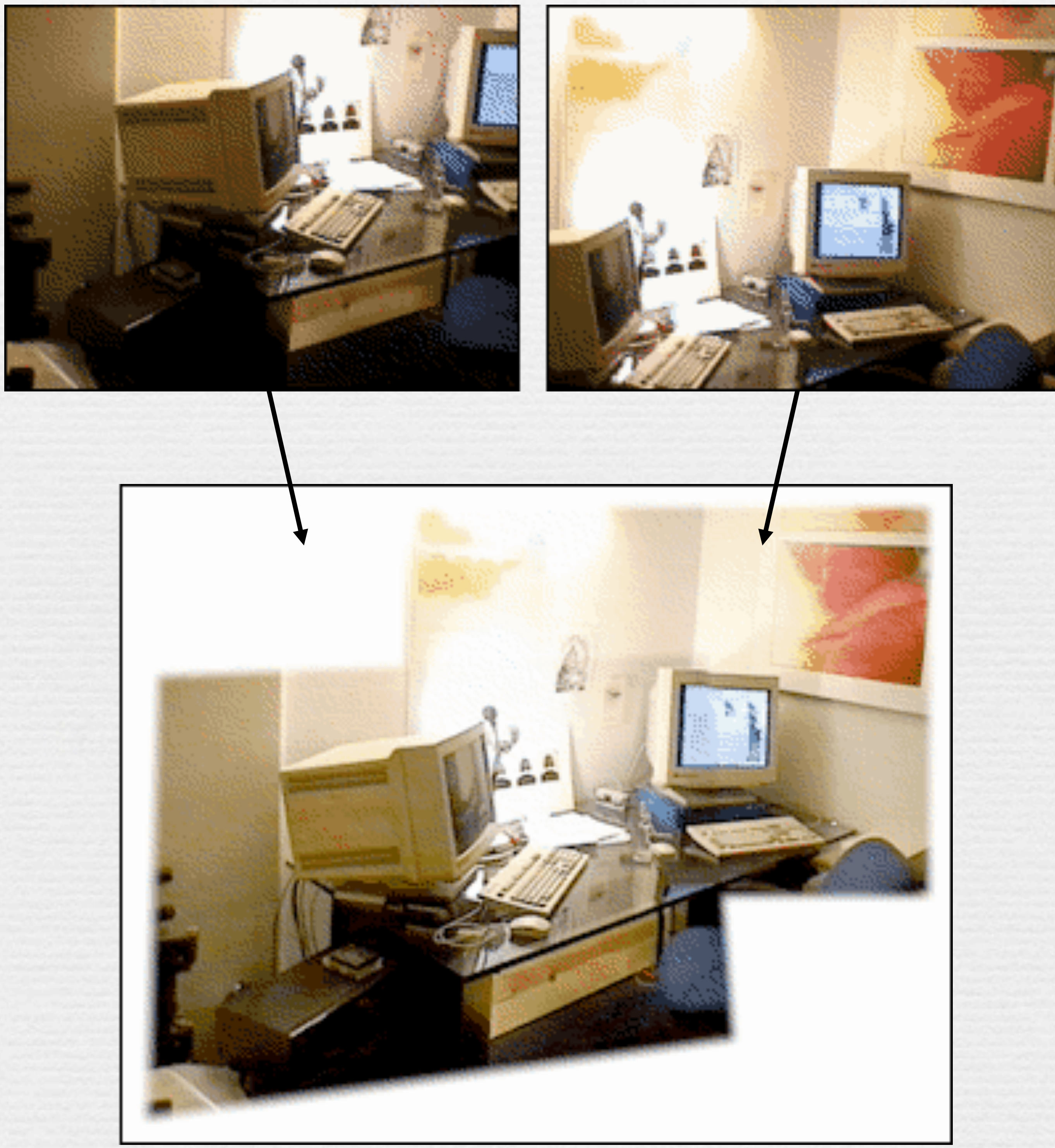


Image alignment: translation



Optical flow for alignment

- Pros:
  - All pixels get used in matching
  - Can get sub-pixel accuracy (important for good mosaicing!)
  - Relatively fast and simple
- Cons:
  - Prone to local minima
  - Images need to be already well-aligned ;-(
- What if, instead, we extract important "features" from the image and just align these?

Other application: retiming

- Generate video at higher frame rate
  - Avoid cross fading artifacts
- Idea: Advect pixels according to optical flow

  http://www.springerlink.com/content/y701u6n1l4j7323m/

frame n

frame n+1

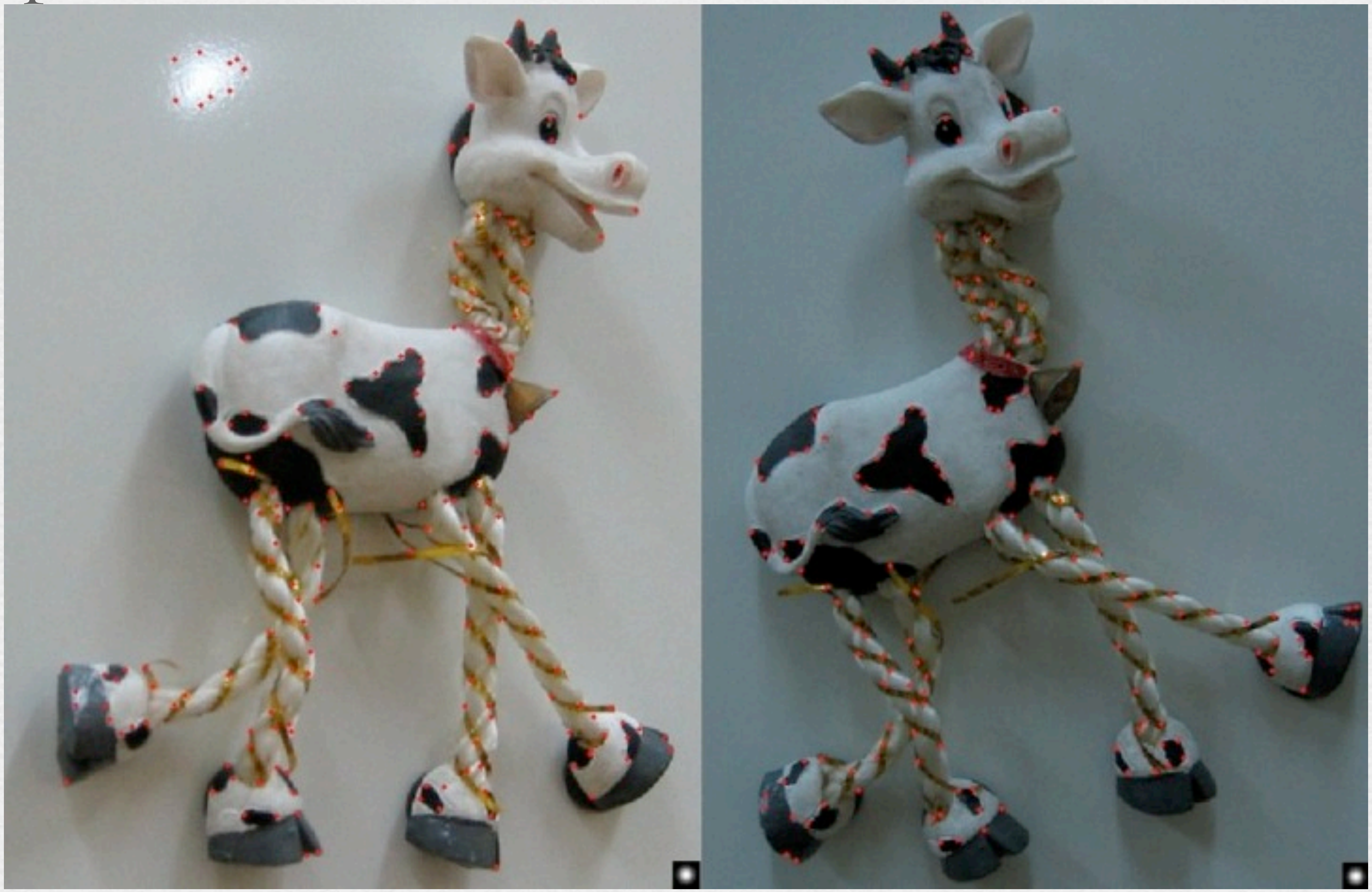
linear interpolation

Optical flow advection

FEATURE TRACKING

Good features to track

- Idea: some pixels are easier to track
  - e.g. corners, because they suffer less from the aperture problem



Condition for solvability

- Optimal (u, v) satisfies Lucas-Kanade equation

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

$$A^T A \qquad\qquad A^T b$$

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] = \sum \nabla I (\nabla I)^T$$

Condition for solvability

- Optimal (u, v) satisfies Lucas-Kanade equation

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

$A^T A$ $\qquad\qquad$ $A^T b$

When is This Solvable?

- $\mathbf{A^T A}$ should be invertible
- $\mathbf{A^T A}$ should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of $\mathbf{A^T A}$ should not be too small
- $\mathbf{A^T A}$ should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] = \sum \nabla I (\nabla I)^T$$

Condition for solvability

- Optimal (u, v) satisfies Lucas-Kanade equation

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

$A^T A$ $\qquad$ $A^T b$

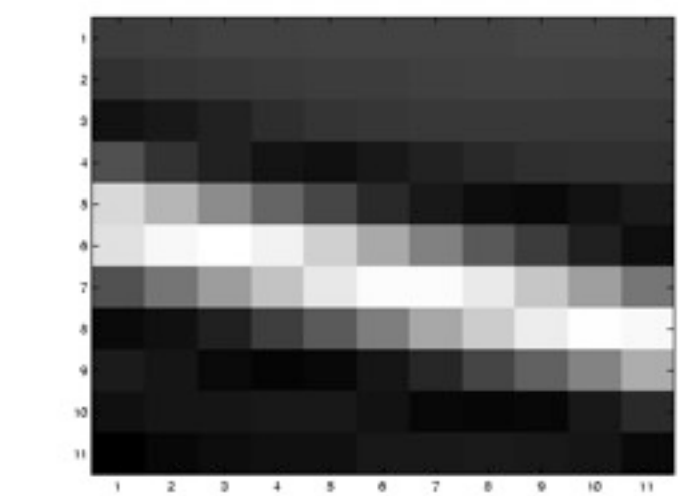
When is This Solvable?

