HDR images acquisition: artifacts removal

dr. Francesco Banterle

francesco.banterle@isti.cnr.it

things can go wrong…

Things can move

- What happens if…
  - the camera moves; not stable ground, handled photography (no tripod), etc.
    - especially bad for long exposure images!
  - the scene is not static; moving objects, background, etc…

a moving camera…

Moving camera

- When the camera moves (even small movements) and the scene is static, the final HDR image will be blurry

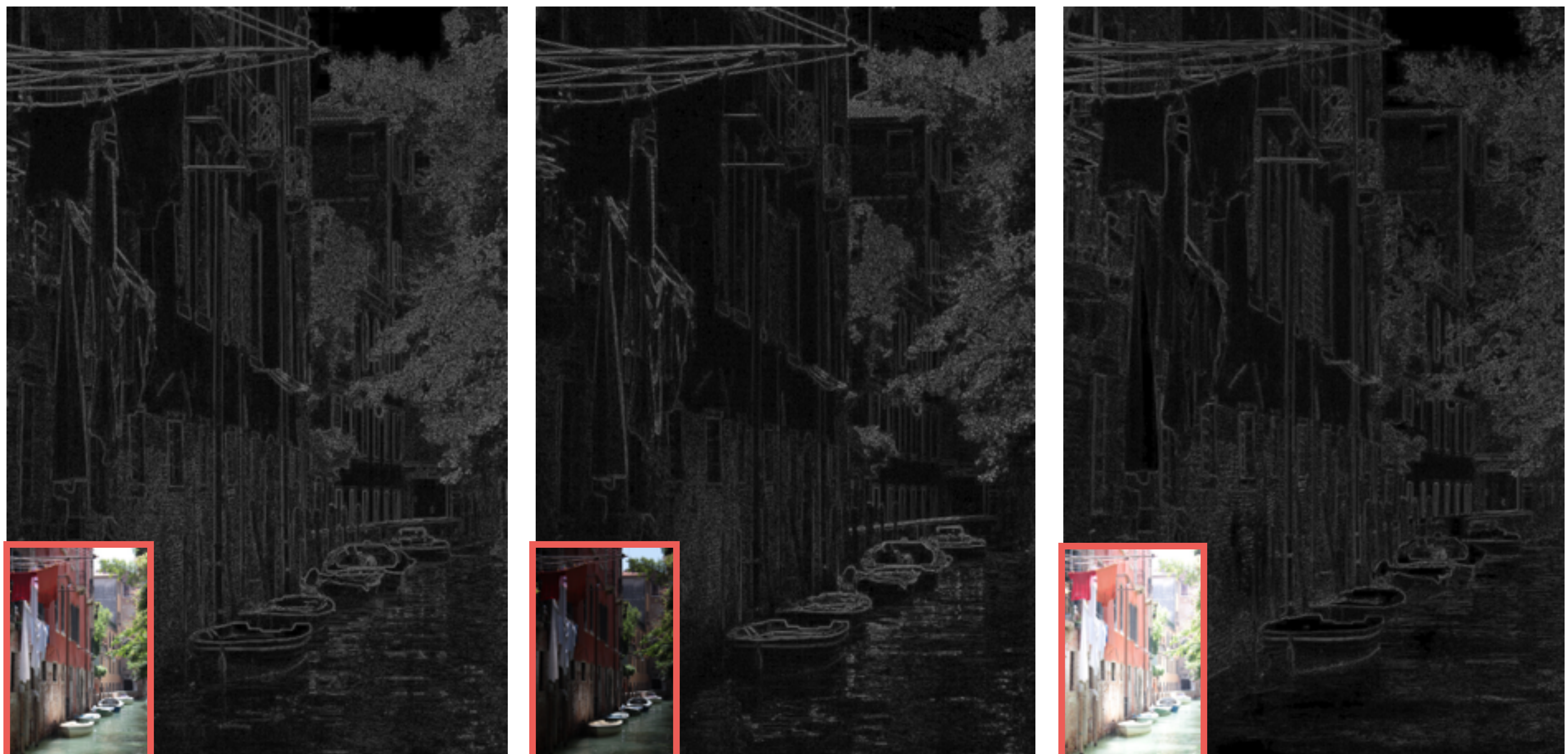



Moving camera

- What to do?
  - Before merging, LDR images need to be aligned to a reference
  - How to select a reference?
    - Typically the image with the highest number of well exposed pixels
    - Typically working in group of three images; hierarchical

Moving camera

- Edges can vary at different exposure times:



Median Threshold Binary Alignement

- MTB, a feature descriptor, is a binary mask:
  - compute the luminance median value, M
  - Then MTB is defined as:

$$MTB(\mathbf{x}) = \begin{cases} 1 & \text{if } L(\mathbf{x}) > M \\ 0 & \text{otherwise} \end{cases}$$

  - It is exposure-time invariant!

MTB Alignement

MTB Alignement

- Hierarchical registration - setup:
  - image pyramid
  - max displacement is 2^depth

