DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF COPENHAGEN

Features II: Detecting Interest Points and Matching

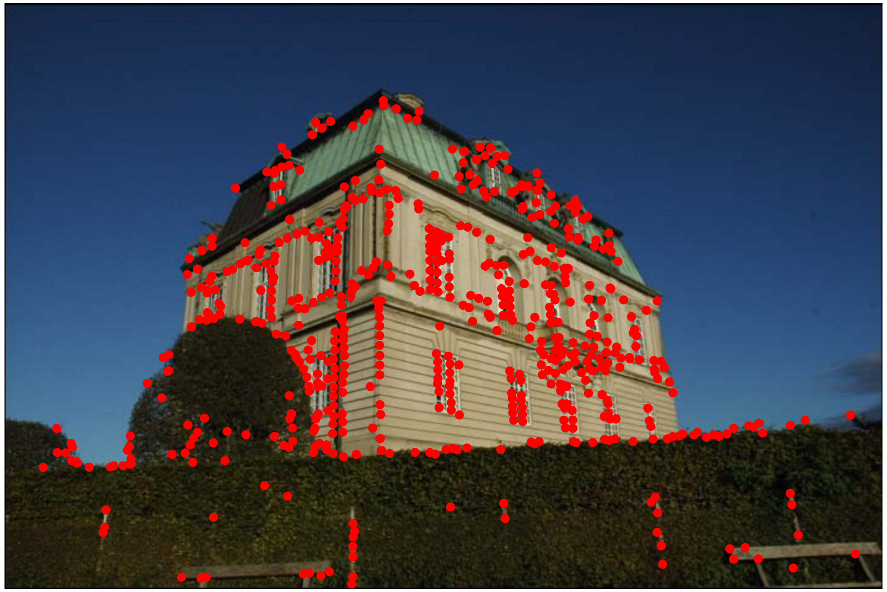
Kim Steenstrup Pedersen

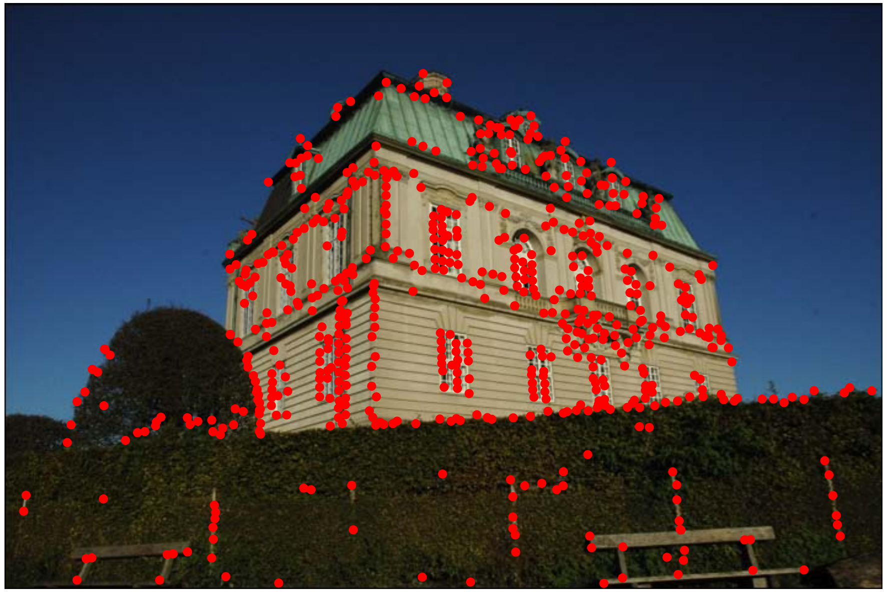
Plan for today and next couple of lectures

- We consider the problem of matching two or more images
- Today we wrap up the problem of detecting good locations to match.
- Then we switch our attention to how to perform the actual matching.

Fundamental Question:
Where did the underlying feature go?


Regular grid vs random points vs salient points





Recall: Linear scale space

- The scale space of *I* is a 1-parameter family

$$L(x,y;\sigma) = (I * G)(x,y;\sigma)$$

$$L(x,y;\sigma = 0) \equiv I(x,y)$$

- Scale is given by σ and is an important parameter in CV algorithms

- As the scale increases details in the image disappear and we focus on the large scale structures that are left.

Recall:
Detecting blobs by Laplacian of Gaussian filter

- Finding extrema in 2D:
  - Find points that are simultaneous extrema in the x and y direction.
  - We can solve this in one go by looking for extrema of the Laplacian of Gaussian filter $\nabla^2 L(x,y;\sigma) = (L_{x^2} + L_{y^2})(x,y;\sigma)$

$$\nabla(\nabla^2 L(x,y;\sigma)) = 0$$

  - Bright blob: $\nabla^2 L(x,y;\sigma) < 0$ Dark blob: $\nabla^2 L(x,y;\sigma) > 0$

- Discrete implementation: Extrema search in 2D
  - Bright blob: $\nabla^2 L(x,y;\sigma) <$ than all neighbor pixels
  - Dark blob: $\nabla^2 L(x,y;\sigma) >$ than all neighbor pixels
  - 4-neighbors or 8-neighbors
  - Keep only blobs where $|\nabla^2 L(x,y;\sigma)| >$ threshold value

Deep structure

The challenge is to understand the image
really on all the levels simultaneously,
and not as an unrelated set of derived images
at different levels of blurring.

Jan Koenderink (1984)

Interest point detectors: Multi-scale Laplacian blob detector

- Finding extrema in scale-space:
  - Due to blurring, the Laplacian response decay over scale – solution: Use scale normalization.
  - Scale normalized Laplacian: $\nabla^2 L(x,y;\sigma) = \sigma^2 (L_{x^2} + L_{y^2})$
  - Bright blob: $\nabla^2 L(x,y;\sigma) < 0$ Dark blob: $\nabla^2 L(x,y;\sigma) > 0$

- Discrete implementation: Extrema search in 3D
  - Bright blob: $\nabla^2 L(x,y;\sigma) <$ than all neighbor pixels
  - Dark blob: $\nabla^2 L(x,y;\sigma) >$ than all neighbor pixels
  - 6-neighbors or 26-neighbors
  - Keep only blobs where $|\nabla^2 L(x,y;\sigma)| >$ threshold value

Interest point detectors:
Detecting Laplacian blobs (fixed scale)

Interest point detectors: Detecting Laplacian blobs (multi-scale)

Interest point detectors: Detecting blobs by Difference of Gaussians (DoG)

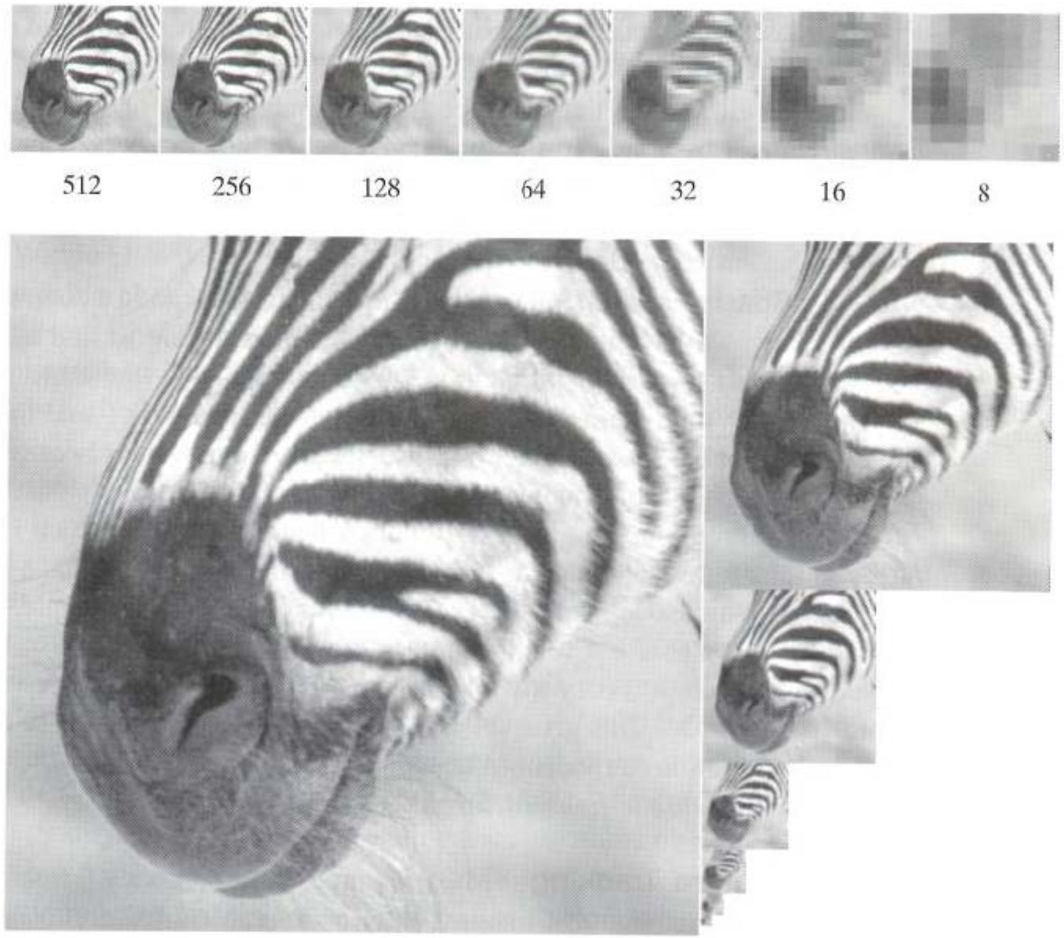
- Difference of Gaussians (DoG) is an approximation to the Laplacian of Gaussian filter