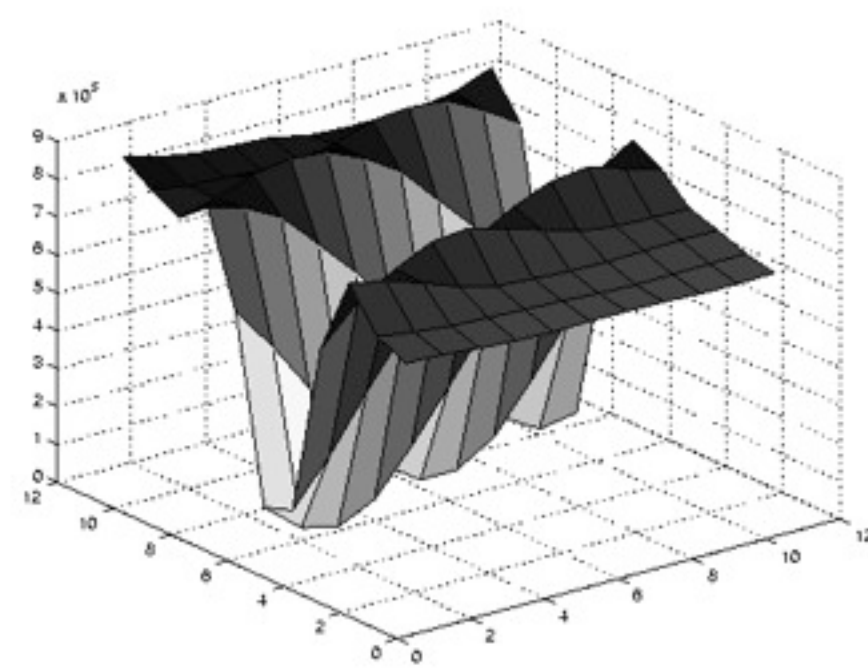
- **A^TA** should be invertible
- **A^TA** should not be too small due to noise
  – eigenvalues λ_1 and λ_2 of **A^TA** should not be too small
- **A^TA** should be well-conditioned
  – λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

A^TA is solvable when there is no aperture problem

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

Edge

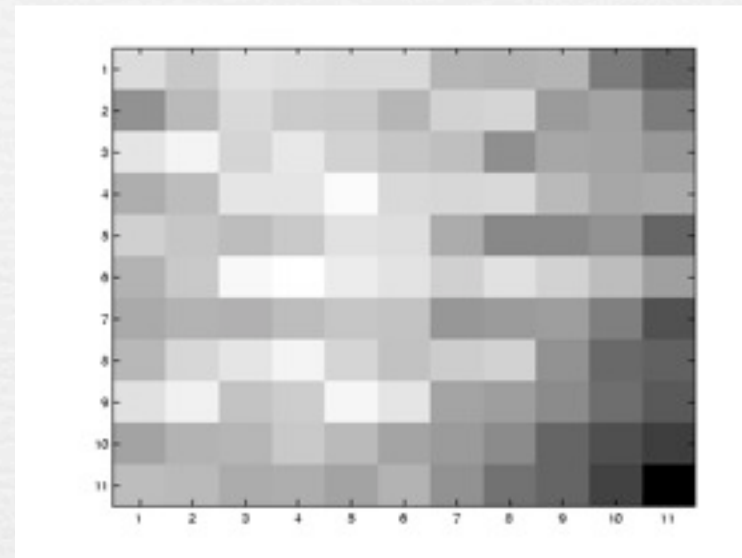


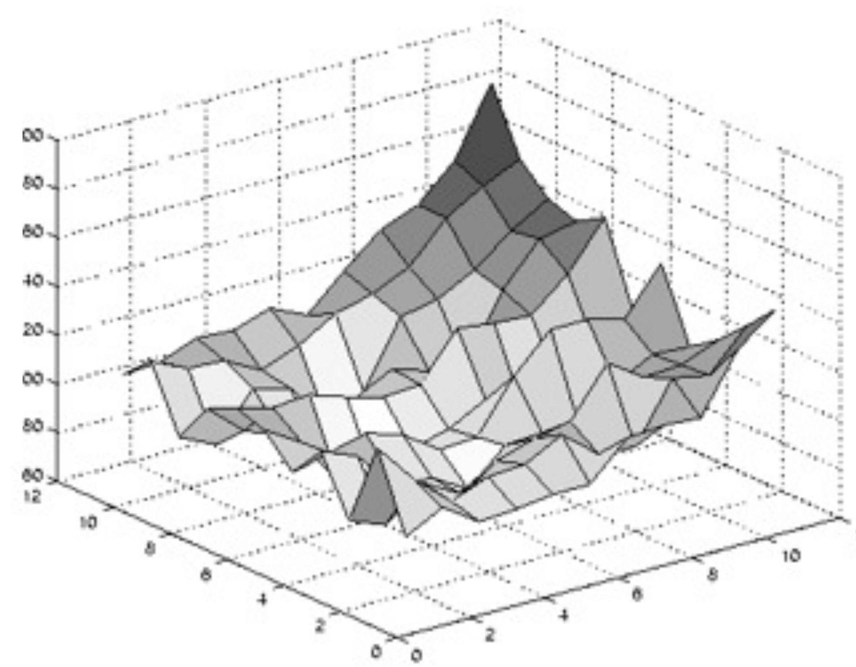


$$\sum \nabla I (\nabla I)^T$$

- large gradients, all the same
- large λ_1, small λ_2

Low texture region

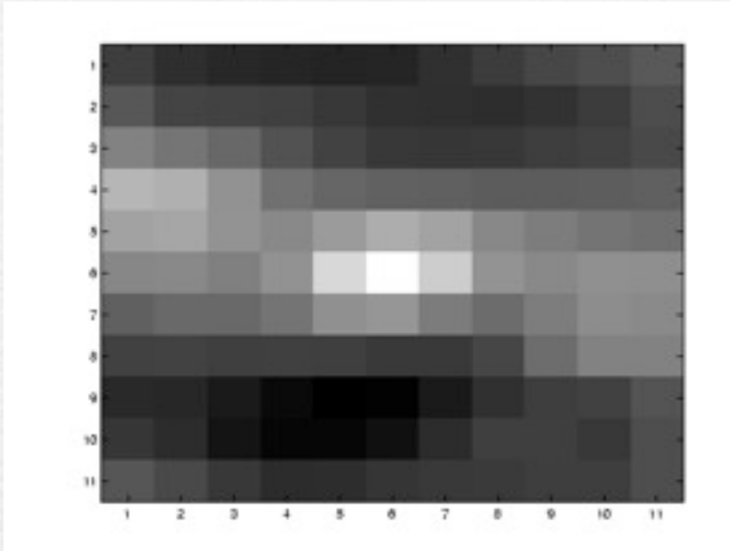


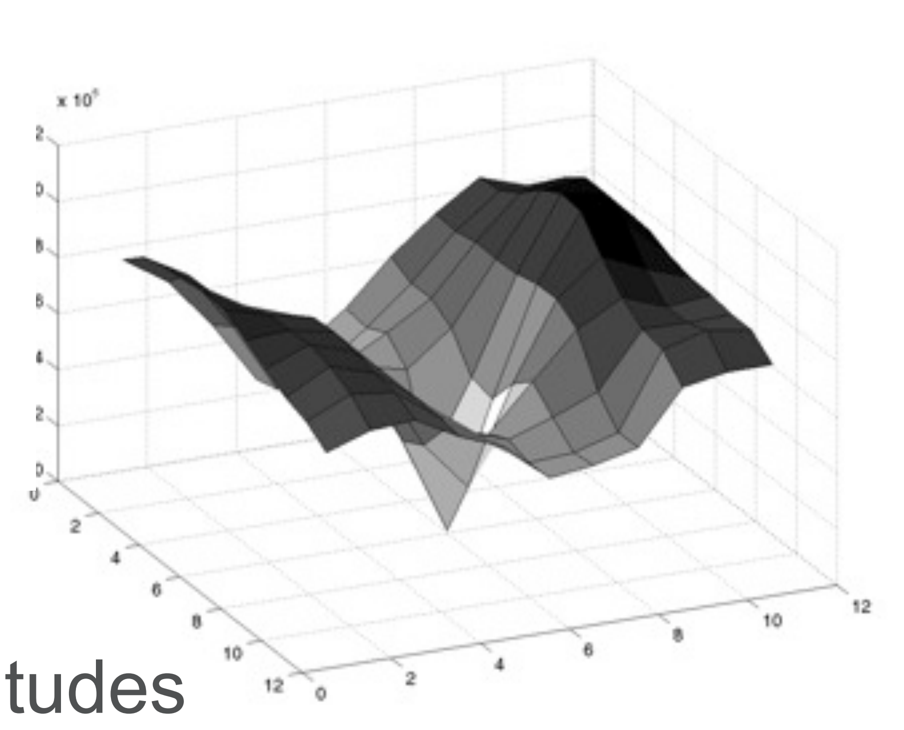


$$\sum \nabla I (\nabla I)^T$$

- gradients have small magnitude
- small λ_1, small λ_2

High textured region





$\sum \nabla I (\nabla I)^T$

- gradients are different, large magnitudes
- large λ_1, large λ_2

Harris Detector

- Average intensity change in direction $[u,v]$ can be expressed as a bilinear Taylor form:

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

- Describe a point in terms of eigenvalues of M: *measure of corner response*

$$R = \lambda_1\lambda_2 - k\left(\lambda_1 + \lambda_2\right)^2$$

- A good (corner) point should have a *large intensity change* in *all directions*, i.e. R should be large positive

- Variation: Shi-Tomasi: Pretty much same as Harris, but use min(λ1,λ2) instead of R