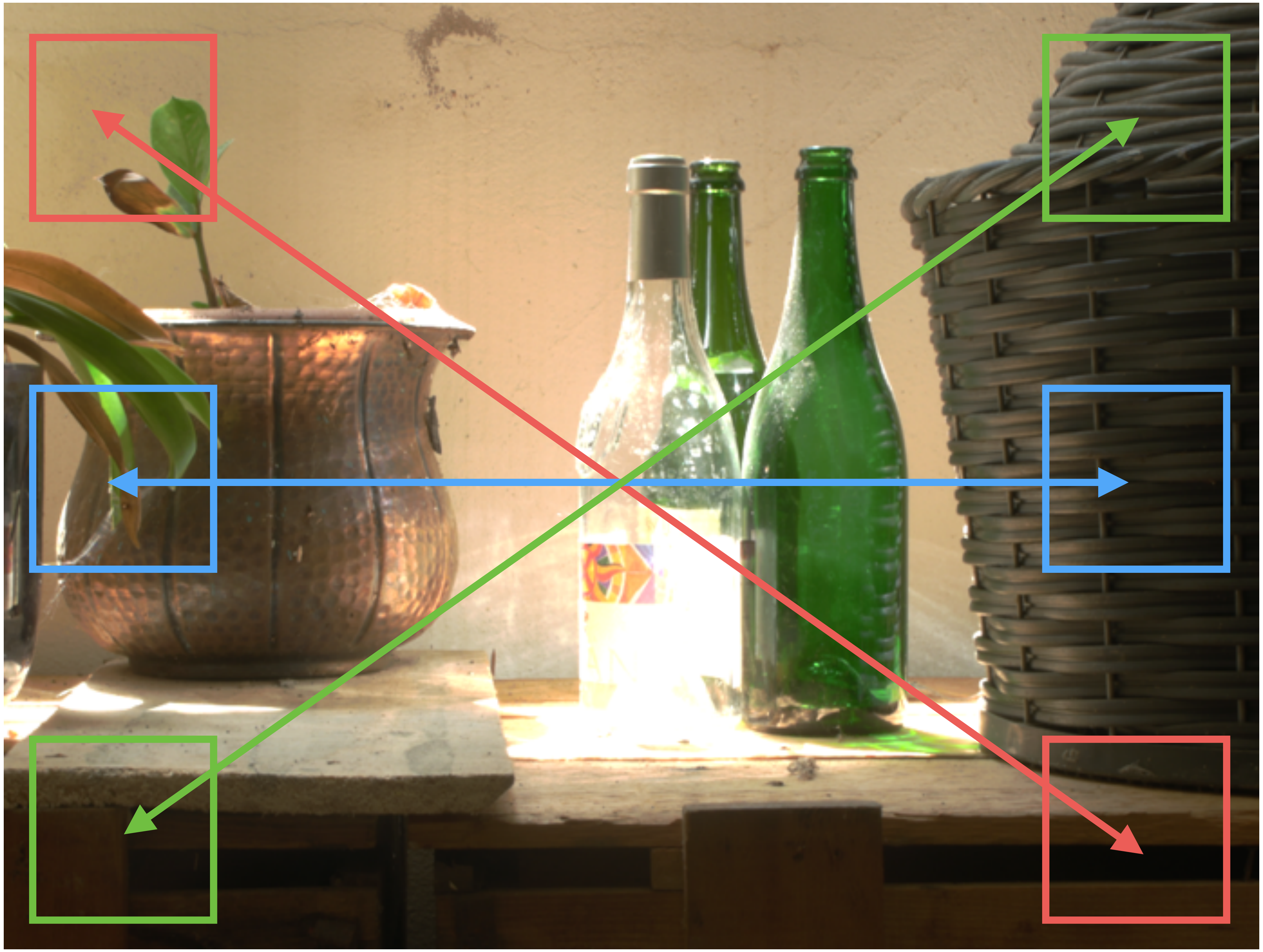
MTB Alignement

- Hierarchical registration:
  - At level n, translation of testing in X and Y (-1, 0, +1)
  - Check the match with XOR
  - Repeat for level n+1 to depth

MTB Alignment: handling camera rotations

- The basic method does not handle rotation, only image translations
- Brute force approach:
  - Run MTB alignment
  - Rotate the testing mask at different degrees and do XOR test. It requires a GPU implementation to achieve fast results
  - Refinement; reapplying MTB Alignment

MTB Alignment: handling camera rotations

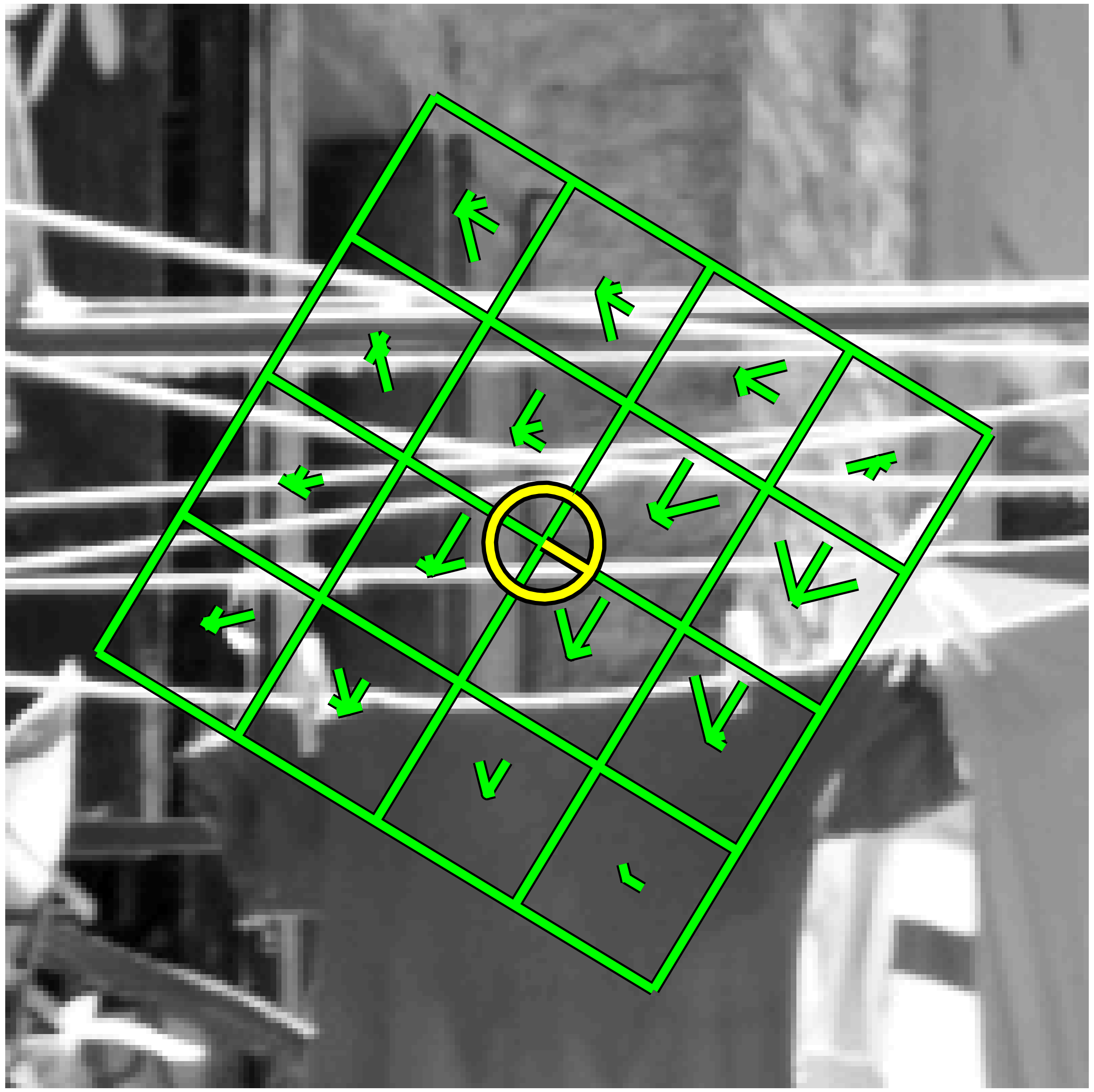


Local Features Alignment

- Detect salient points in an image; i.e. corners or key-points:
  - DoG pyramid method
  - Harris corner detector
  - SUSAN corner detector
  - etc…

Local Features Alignment

- For each key-point:
  - Compute a local descriptor of the image around it



Local Features Alignment

Local Features Alignment

- After matching —> finding a transformation H

- H needs to map 2D coordinates between image0 and image1:

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

- H has to be a *homography*

Local Features Alignment

- A homography is defined as:

$$\mathbf{H} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & 1 \end{bmatrix}$$