$$D(x,y;\sigma) = (G(x,y;k\sigma) - G(x,y;\sigma)) * I(x,y)$$

$$= L(x,y;k\sigma) - L(x,y;\sigma) \approx \nabla^2 L(x,y;\sigma)$$

- Either subtract Gaussian filters prior to filtering or subtract filtered images.
- As before look for extrema in $D(x,y;\sigma)$
- In a multi-scale setting compute the scale space images with factor *k* increase in scale and search for extrema in scale space. Can be combined with a Gaussian pyramid for increased processing speed.

Multi-scale versus multi-resolution: The Gaussian pyramid



Interest point detectors:
Detecting Harris corners (fixed scale)

Find max. of $R(x,y;\sigma) = \det(\mathbf{A}) - \alpha\,\mathrm{trace}(\mathbf{A})^2$

$$\mathbf{A}(x,y;\sigma) = \begin{bmatrix} G(x,y;k\sigma) * L_x^2(x,y;\sigma) & G(x,y;k\sigma) * (L_x(x,y;\sigma)L_y(x,y;\sigma)) \\ G(x,y;k\sigma) * (L_x(x,y;\sigma)L_y(x,y;\sigma)) & G(x,y;k\sigma) * L_y^2(x,y;\sigma) \end{bmatrix}$$

Interest point detectors: Detecting Harris corners (multi-scale)

Find max. of $R(x,y;\sigma) = \sigma^4(\det(\mathbf{A}) - \alpha\,\mathrm{trace}(\mathbf{A})^2)$

$$\mathbf{A}(x,y;\sigma) = \begin{bmatrix} G(x,y;k\sigma) * L_x^2(x,y;\sigma) & G(x,y;k\sigma) * (L_x(x,y;\sigma)L_y(x,y;\sigma)) \\ G(x,y;k\sigma) * (L_x(x,y;\sigma)L_y(x,y;\sigma)) & G(x,y;k\sigma) * L_y^2(x,y;\sigma) \end{bmatrix}$$

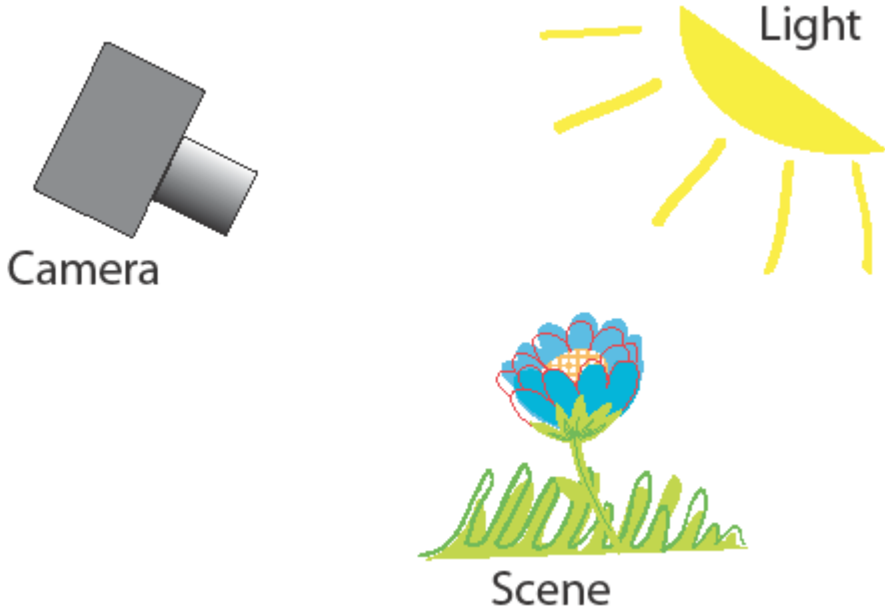
So which detectors are the best?

Based on joint work with Anders Dahl and Henrik Aanæs from DTU

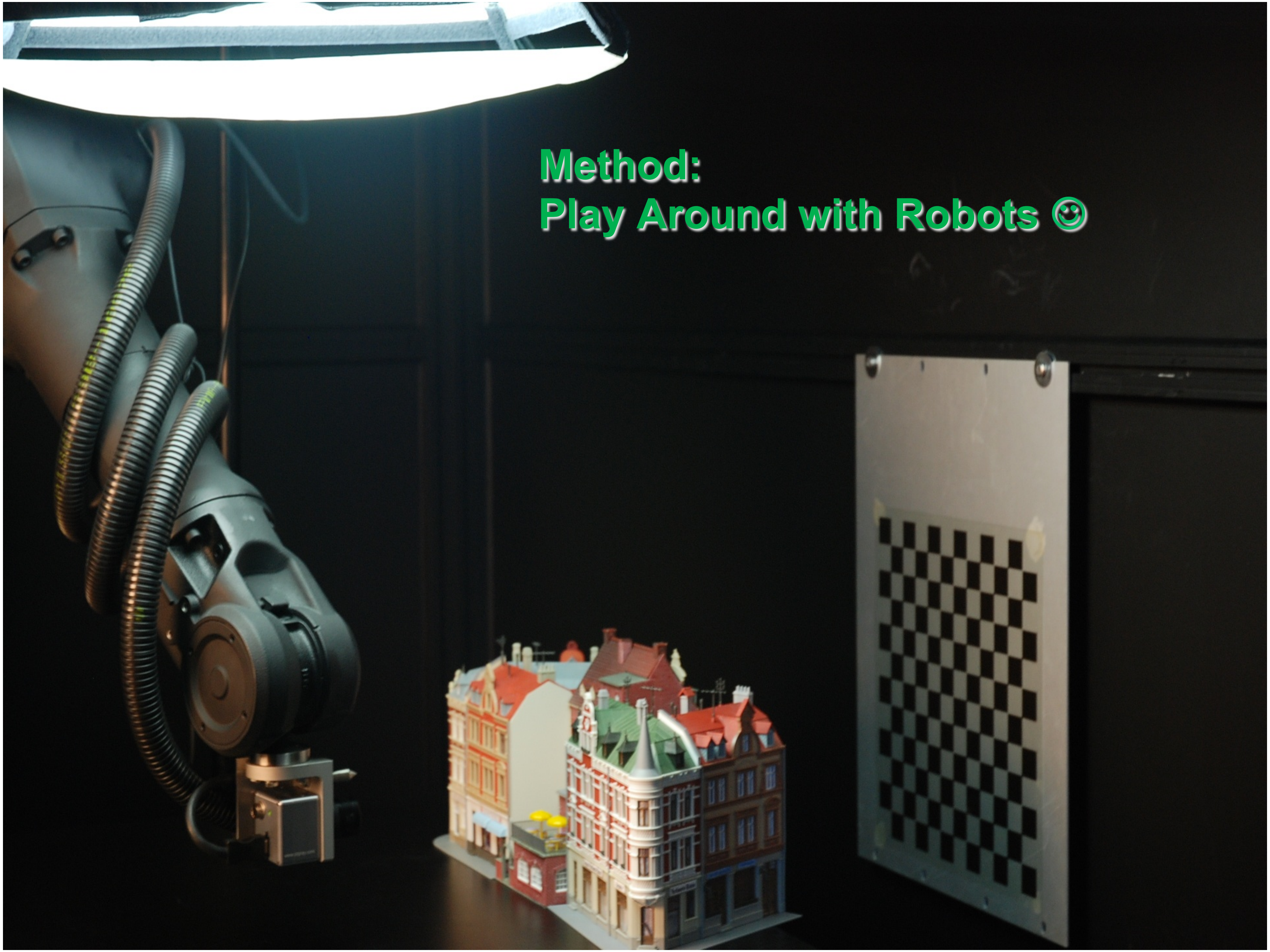
Interesting Interest Points.