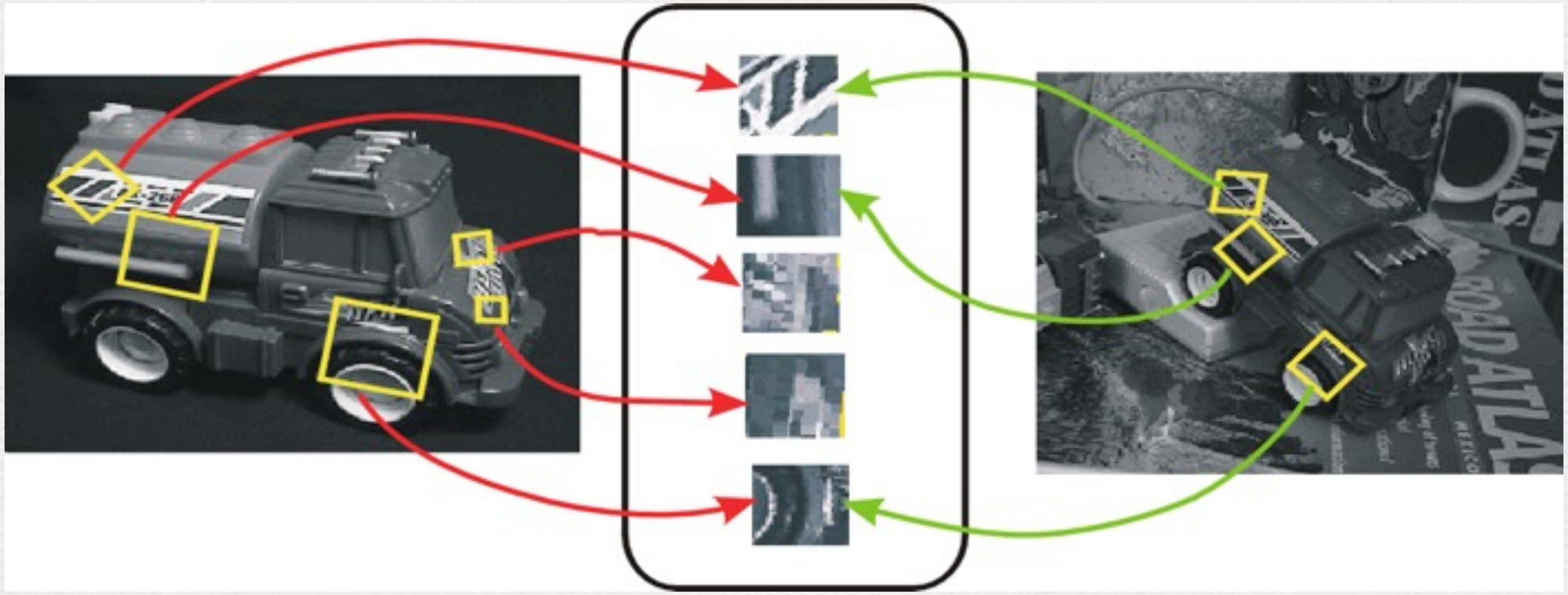
Recap

- Brute force : dense, model based
- Optical flow: dense, non-parametric
  - nearby pixels, same color
  - uses all pixels
  - can be unstable
- Feature tracking: sparse, non-parametric
  - find corners
  - then apply optical flow to them

  - problem: what if the motion is too big?

Rich feature descriptors

- e.g. SIFT, Schmid & Mohr 1997, Lowe 1999, Baumberg 2000, Tuytelaars & Van Gool 2000, Mikolajczyk & Schmid 2001, Brown & Lowe 2002, Matas et. al. 2002, Schaffalitzky & Zisserman 2002
- Detect points of interest
- Associate rich descriptor of patch (histogram of gradient in 4x4 subwindows)
  - Can be matched across images



MODEL FITTING

Fitting a model

- Often, we want to infer a simple low-order motion model
  - e.g. translation, affine, projective
  - because we know the motion
  - or to regularize (get a smoother estimate)
- How do we do, given a number of correspondences or flow vectors?

Fitting a model

- e.g. affine:
  $x' = ax + by + c$
  $y' = dx + ey + f$
- Find a, b, c, d, e, f given a number of pairs (x', y'), (x, y)
- Simple linear least squares: two equations per pair of 2D points, need at least 3 points.

Robustness to bad matches

- RANSAC
  (RANdom SAmple Consensus)
  - Fit a model with random subset of correspondences
  - Count how many correspondences it matches
  - Iterate
- Reweighted least squares
  - Fit model with least square
  - Reweight correspondences based on how close they are from their predicted new location
  - Iterate

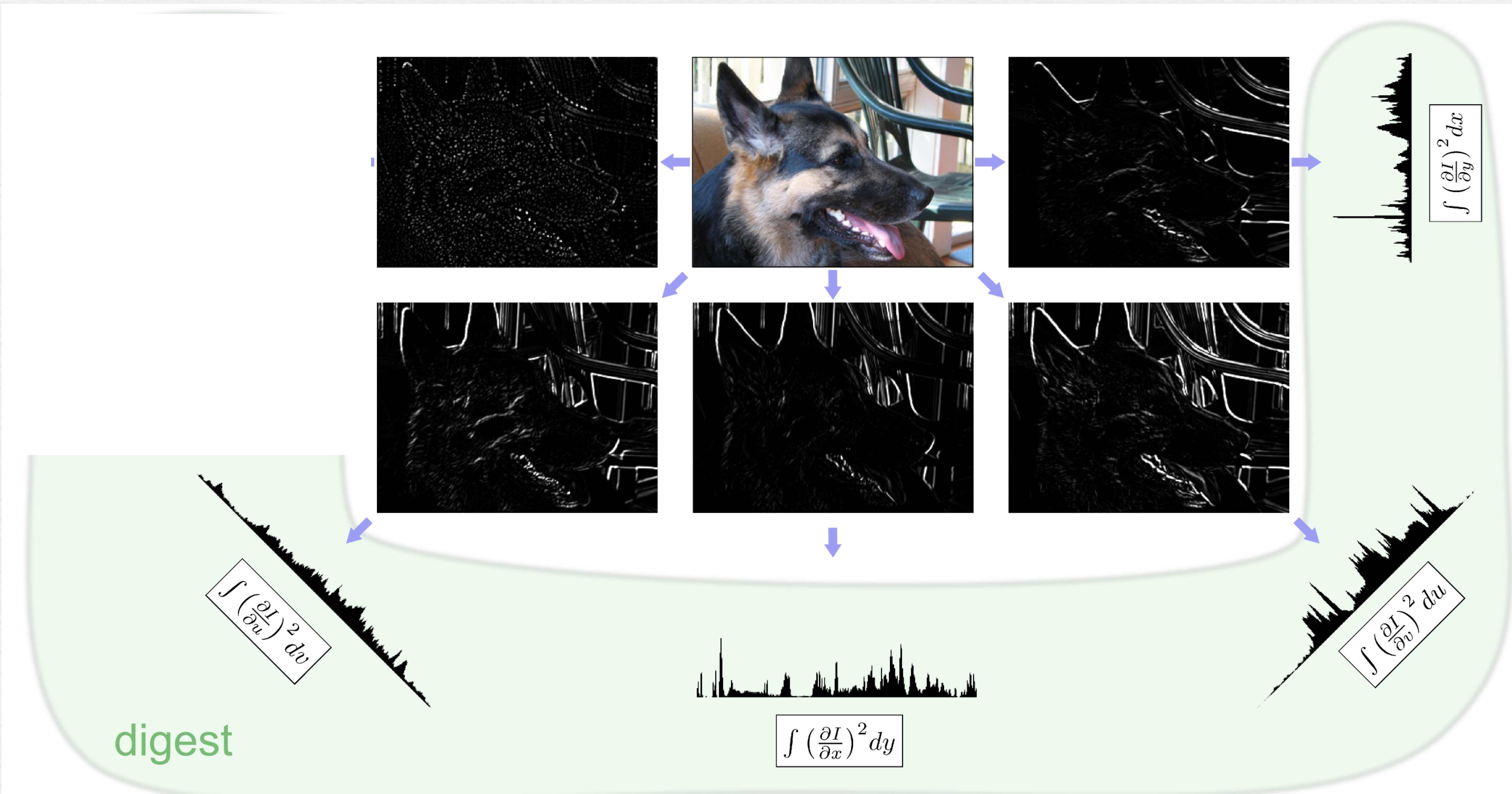
VIEWFINDER ALIGNMENT

Challenge: real time on cell phone

- Viewfinder Alignment. Andrew Adams, Natasha Gelfand, Kari Pulli, Eurographics 2008
- http://graphics.stanford.edu/papers/viewfinderalignment/