- So 8 matches (minimum) are required to estimate H:
  - better more points to avoid noise
  - better to use RANSAC to avoid outliers
- H estimation requires to solve a linear system + non-linear optimization

Local Features Alignment

- Once, H is computed, pixels in image1 to be aligned to image0 need to be warped:

  for i=0 to height

    for j= 0 to width

  $$(u, v) = \mathbf{H}[i, j, 1]^T$$

  $$\text{image}'_1(i, j) = \text{image}_1(u, v)$$

    end

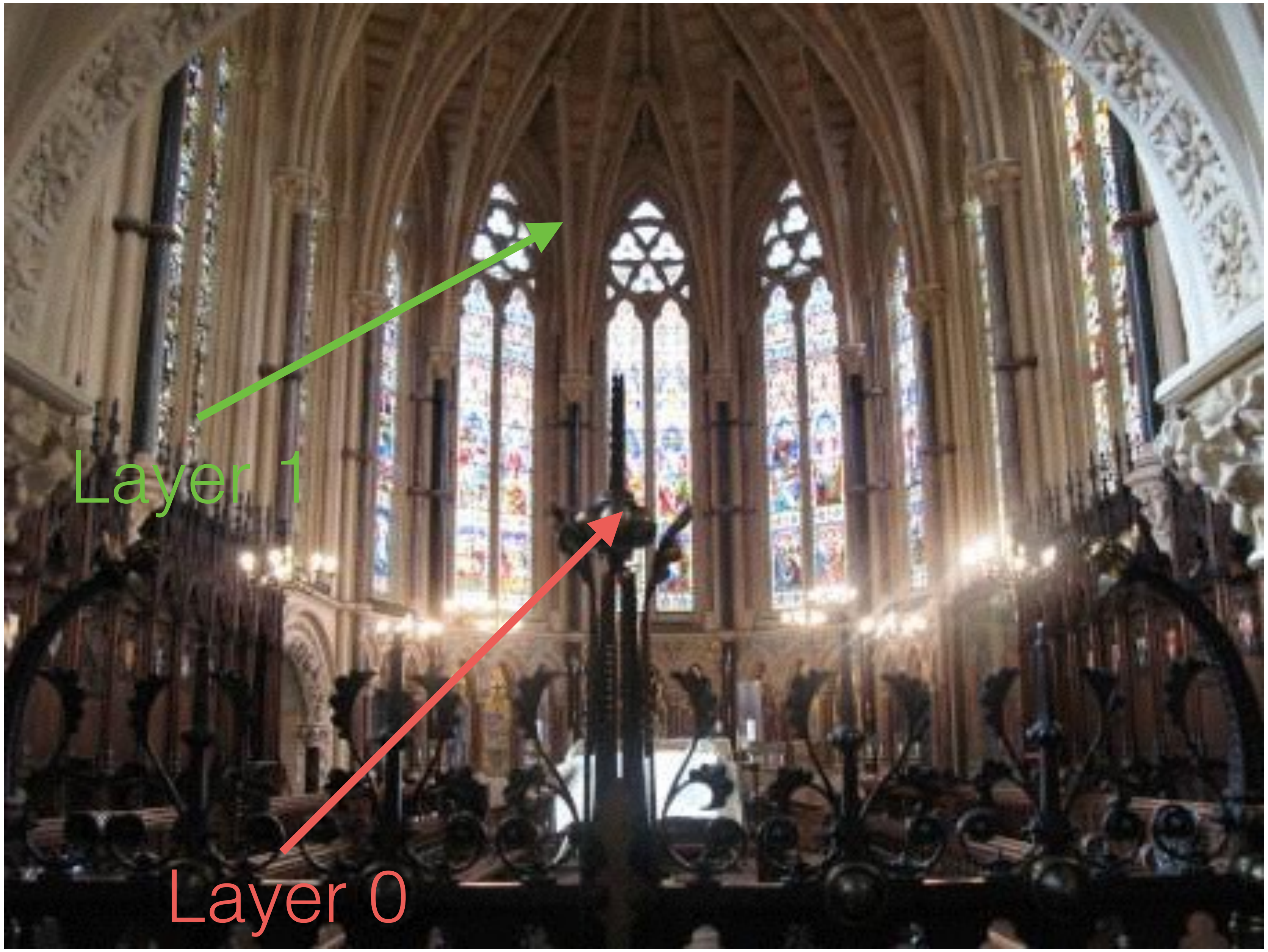
  end

Local Features Alignment

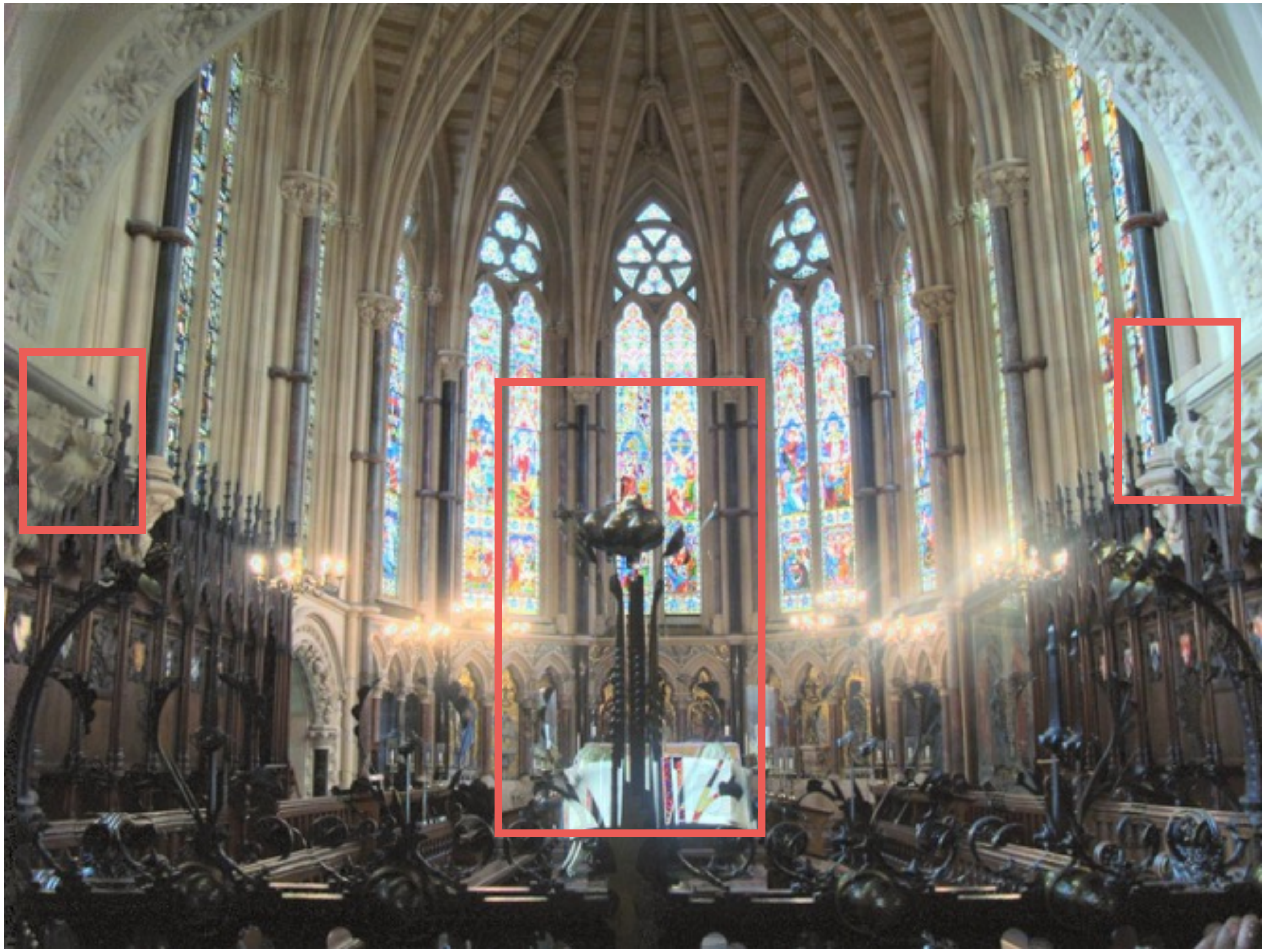
Local Features Alignment: failure cases

- Homography —> planar scene
    - all objects cannot be aligned when they have different depths —> parallax problem!