International Journal of Computer Vision, 97:18 – 35, 2012

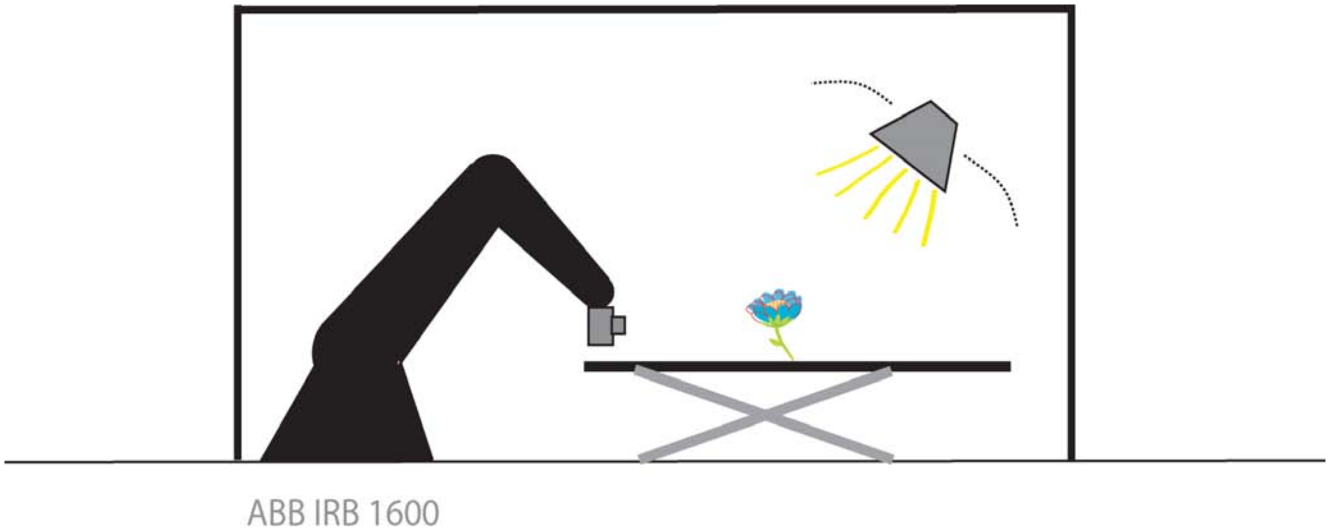
Computer Vision Basics:
The Image Formation Process



We Want to Control it All in a Systematic Way


Method:
Play Around with Robots ☺

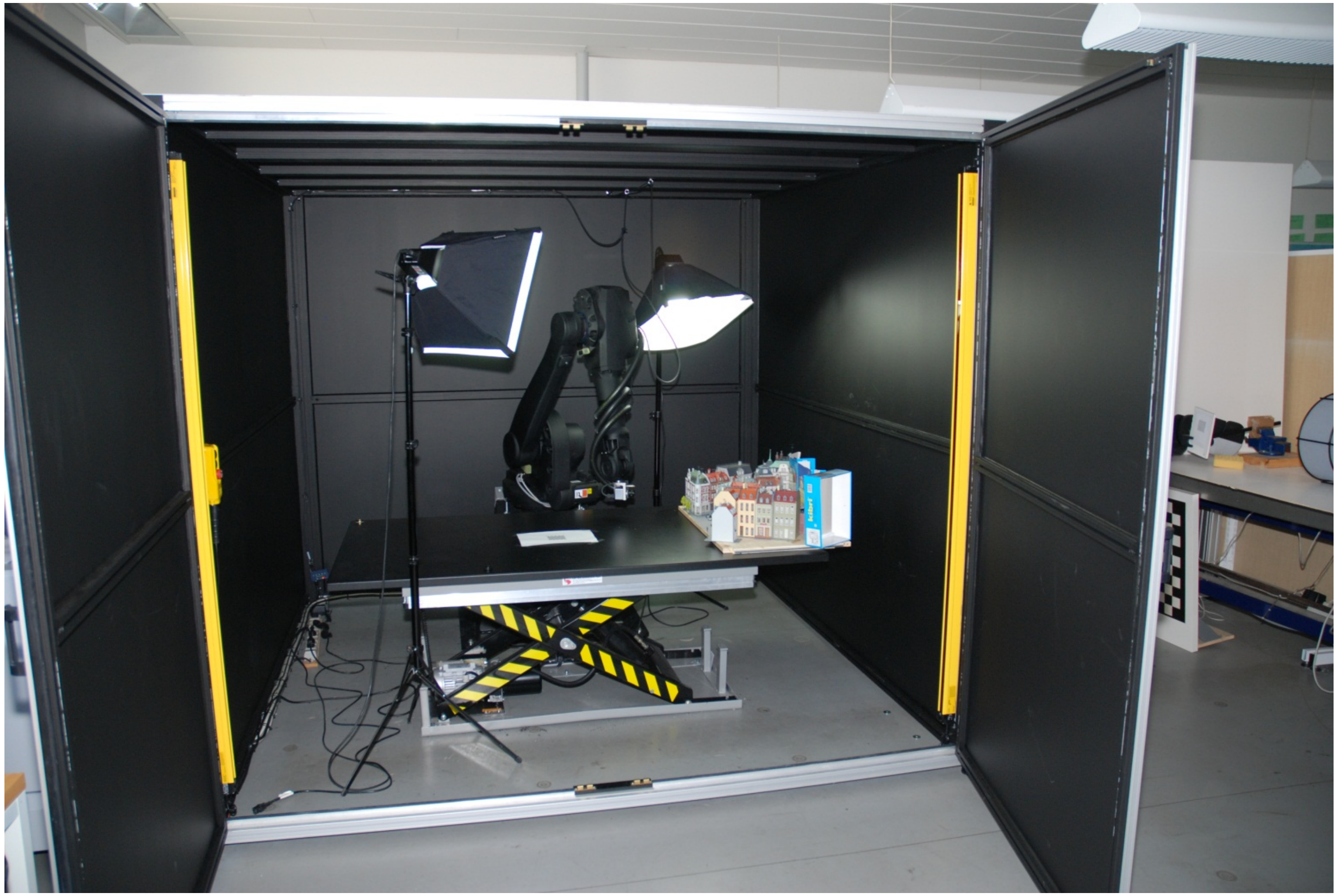
Method:
A Configurable Computer Vision Workbench



Controlled/Measured:

- Scene
- Geometry
- Camera position
- Camera Optics
- Light

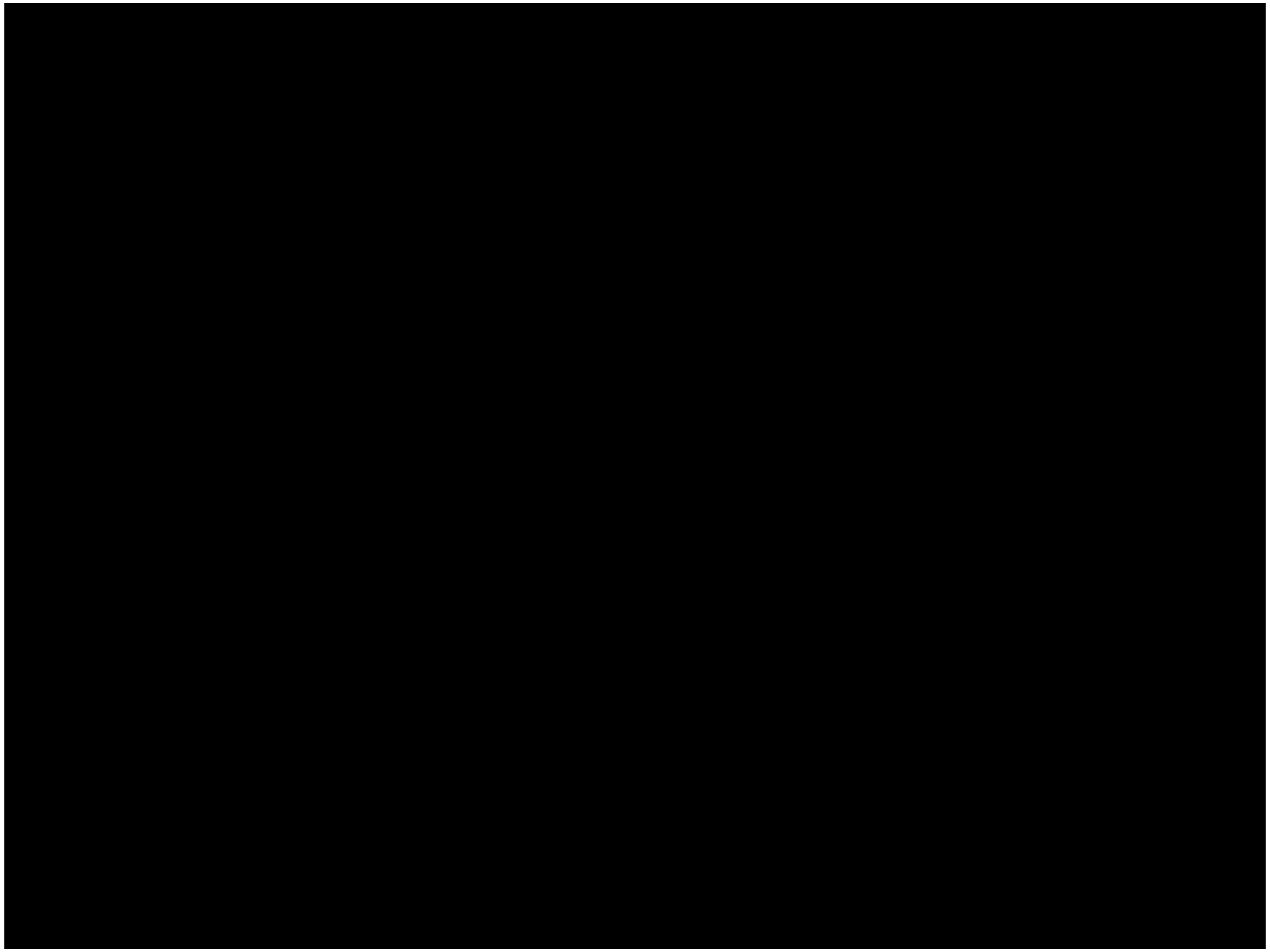
Method:
A Configurable Computer Vision Workbench



Light

“**A Lighting Reproduction Approach to Live-Action Compositing** “, Debevec et al.