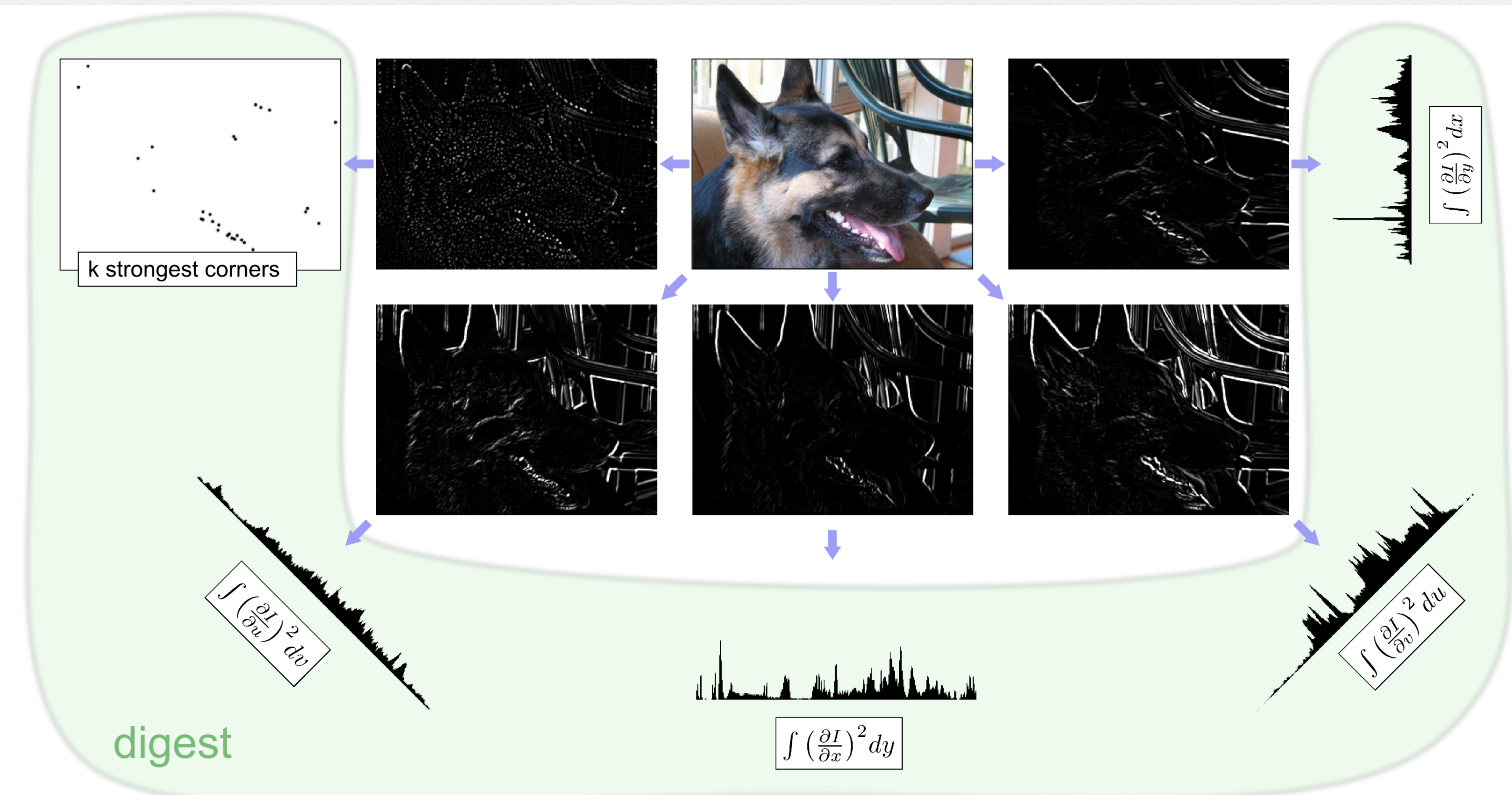
Idea: 1D matches of gradient

- Compute and project gradient along 4 directions
- Brute force search for 1D translations



Idea: 1D matches of gradient

- Compute and project gradient along 4 directions
- Also extract strong corners for rotation inference



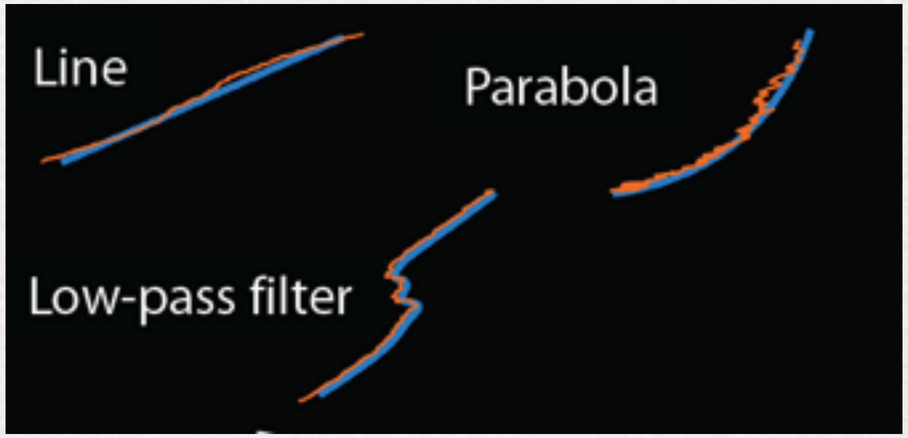
Application: denoising

VIDEO STABILIZATION

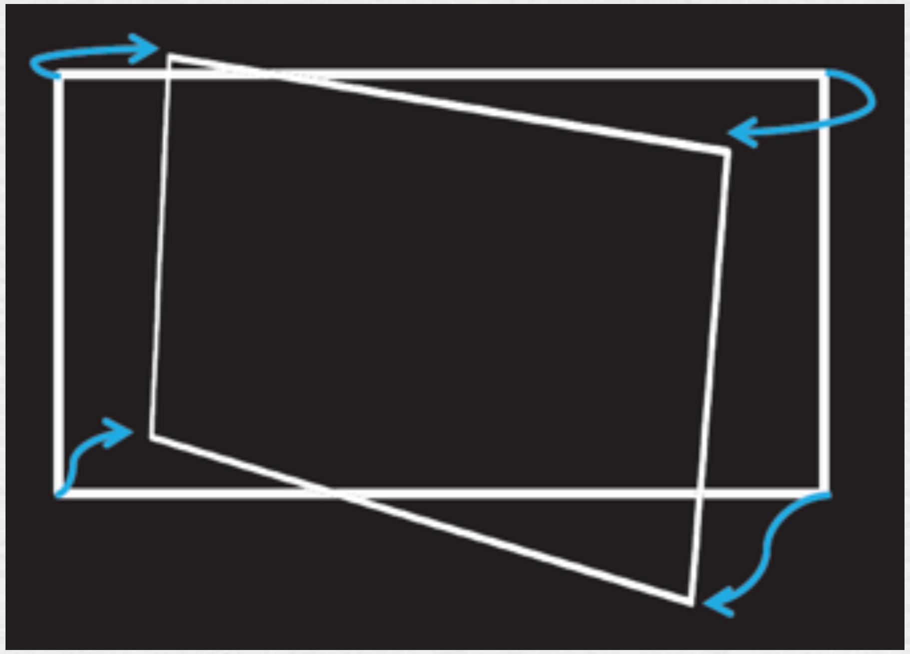
Video stabilization : 3 steps

- Estimate motion
  - local motion vector
  - Fit per-frame global motion

- Smooth motion temporally
  - e.g. low pass or model fitting

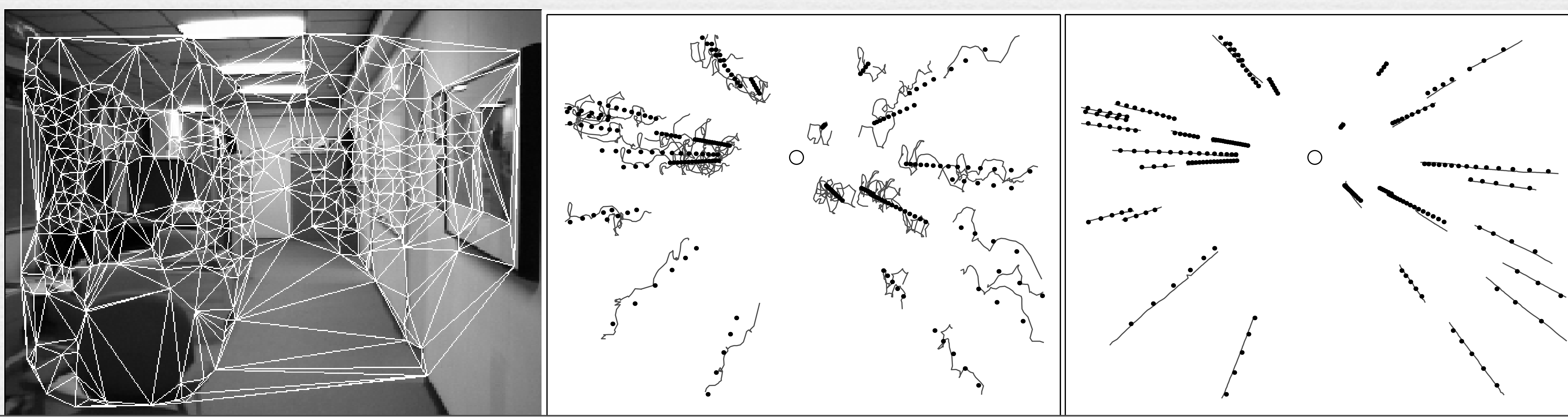


- Warp frames



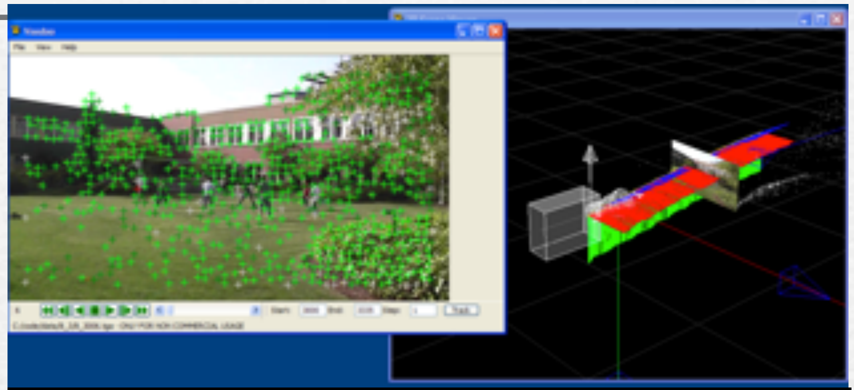
More advanced: 3D motion

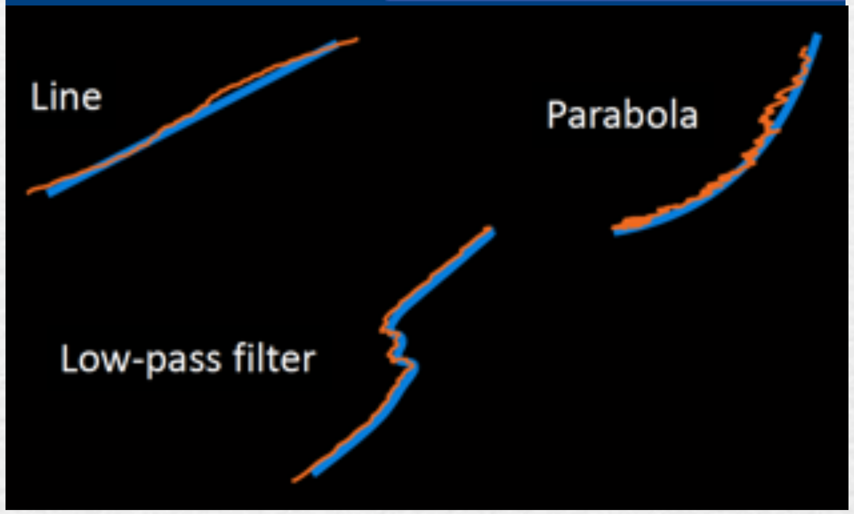
- [Buehler et al. CVPR 2001]
- Given correspondences and assuming a rigid object, estimate camera pose & 3D coordinates
- Smooth 3D motion for more realistic stabilization
- Triangulate features and warp individual triangles
- Maybe use other frames to fill in missing info