Local Features Alignment: failure cases



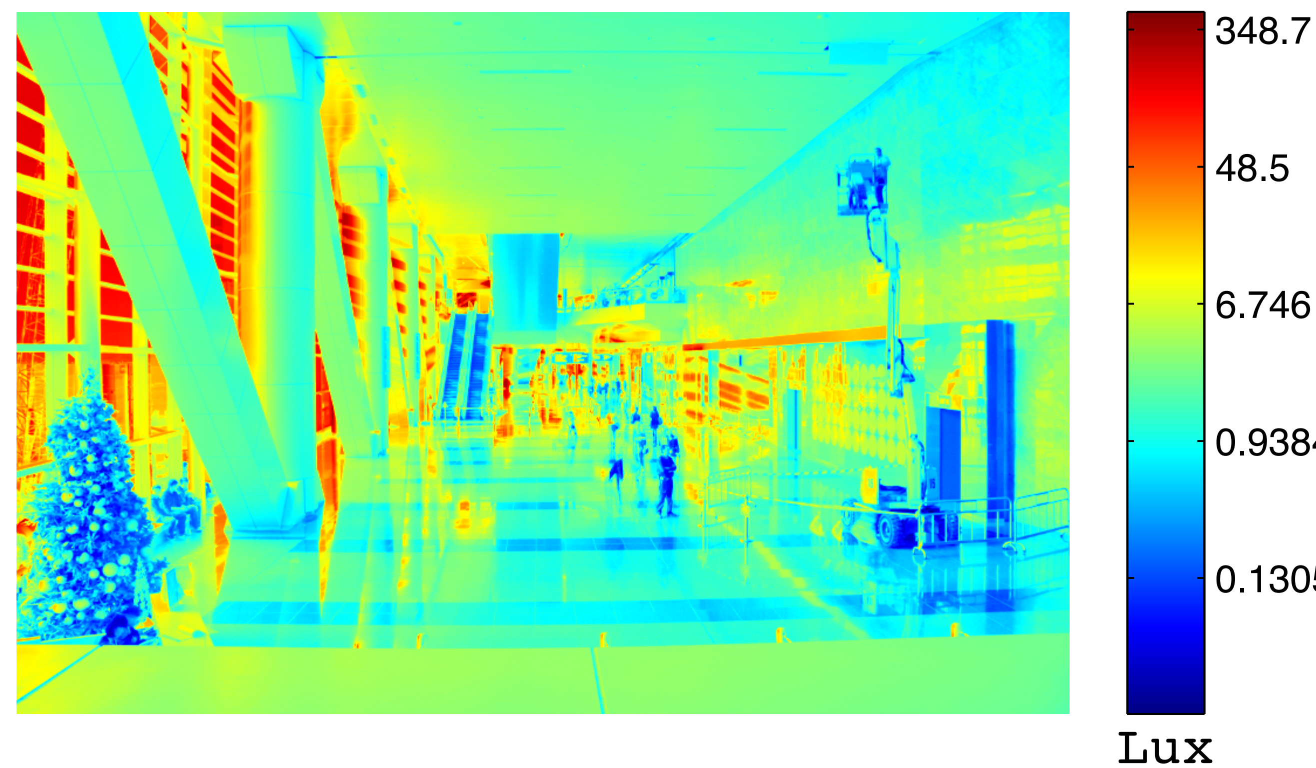
Local Features Alignment: failure cases



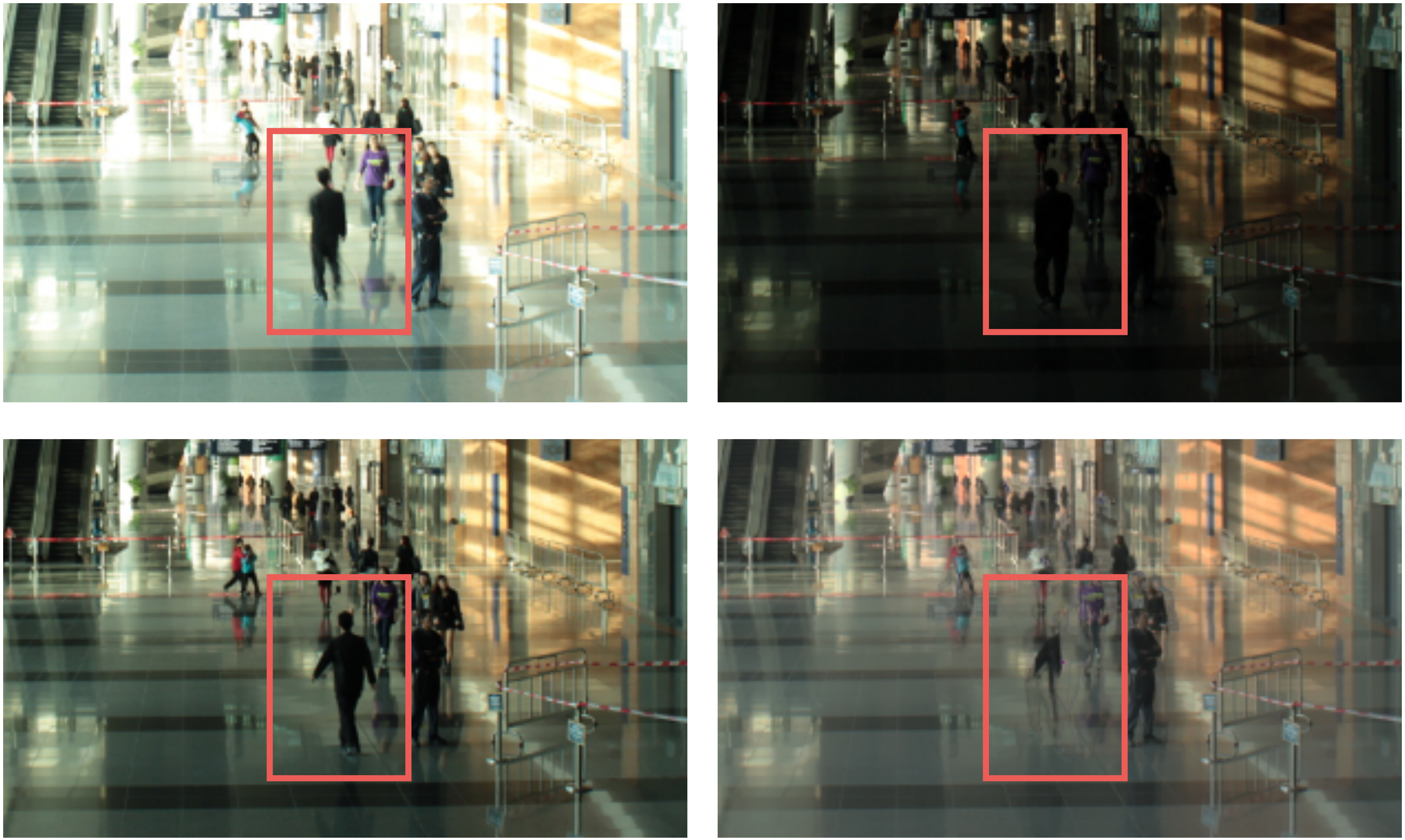
a moving scene…

Ghosts

HDR Merge



Ghosts



Deghosting: reference-based

- **Idea**: to choose an LDR image as reference, and to detect ghost based on the reference
- Selection, how?
  - Manual: select an image which has a good (from an artistic point of view) scene composition