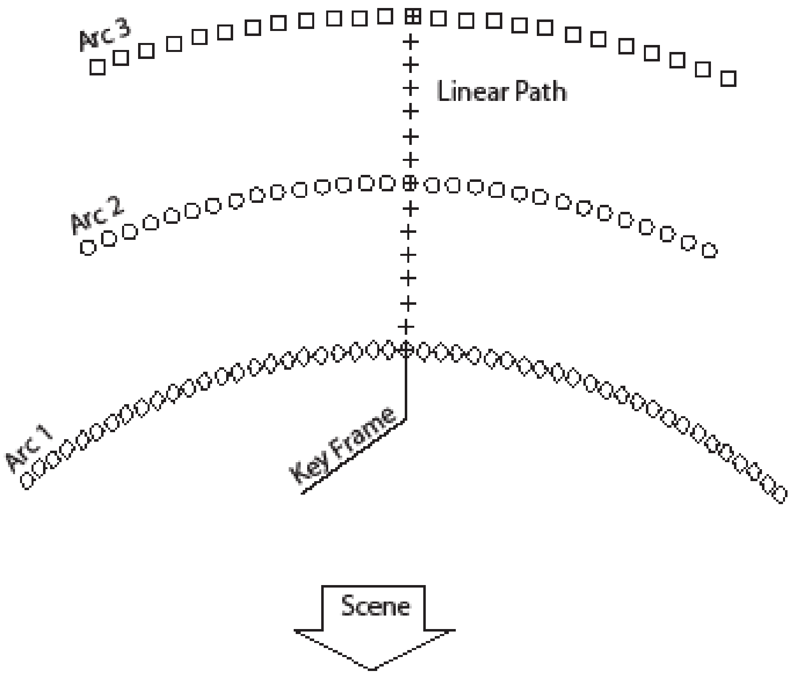
Data Recording – Fast forward



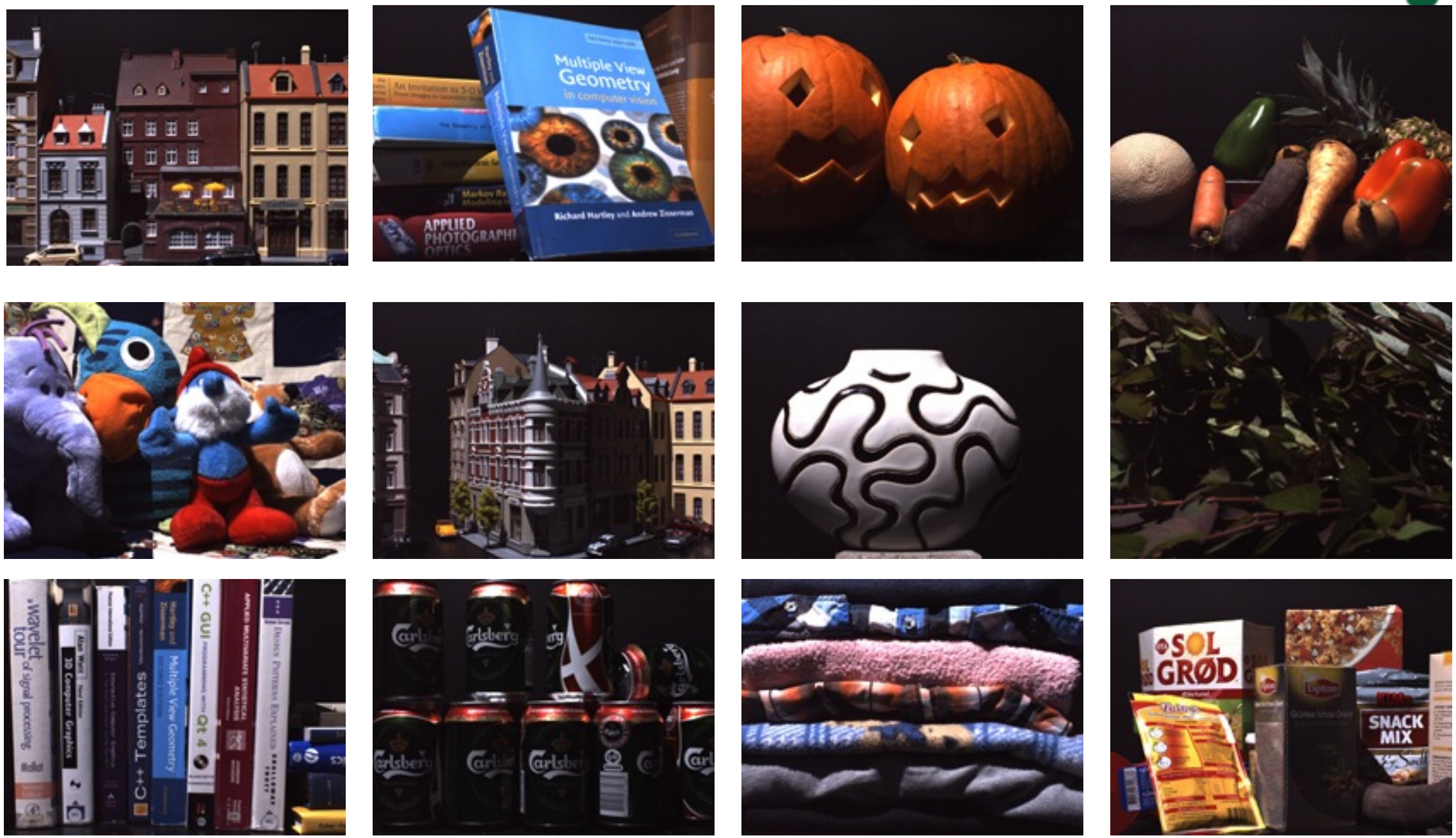
What we used it for

FEATURE MATCHING EVALUATION

Data set:
60 Scenes, 119 Positions

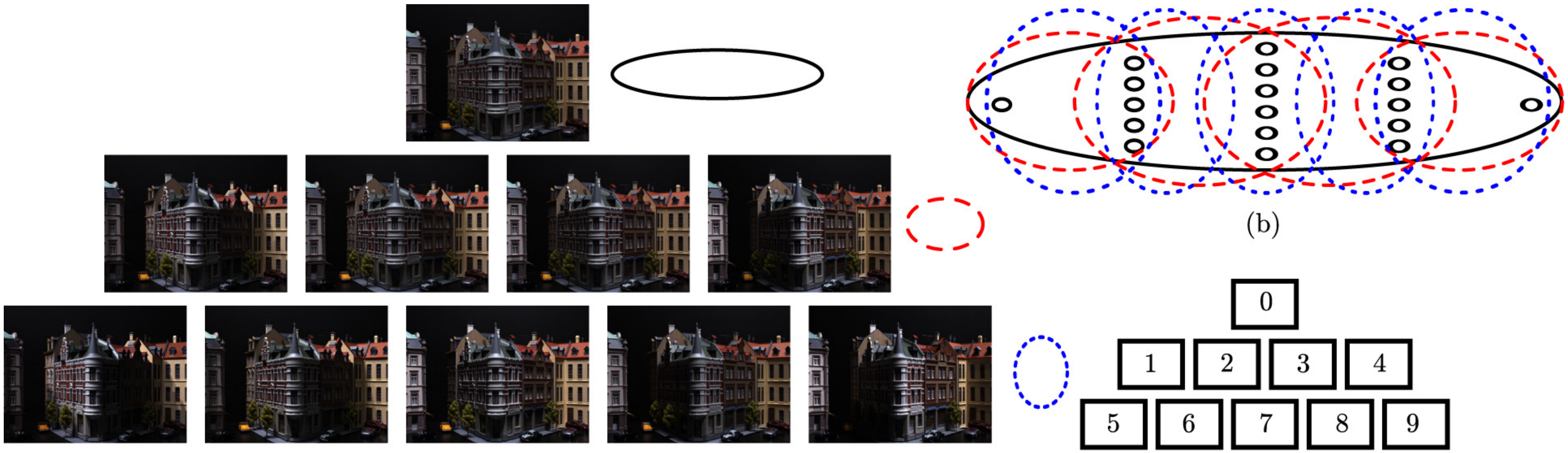


Sample Scenes – 60 in All



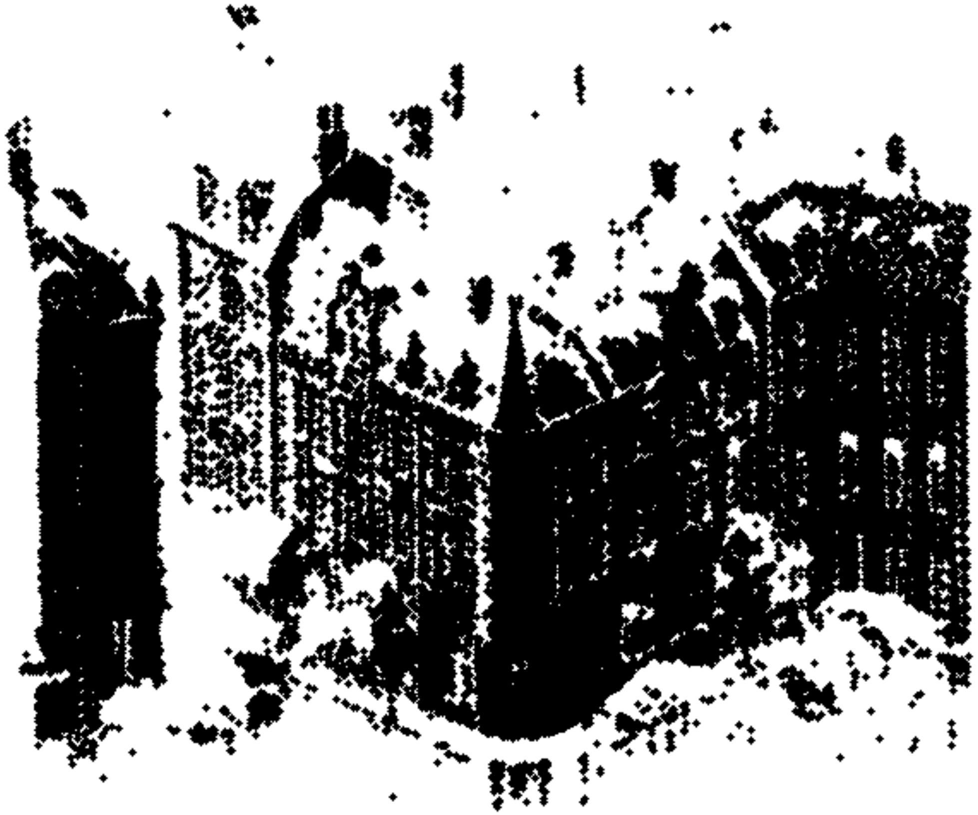
An Example

Light

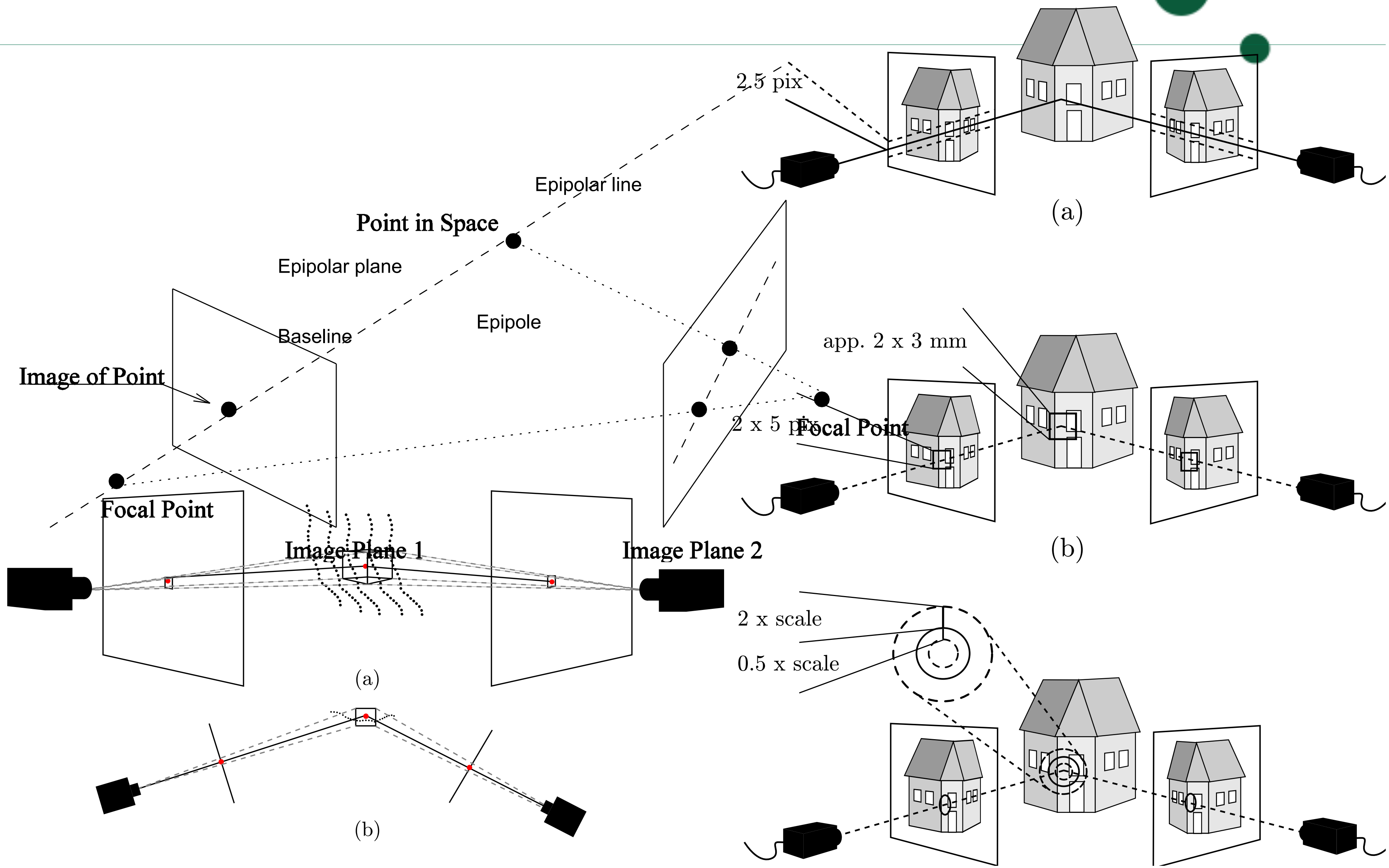


(a) (b) (c)