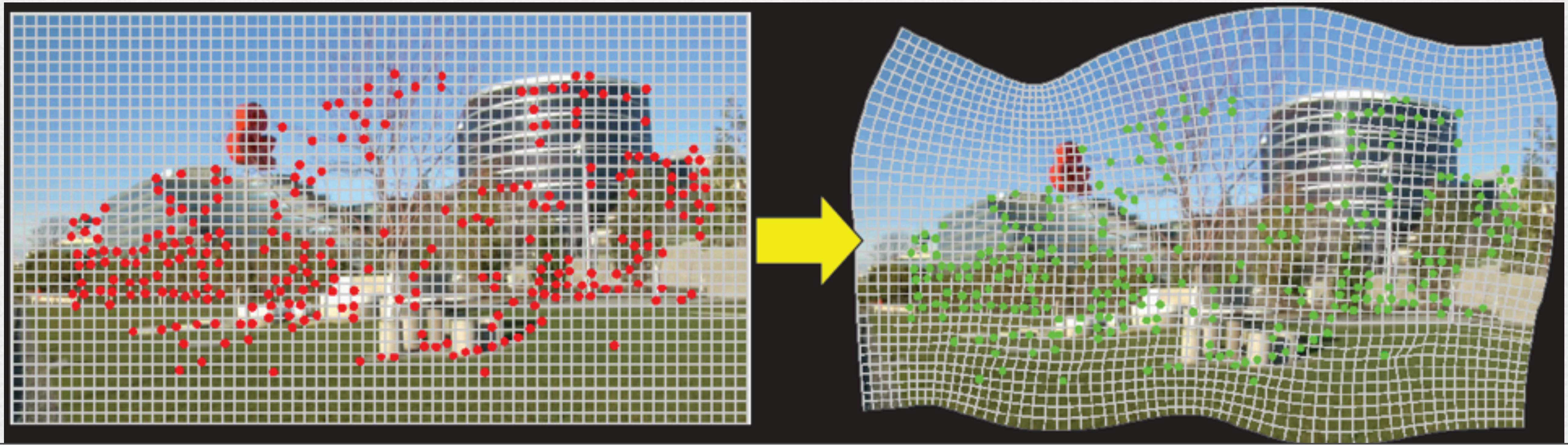


Content-Preserving Warps

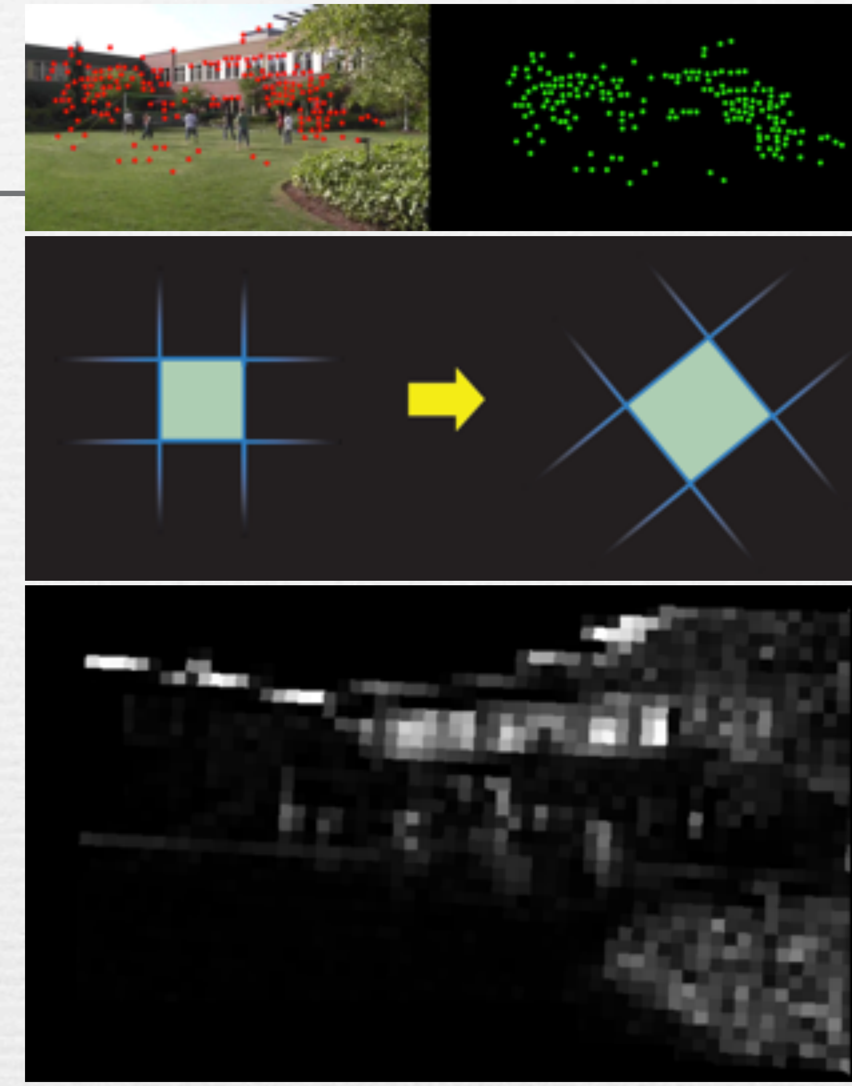
- Liu et al. SIGGRAPH 2009
  http://pages.cs.wisc.edu/~fliu/project/3dstab.htm
- Extract camera path &
  3D feature coord.
- Smooth 3D motion
- Content-preserving warp