  - Automatic: image that maximizes well-exposed pixels

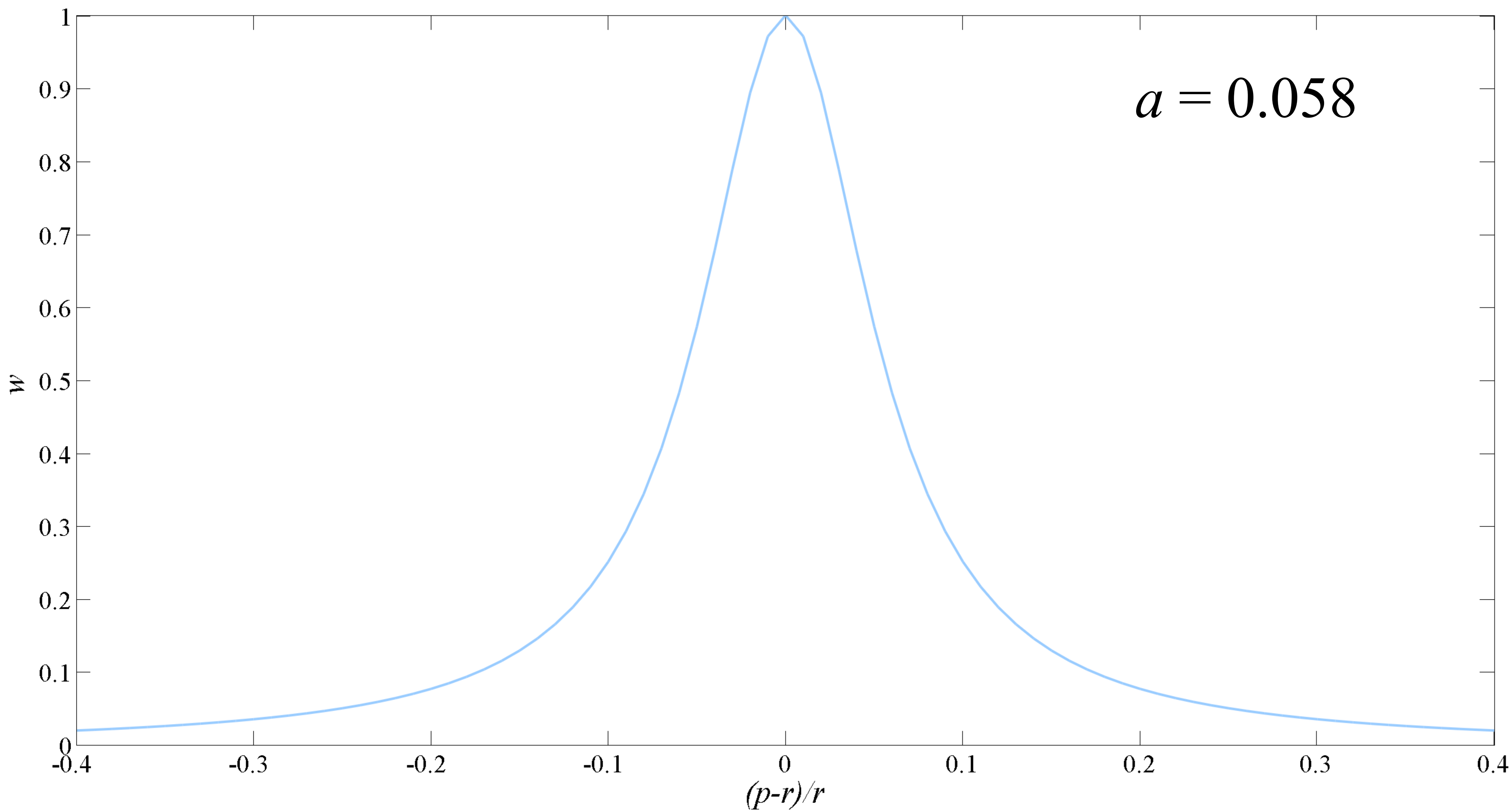
Deghosting: reference-based

- Now that we have a reference…
- Weighting other exposure images based on the selected reference —> weights to be used in the merging

$$w = \frac{a(r)^2}{a(r)^2 + \left(\frac{p-r}{r}\right)^2}$$

$$a(x) = \begin{cases} 0.058 + 0.68(x - 0.85) & \text{if } x \leq 0.85 \\ 0.04 + 0.12(1 - x) & \text{otherwise} \end{cases}$$