Ground 'Truth' Via Structured Light



Evaluation Criteria:
Determining the Correctness of a Match



Our Analysis:

Investigate the recall rate of Interest Point Detectors

$$\text{Recall} = \frac{\text{Potential Matches}}{\text{Total Interest Points}}$$

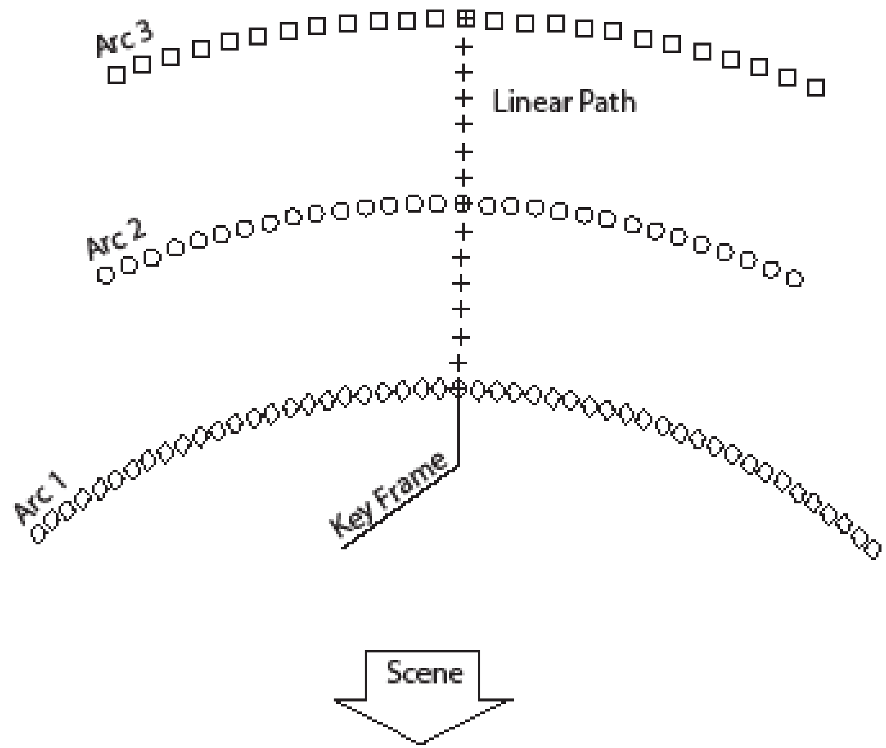
<u>Detectors used:</u>

- Harris and Hessian Corner Detectors — Mikolajczyk & Schmid 2004
  - Plain
  - Laplace
  - Affine

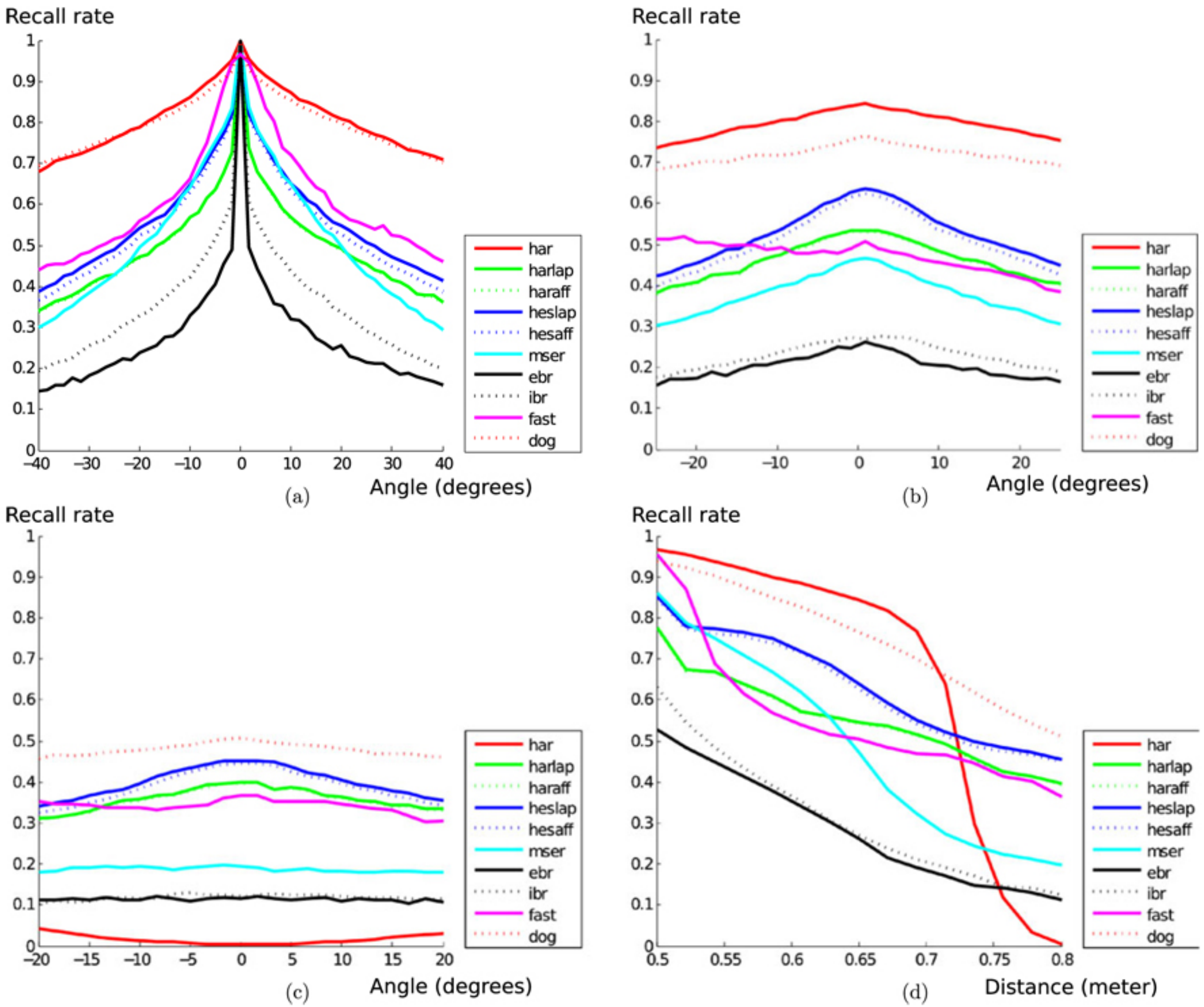
- MSER — Matas et al. 2004
- Intensity & Edge Based Regions — Tuytelaars & Van Gool 2009
- Fast Corner detector — Trajkovic & Hedley 1998
- DoG — Lowe 2004