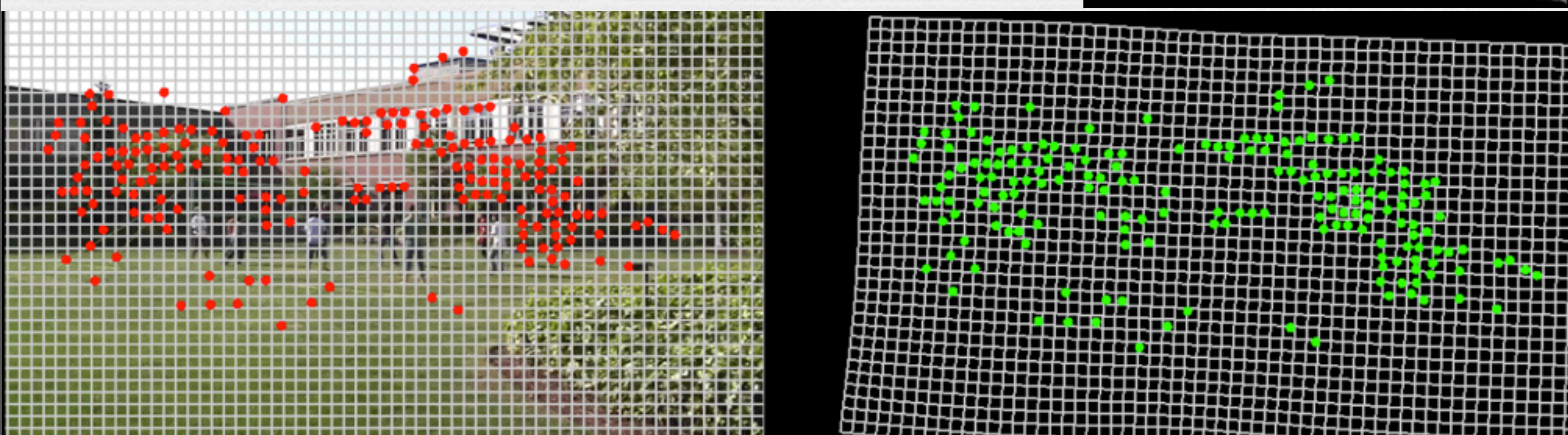




Content-Preserving Warps

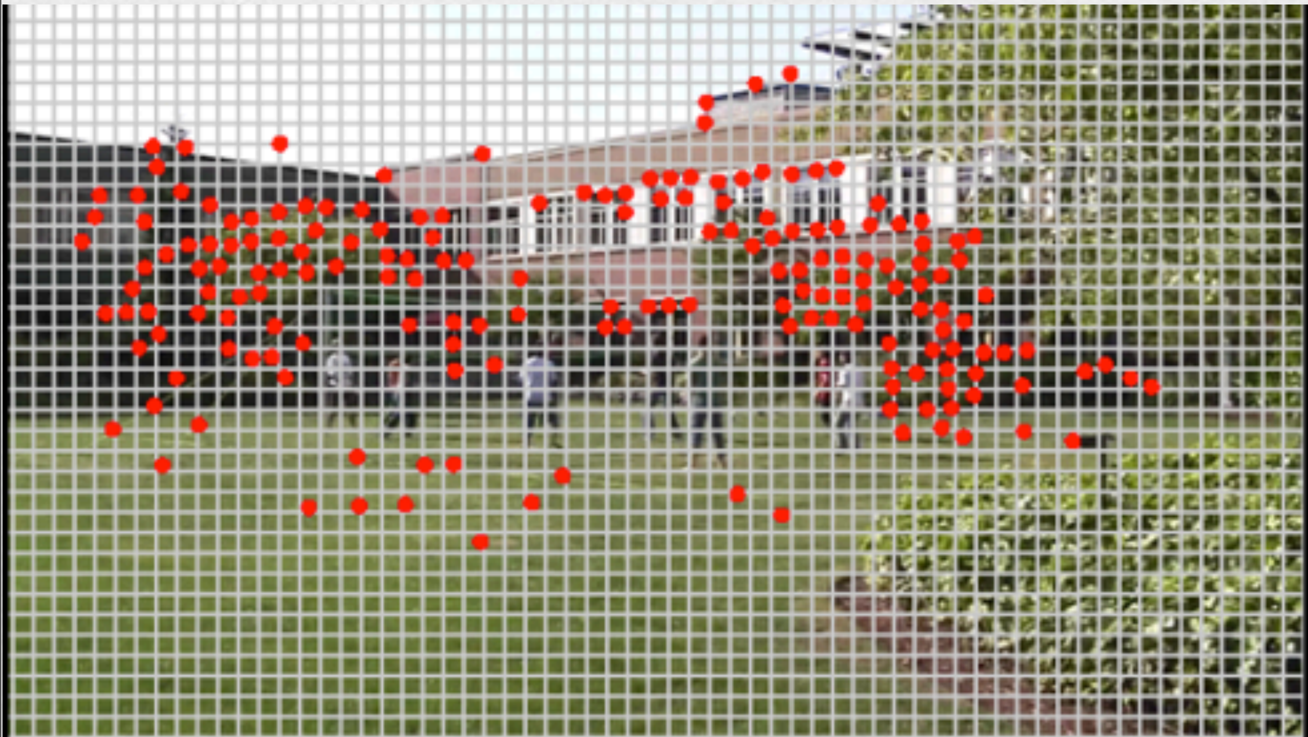


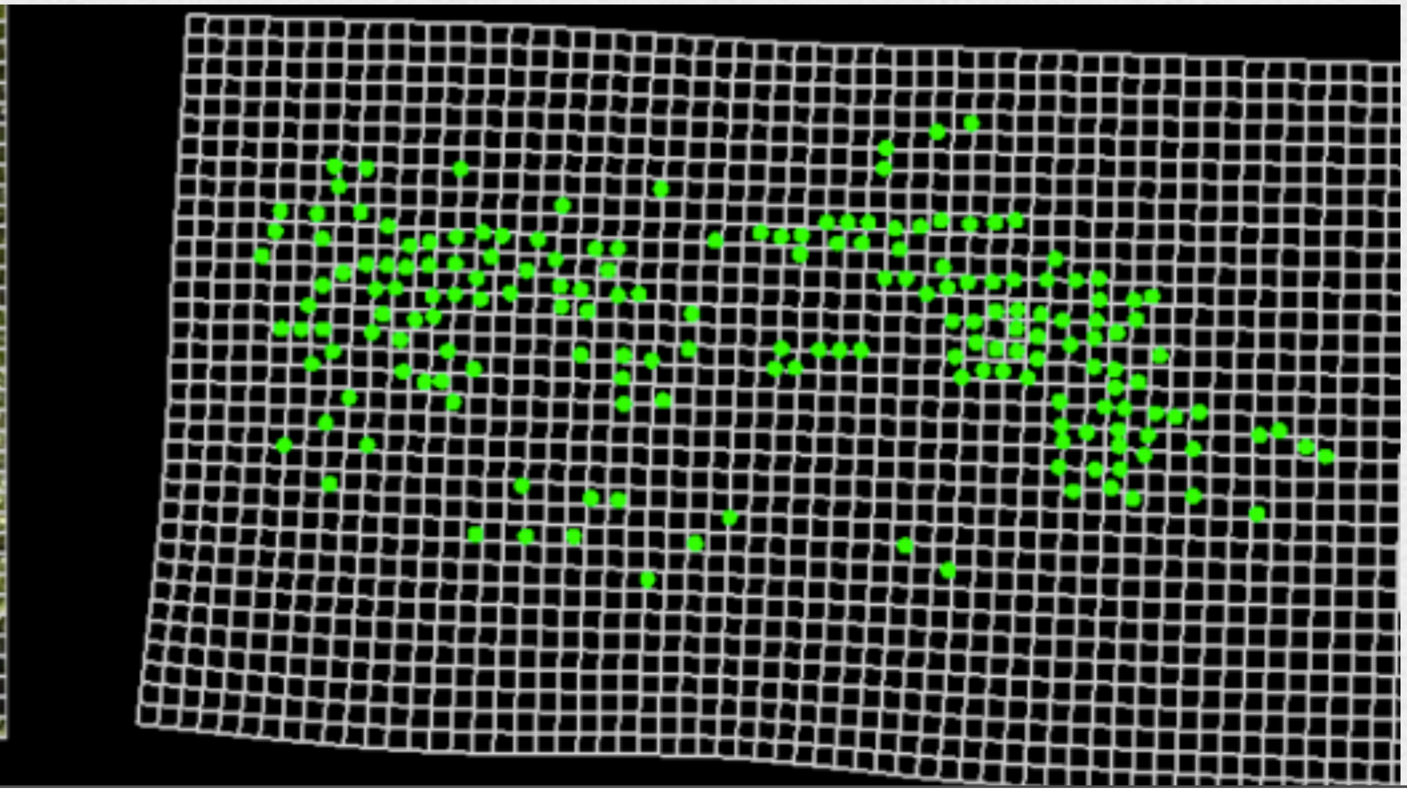
- Use smoothed feature locations as constraints
- Preserve local aspect ratios (conformal mapping)
- Preserve more in salient regions



Solving for warp

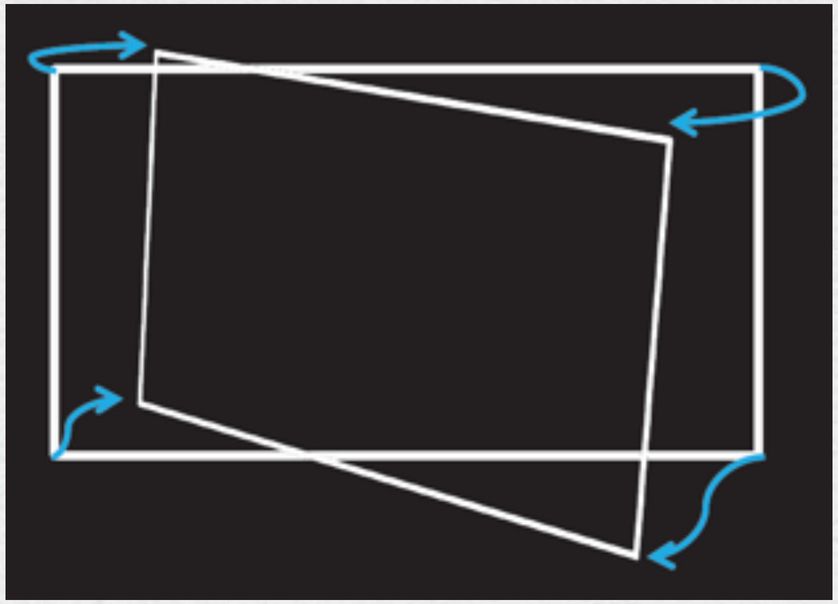
- Grid over image, solve for coordinate of vertices
- Least square minimization
- Data term: feature location
- Smoothness term: local similarity (conformal)

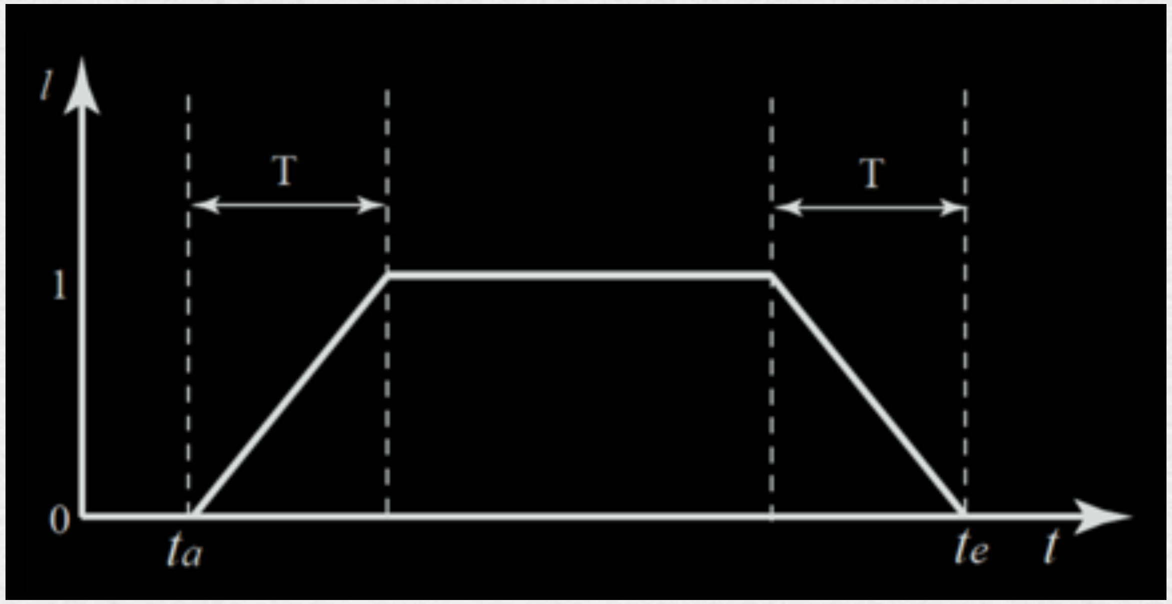




Bells and whistles

- Global projective (homography) pre-warp
  - to take care of most of the job
- Cross-fade the influence of feature points
  - Because they appear and disappear.