Deghosting: reference-based



Deghosting: reference-based

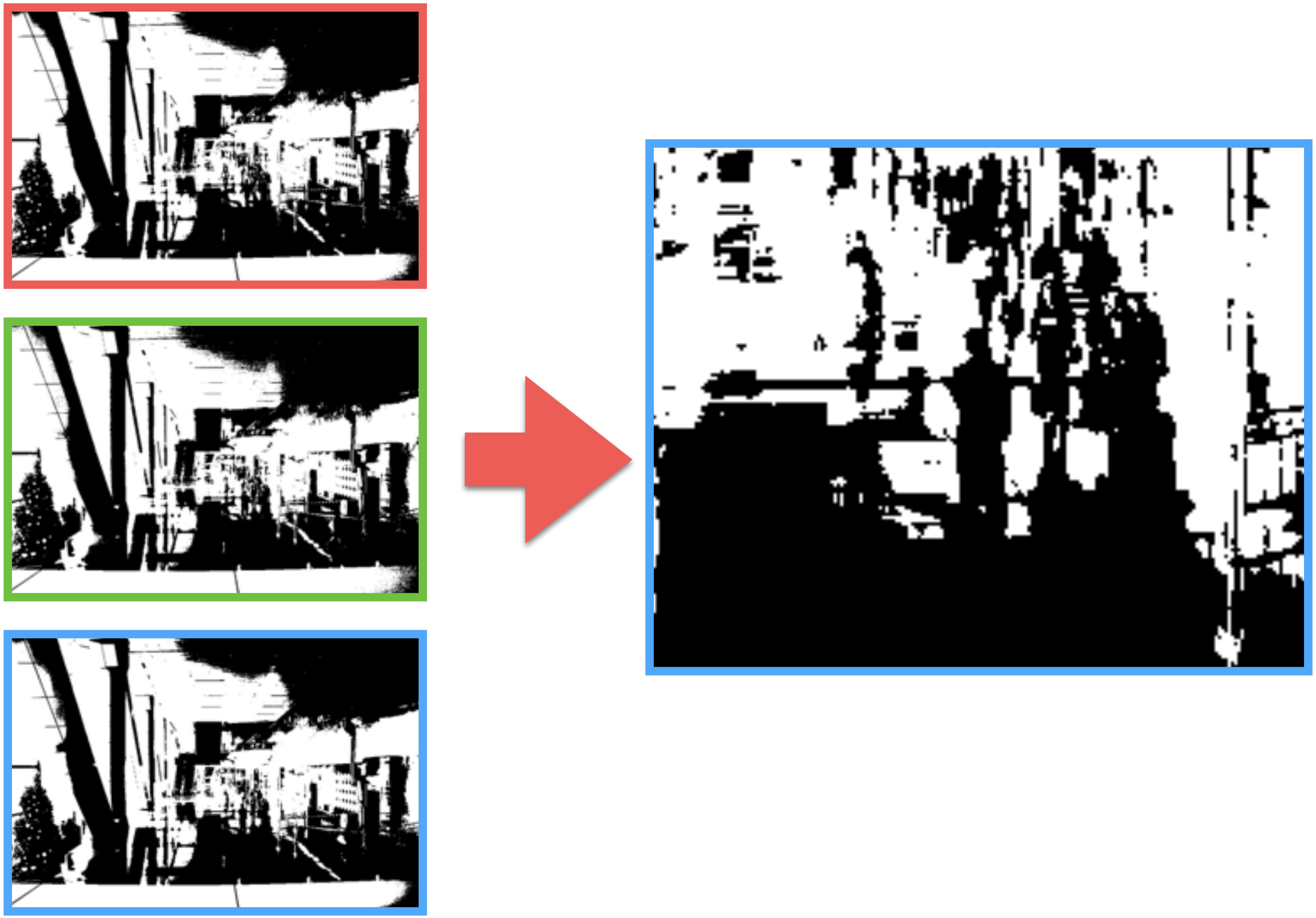
without deghosting

with deghosting

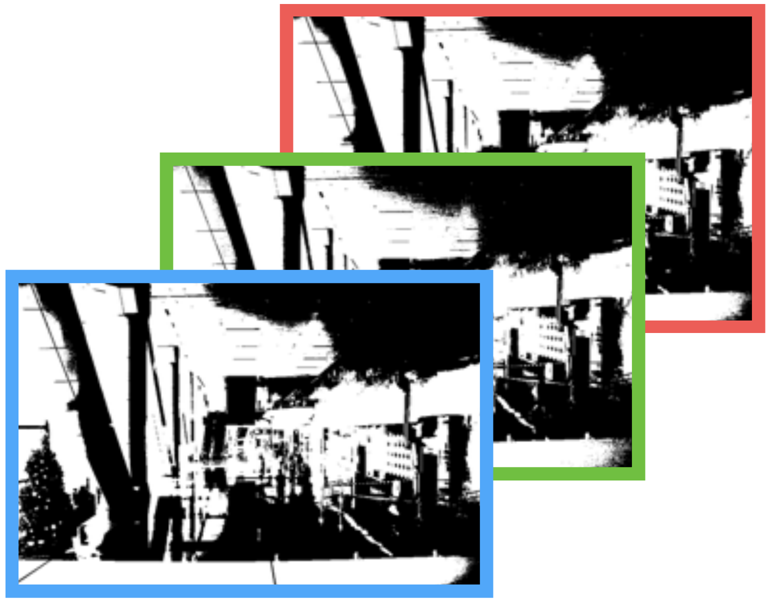
Deghosting: MTB-based

- **Idea**: the MTB descriptor is invariant
- Selection, how?
  - Manual: select an image which has a good (from an artistic point of view) scene composition
  - Automatic: image that maximizes well-exposed pixels