Implementations downloaded of the net

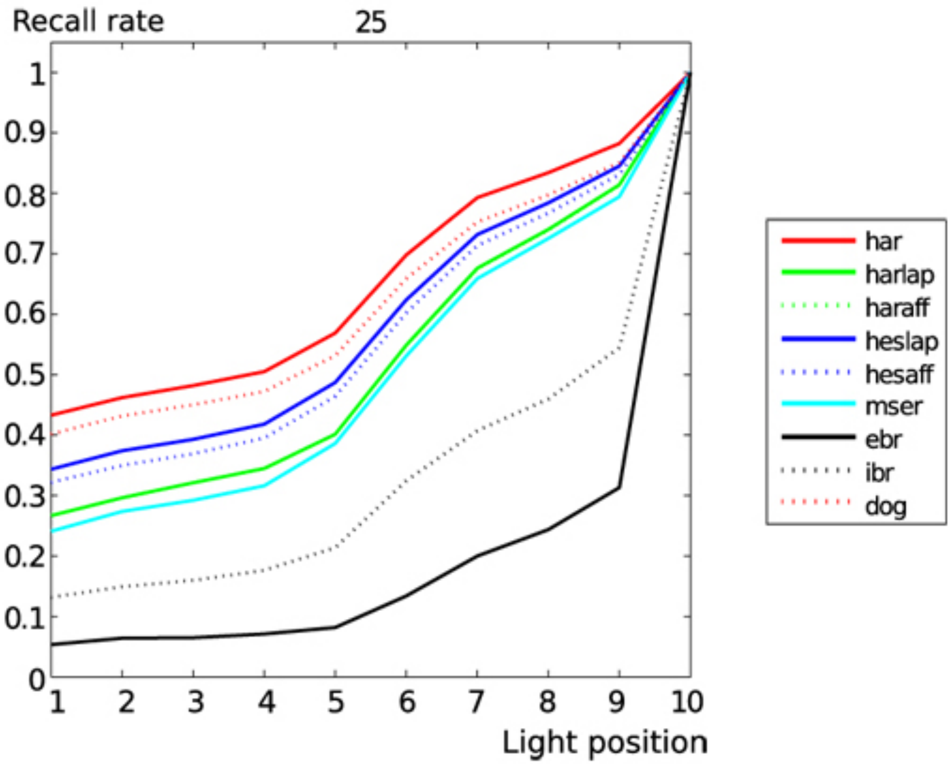
Effect of Position



Effect of position



Effects of light



A Challenge with Changing Light

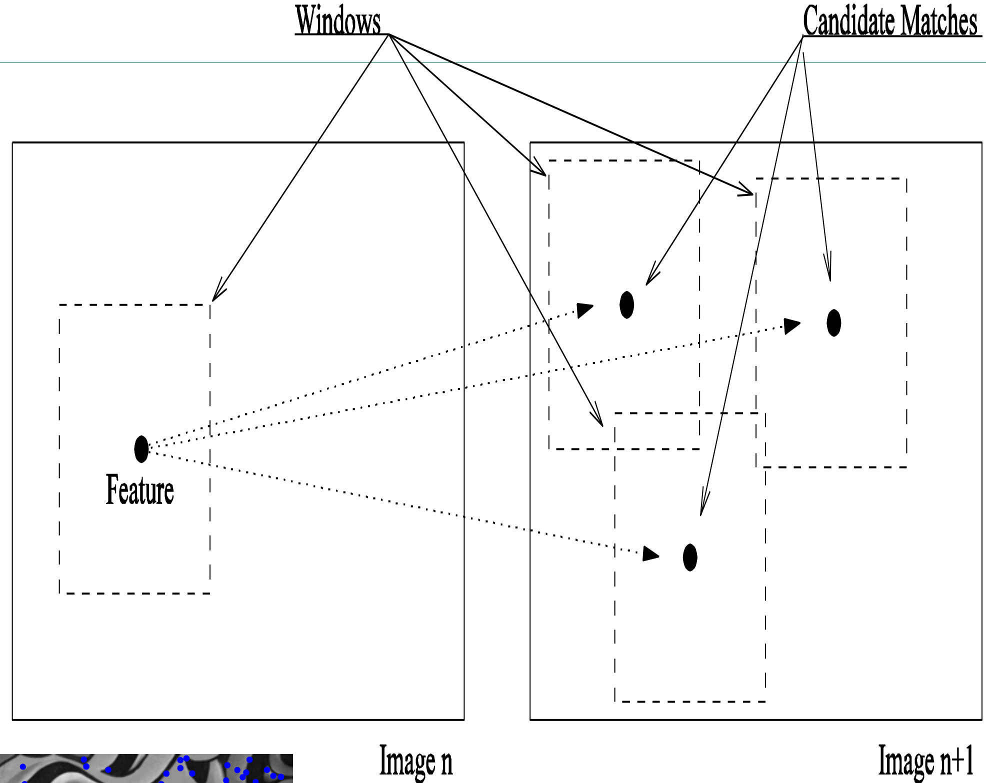
Outline of First Stage Findings

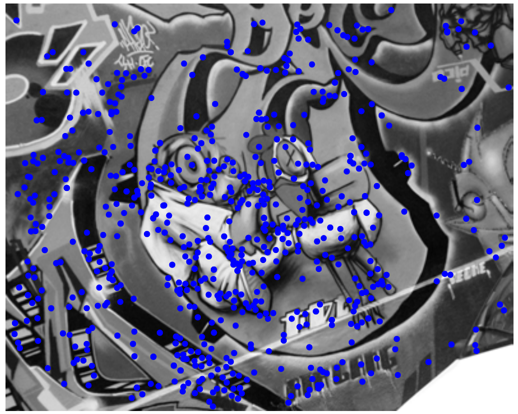
- Light is more disruptive than position (angle & scale).

- Best Performers are:
  - Harris Variants
  - DoG (SIFT Blobs)
  - MSER
  - Hessian Laplace Variants
- Observations:
  - DoG (SIFT Blobs) is much better than Hessian Laplace.
  - Plain Harris is better than extensions, when scale does not change.
  - FAST is very erratic wrt. to number of features.

How can we perform the actual matching without ground truth?

Matching Strategy Illustrated





Salient Feature Matching Strategy

For a pair of images:

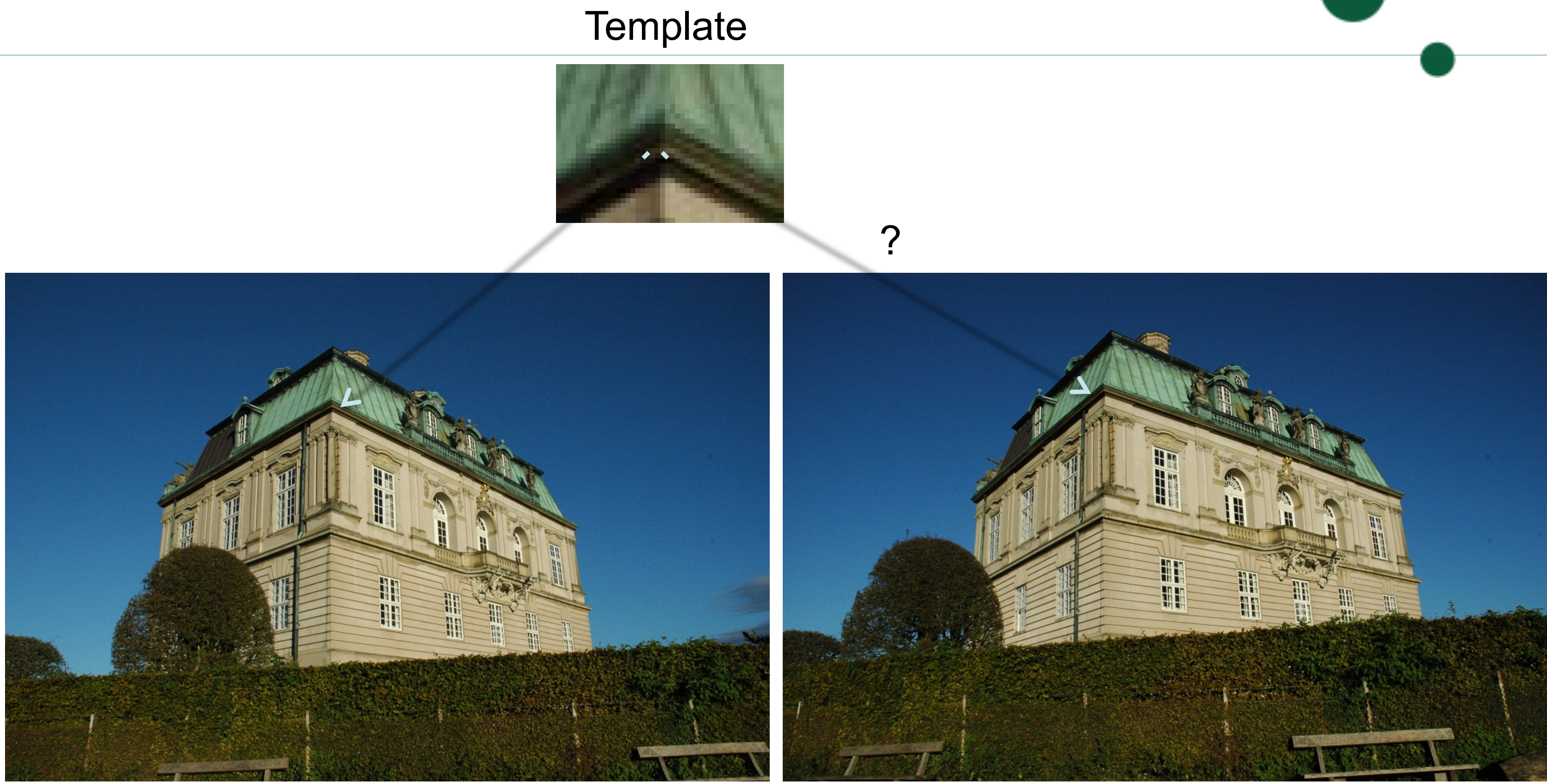
1. Extract salient points, via **detectors**
e.g. Harris corners, DoG (blobs), MSER