Results

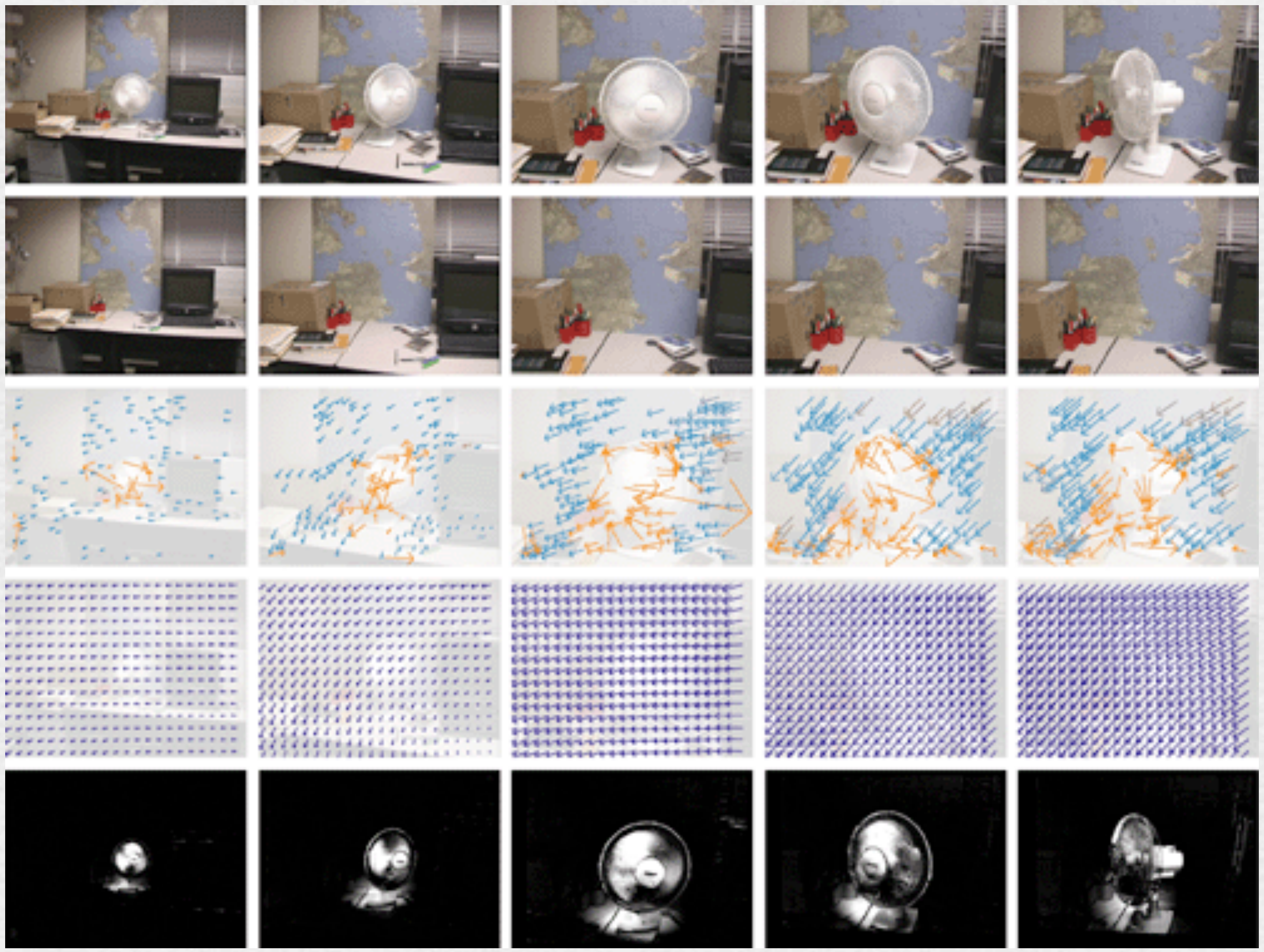
Results

Video

VIDEO MATCHING

Video matching

- Sand and Teller SIGGRAPH 2004
  http://rvsn.csail.mit.edu/vid-match/
- Robust to scene changes, timing change



MATCH MOVE

Match move

- For compositing with moving camera
- Given video sequence, deduce 3D camera motion
- Match with computer graphics camera, miniature camera, etc.

http://www.digilab.uni-hannover.de/docs/manual.html#overview

Example: music video by P. Sand

- Compositing of live action into miniature
- Match camera motion
- http://peter-sand.org/

Live action

- Note orange balls to create good features
- Green screen for compositing

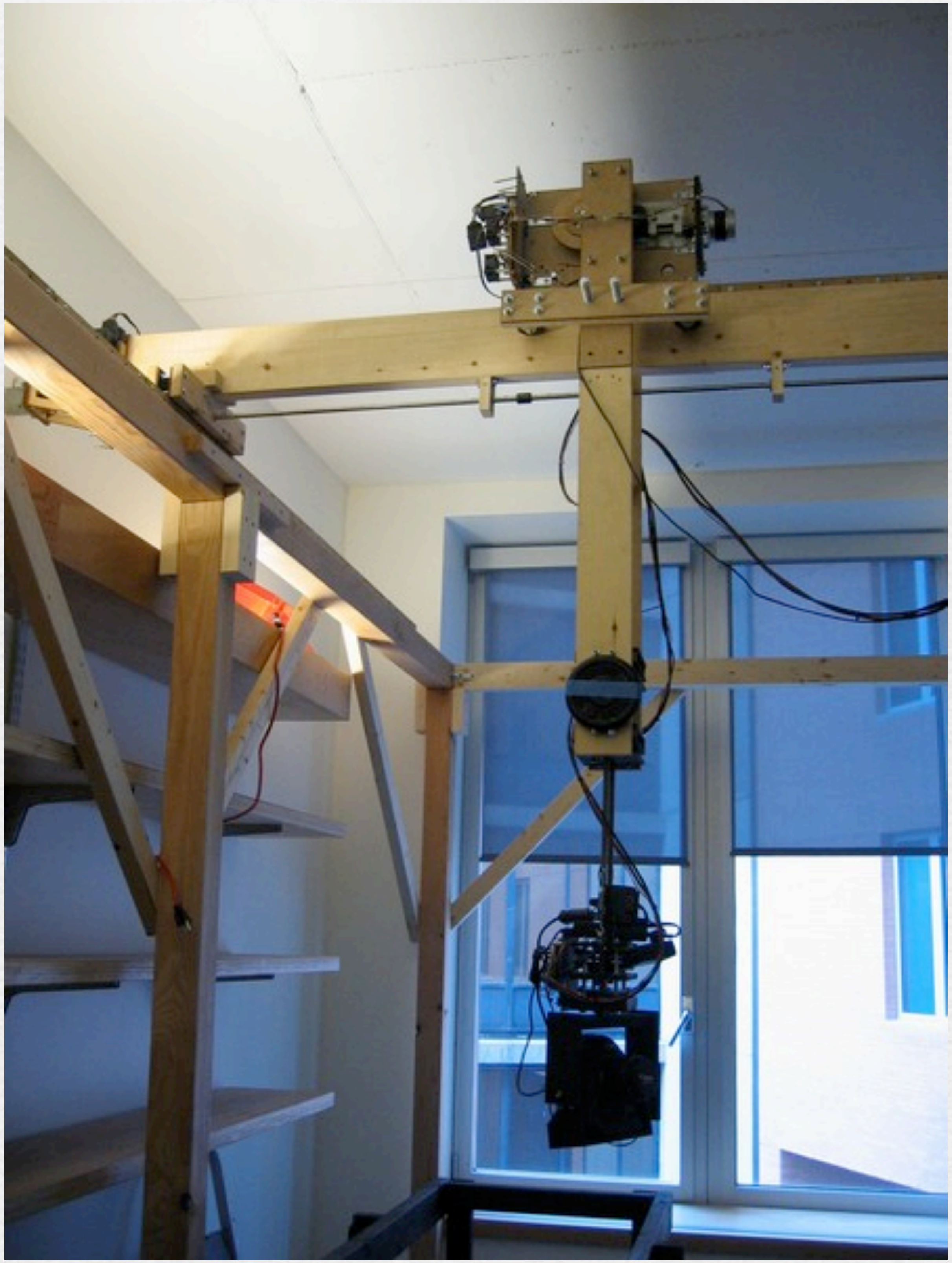
Live action

Miniature construction

Miniature

- Note the big camera (DSLR)