Deghosting:
MTB-based

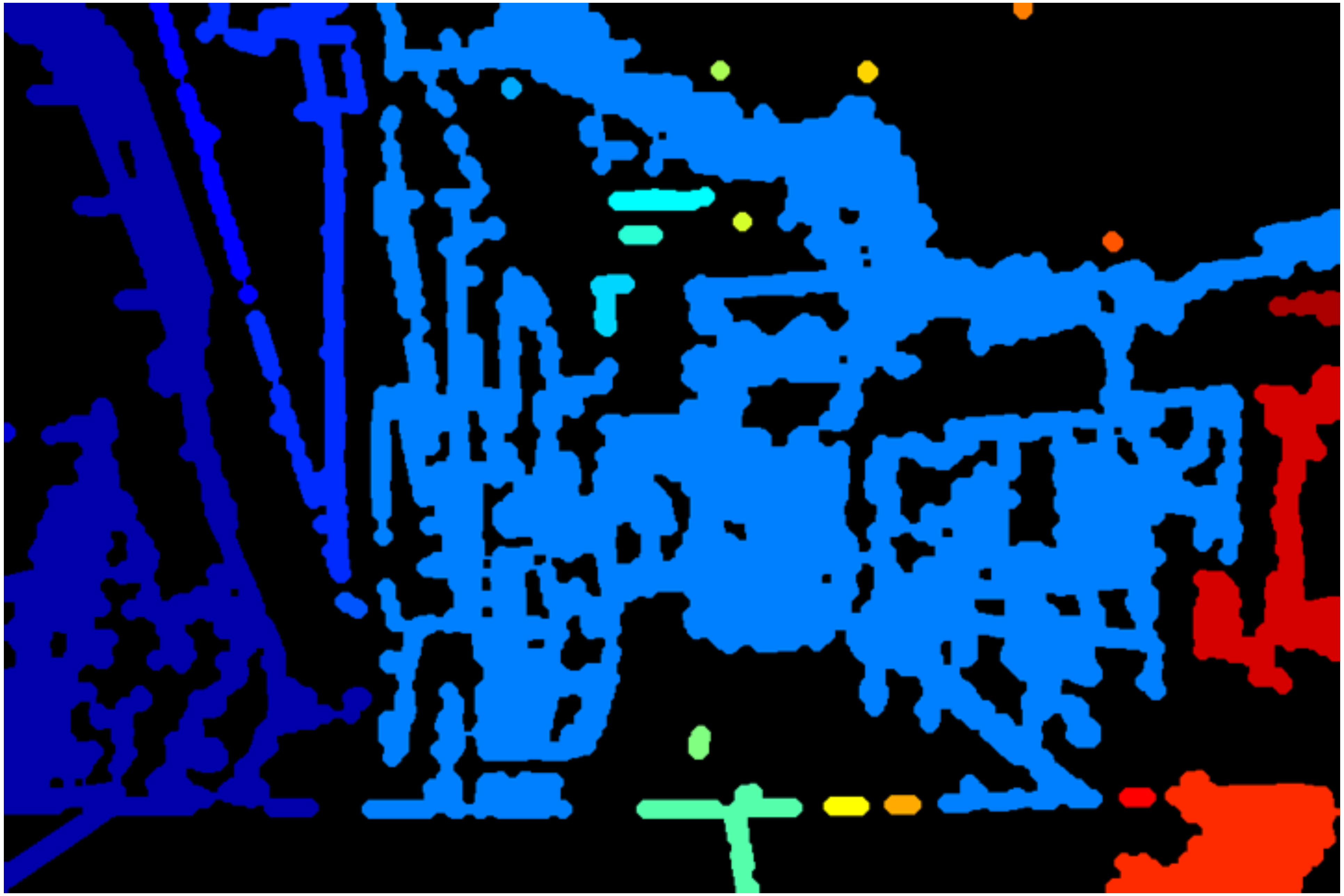


Deghosting: MTB-based



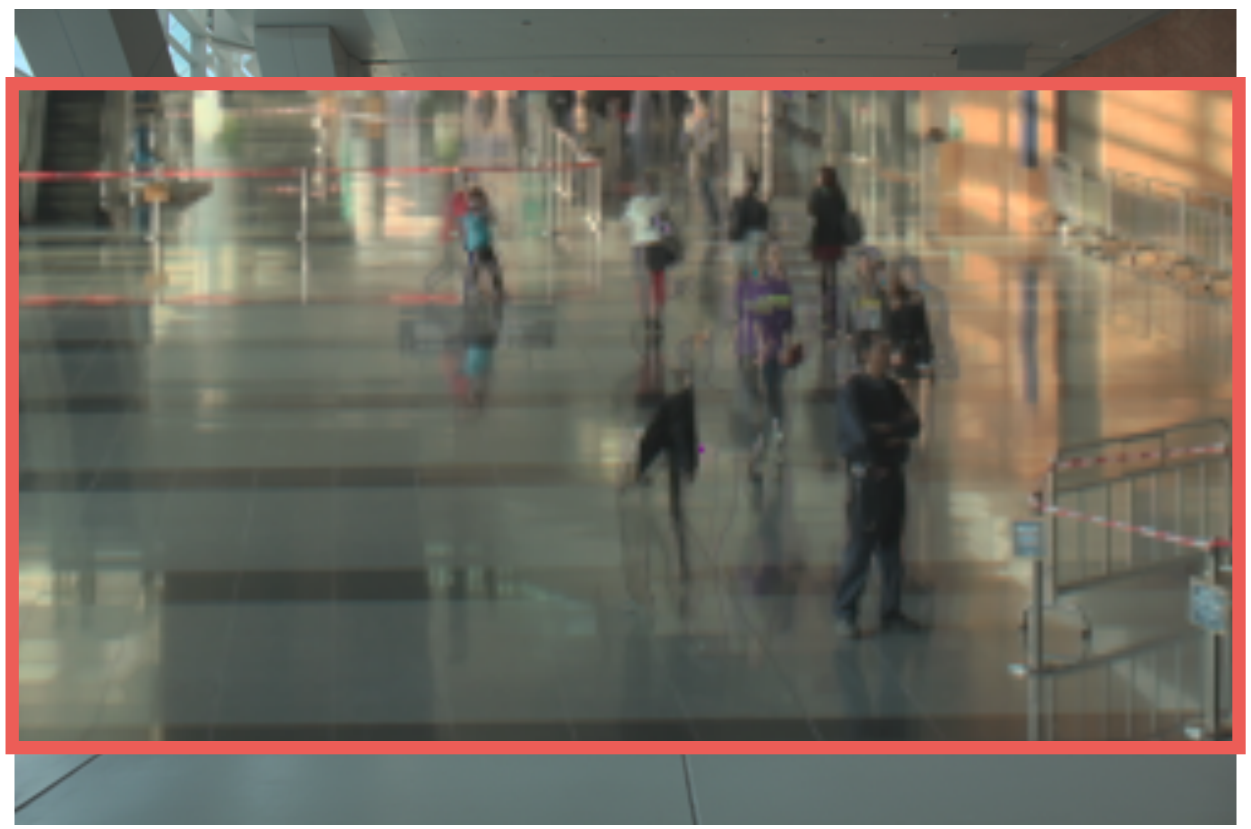
$$\text{ghost}(i,j) = \begin{cases} 1 & \text{if } M(i,j) > 0 \quad \wedge \quad M(i,j) < N \\ 0 & \text{otherwise} \end{cases}$$