2. Compute Feature **descriptors**,
eg. Raw patch (correlation), SIFT, DAISY

3. Match features by pairing similar descriptors

Aggregate solution to multiple images, if needed.

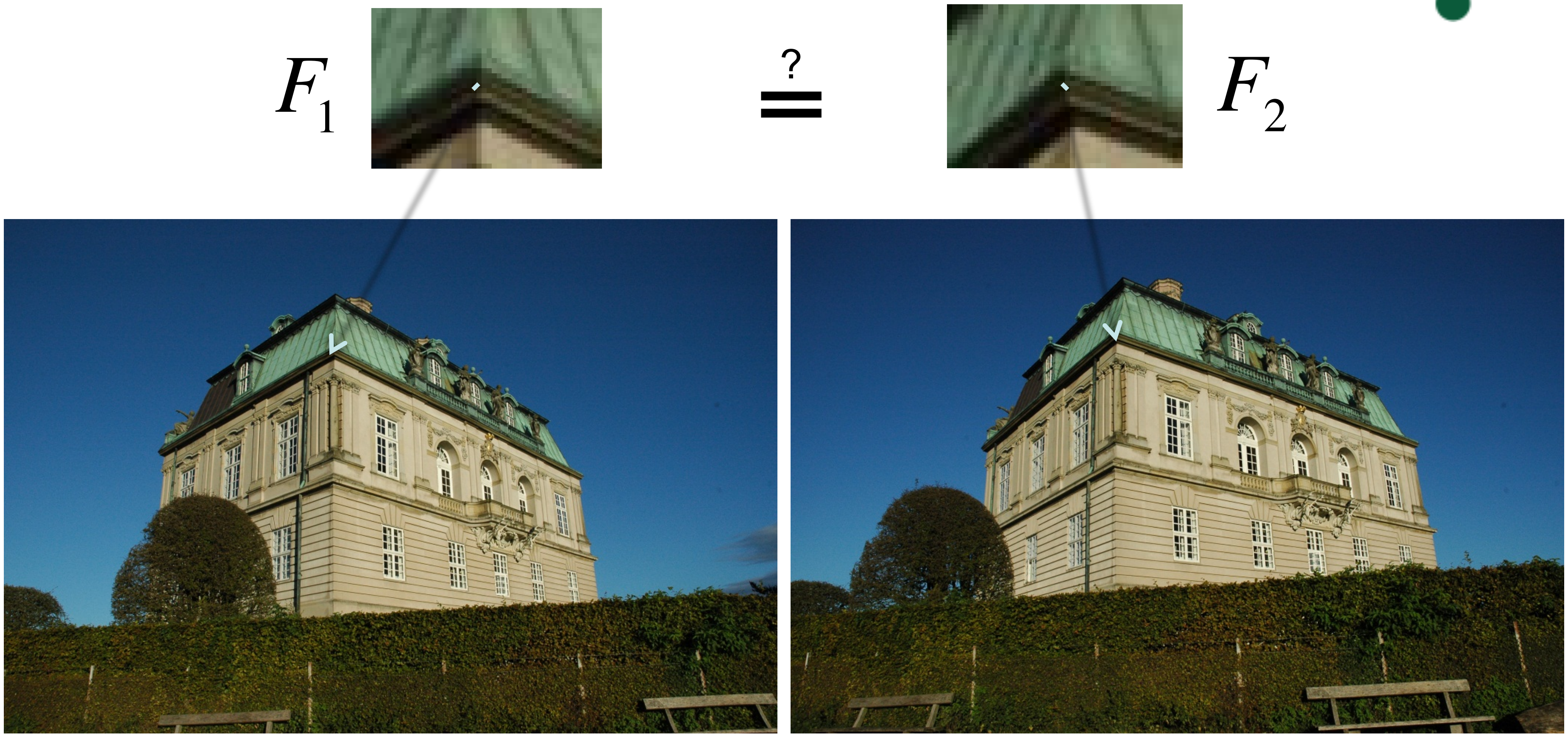
Detecting points by template matching



How to construct a generic corner template?   Bad idea – won't work!

A bad idea for point detection, but what about description?

Matching patches



Using raw pixel patches as descriptors

- Represent a patch with the raw pixels – the patch is the descriptor.
- Compare patches pixel by pixel, e.g. by

$$d_1(F_1,F_2) = \sum_x \sum_y \left| F_1(x,y) - F_2(x,y) \right| \qquad \text{(L1 - norm)}$$

$$d_2(F_1,F_2) = \sqrt{\sum_x \sum_y (F_1(x,y) - F_2(x,y))^2} \qquad \text{(L2 - norm)}$$

(For simplicity lets assume gray scale intensities and the sums are over all pixels in the patches.)
(Patches must be of equal size)

- This is referred to as either distances or dissimilarity measures

Using raw pixel patches as descriptors

- We need to compensate for changes in illumination conditions, because …
- Contrast change *a* and change in brightness level *c* (affine model):

$$F' = aF + c$$

- We want *F'* and *F* to have zero distance, but

$$d_1(F, F') \neq 0$$

$$d_2(F, F') \neq 0$$

- Invariance wrt. affine intensity changes by normalization:

$$F' = aF + c \Rightarrow F = \frac{F' - c}{a}$$

Matching patches: Normalized cross correlation

- Compute mean intensity and standard deviation for each patch

$$\overline{F_1} = \frac{1}{n}\sum_{x,y} F_1(x,y)\ ,\ \overline{F_2} = \frac{1}{n}\sum_{x,y} F_2(x,y)$$

$$\sigma_1^2 = \frac{1}{n}\sum_{x,y}(F_1(x,y)-\overline{F_1})^2\ ,\ \sigma_2^2 = \frac{1}{n}\sum_{x,y}(F_2(x,y)-\overline{F_2})^2$$

(sum over all *n* pixels in patches)

- Measure distance with normalized cross correlation

$$NCC(F_1,F_2) = 1 - \frac{1}{n}\sum_{x,y}\frac{(F_1(x,y)-\overline{F_1})(F_2(x,y)-\overline{F_2})}{\sigma_1\sigma_2}$$

- Affine intensity invariance: $F' = aF + c$

$$NCC(F,F') = 0$$

Open problems

- What patch size should we use?
  - Use the detection scale and resample so both patches have equal size in pixels

- Is this approach robust to scale changes in the scene?
  - Yes, if we do the resampling (see above)

- Is this approach robust to rotation and translation in the scene?
  - No, this will lead to large dissimilarities

- Is it robust to perspective distortions?
  - No, this will lead to large dissimilarities

More on this at next lecture

Summary

- Multi-scale interest point detection
  - Blobs (DoG, Laplacian)
  - Corners (Harris corners)
- Comparison of state of the art detection methods
- Matching salient points
  - Using raw pixels and normalized cross correlation
  - Works sometimes, but not a robust solution

Literature

Reading material:

- Aanæs-Dahl-Pedersen IJCV 2012

Additional material:

- T. Lindeberg 1996 (Multi-scale detection)
- T. Lindeberg. Feature detection with automatic scale selection. International Journal of Computer Vision 30(2): 79 – 116, 1998.

Any questions about the assignment?