5 degrees of freedom camera robot

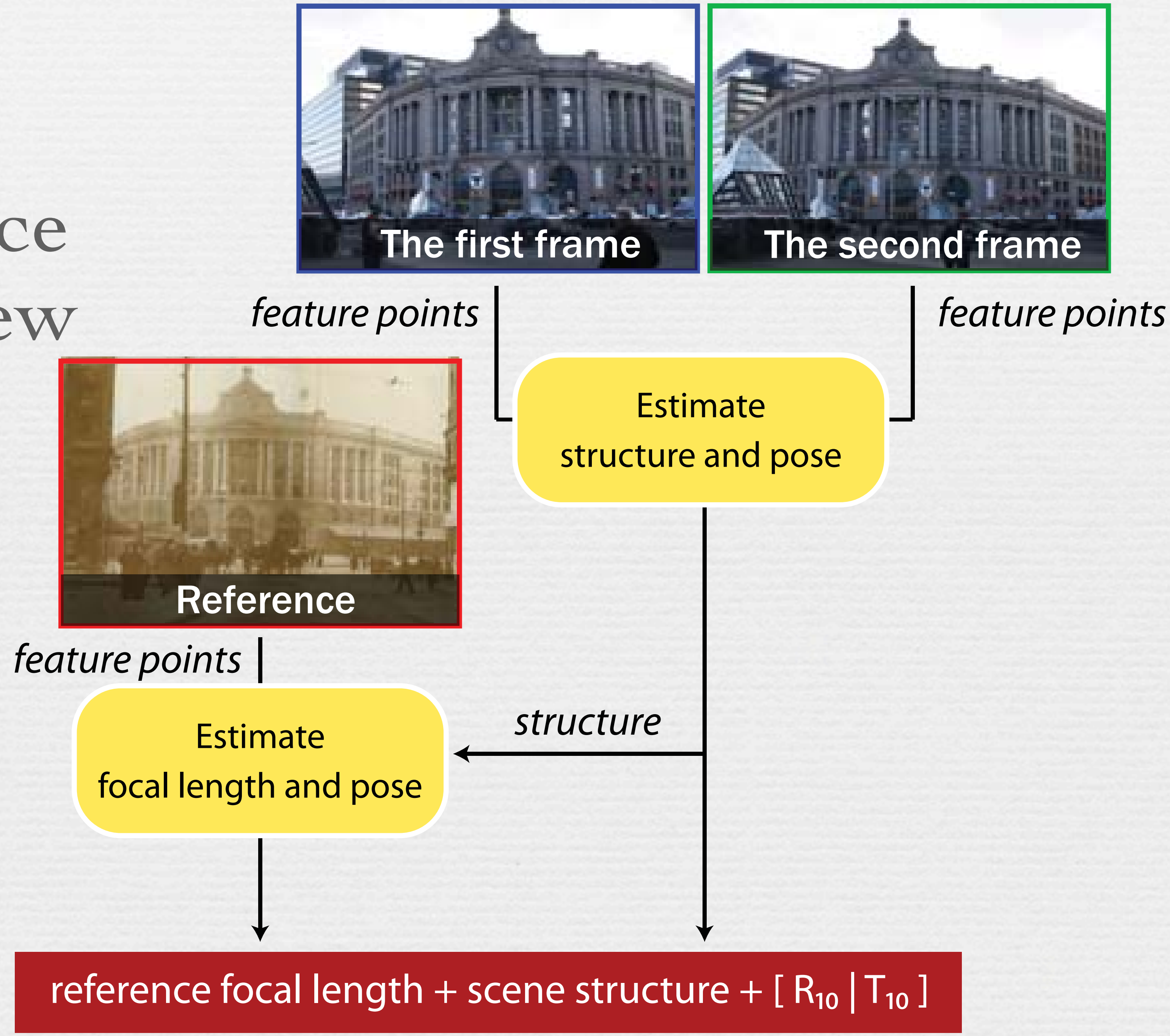


Video

RE-PHOTOGRAPHY

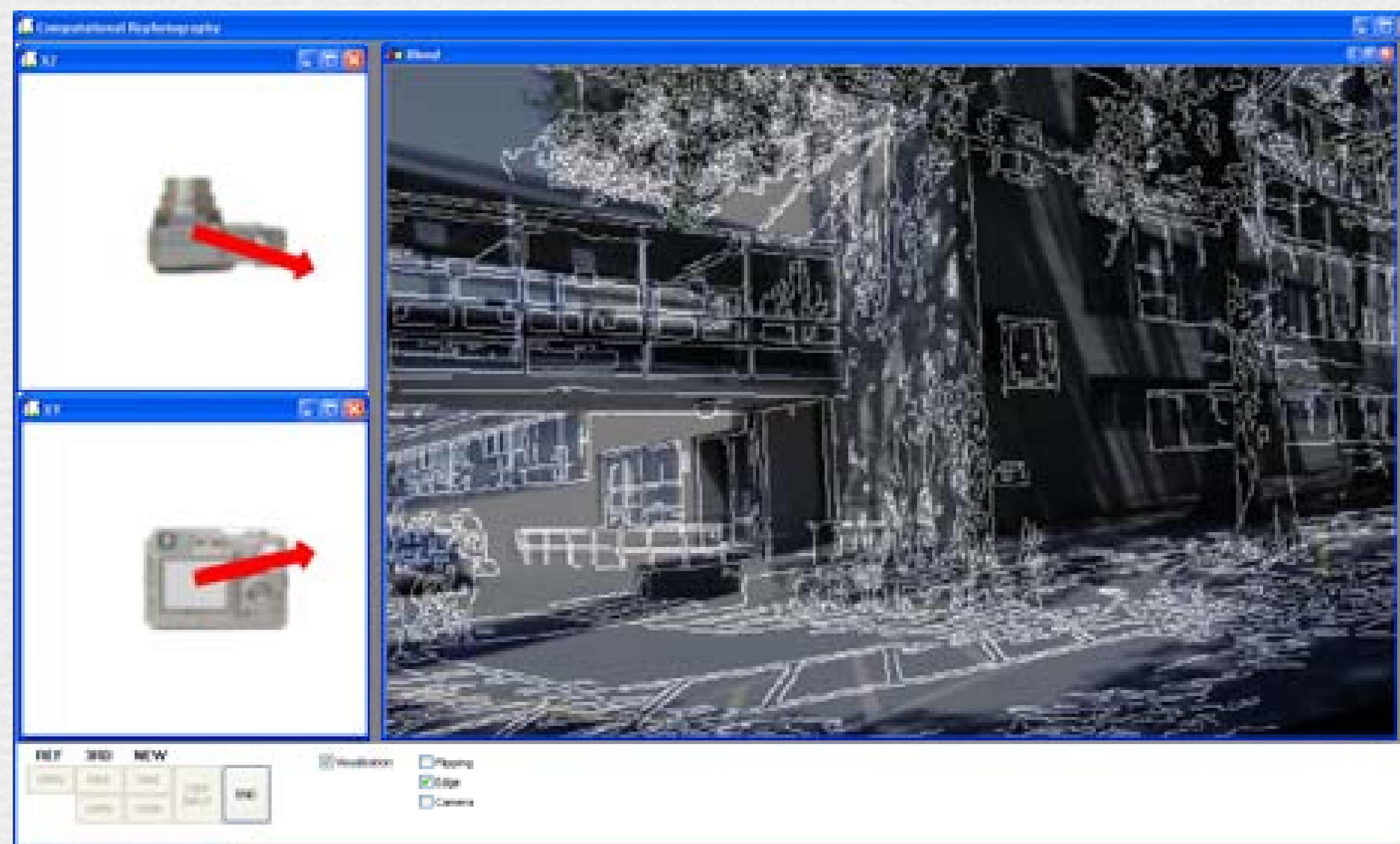
Computational Re-Photography

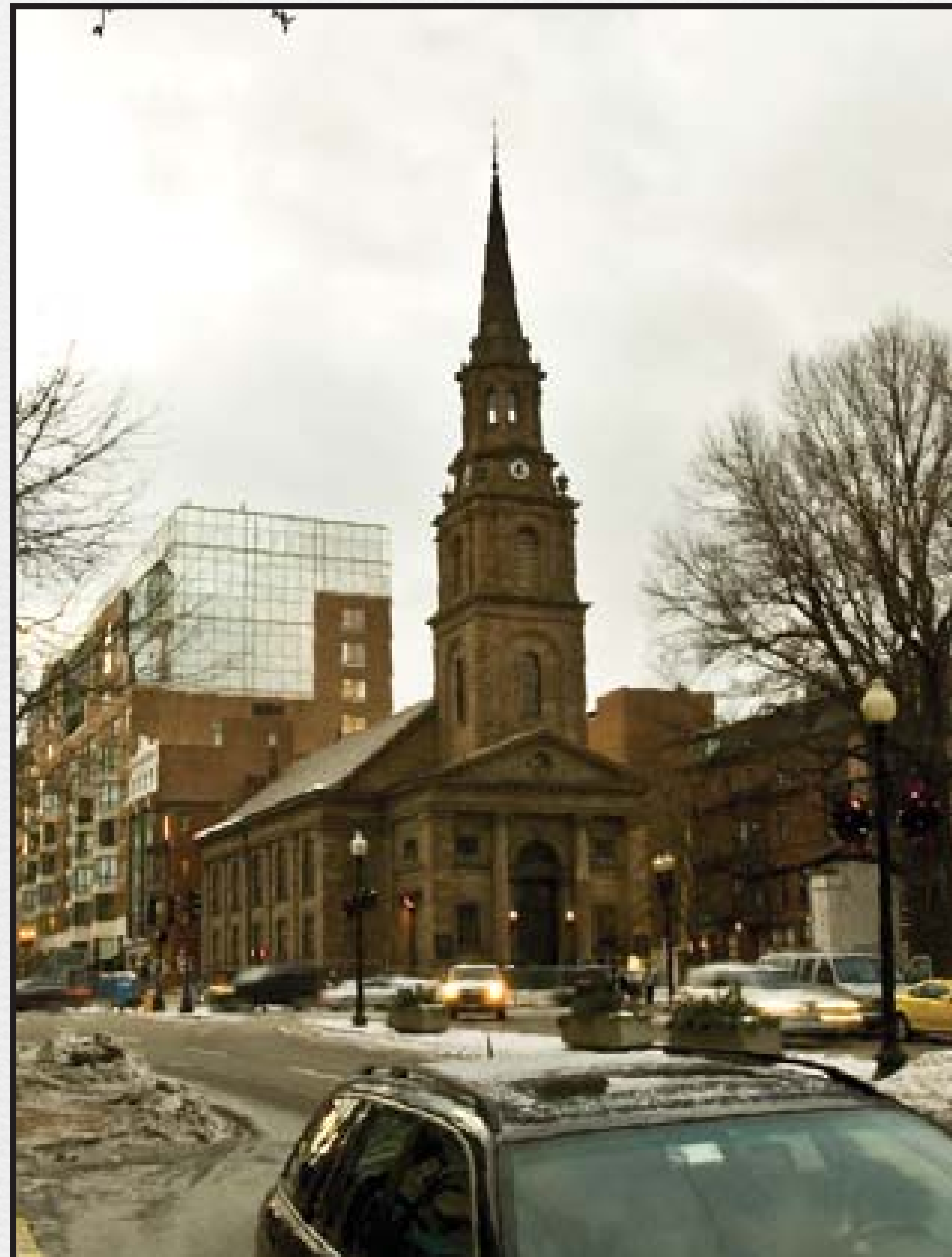
- Bae, Agarwala & Durand, to appear
- Goal: given reference (old) photo, take new photo at same viewpoint



Guidance visualization

- Camera tethered to laptop
- Arrows tell user where to go
- Overlay edges for finer grain





REFERENCES

Video stabilization

- http://www.visionbib.com/bibliography/motion-i781.html
- http://research.microsoft.com/en-us/people/yasumat/fullframe_cvpr05.pdf
- http://pages.cs.wisc.edu/~fliu/project/3dstab.htm
- http://pages.cs.wisc.edu/~gleicher/Web/Projects/ReCinematography
- http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F30%2F31480%2F01467968.pdf%3Farnumber%3D1467968&authDecision=-203
- http://www.cs.unc.edu/~mcmillan/papers/CVPR01_buehler.pdf
- http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04272080
-

Commercial stabilization

- http://www.ovation.co.uk/Video-Stabilization.html
-

Tracking

- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.8498&rep=rep1&type=pdf
-

Features

- http://www.cs.toronto.edu/~jepson/csc2503/tutSIFT04.pdf
- http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.8899