Deghosting: MTB-based

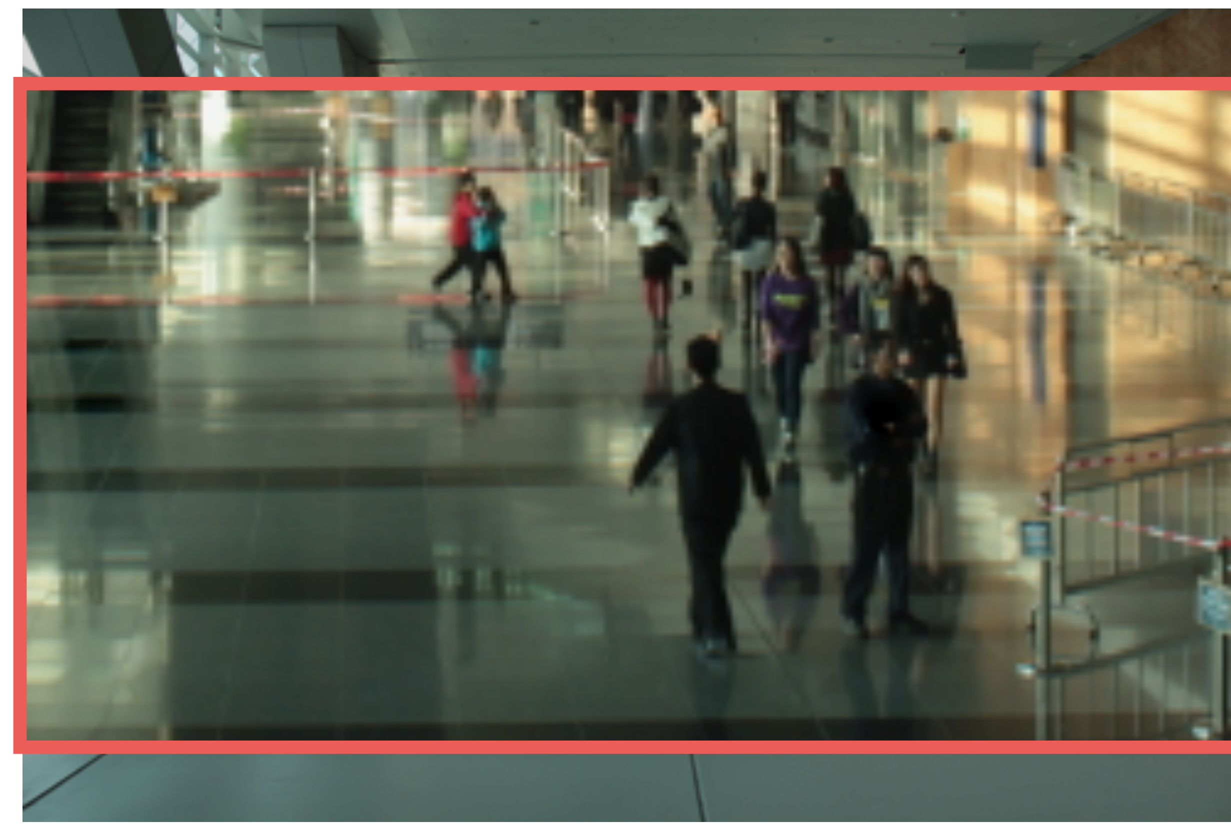


Deghosting: MTB a glimpse

- To give higher weights to better exposed blocks



without deghosting



with deghosting

Deghosting: other approaches

- Other approaches to deghosting:
  - Background extraction: many exposure images are needed to achieve good quality results
  - Optical Flow

What to do?

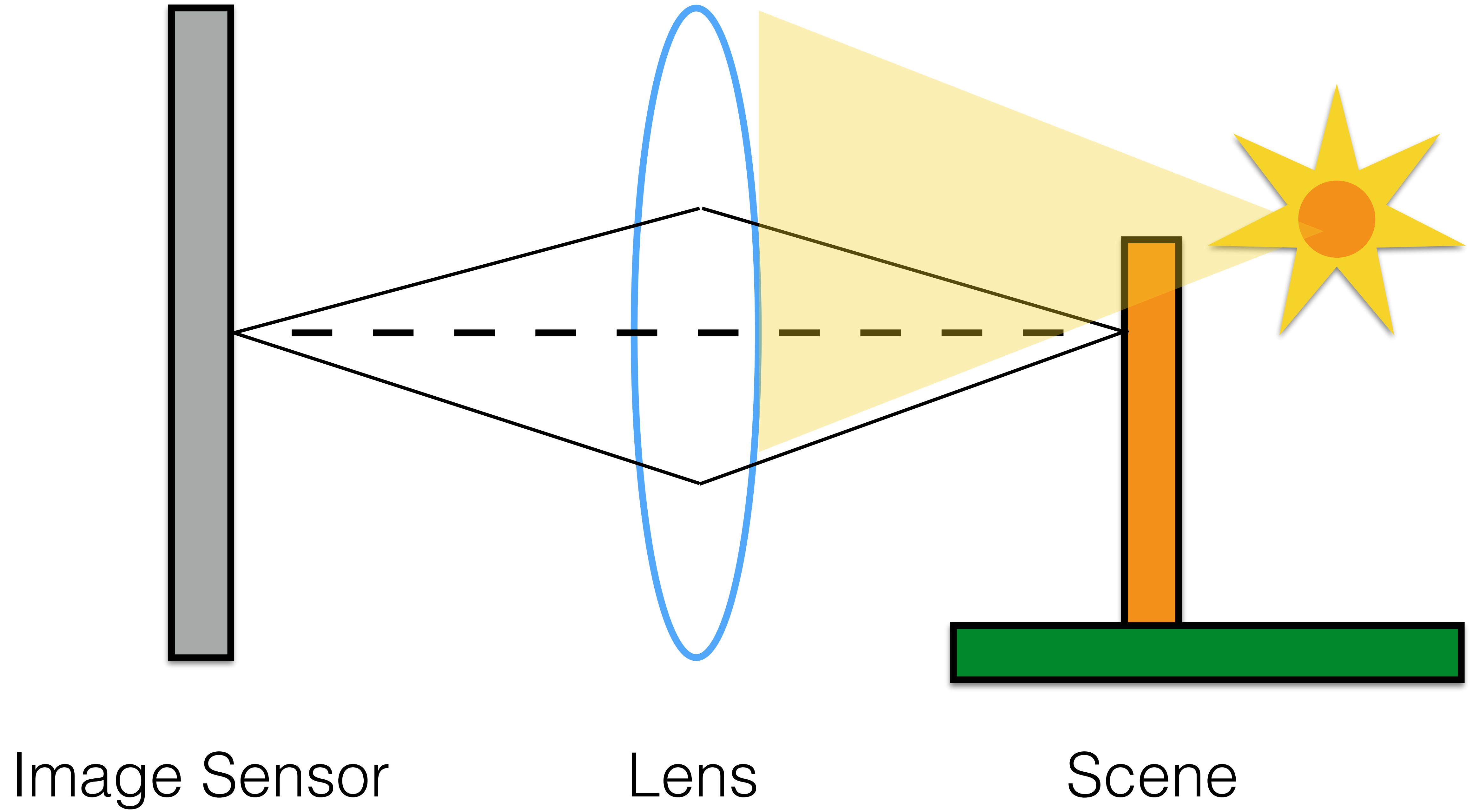
- When everything moves there is a typical strategy:
  - First step: global estimation (MTB, Local Features, etc…)
  - Second step: removing ghosts with a ghost removal technique
- This approach may be suboptimal, not solving the whole problem

lens flare…

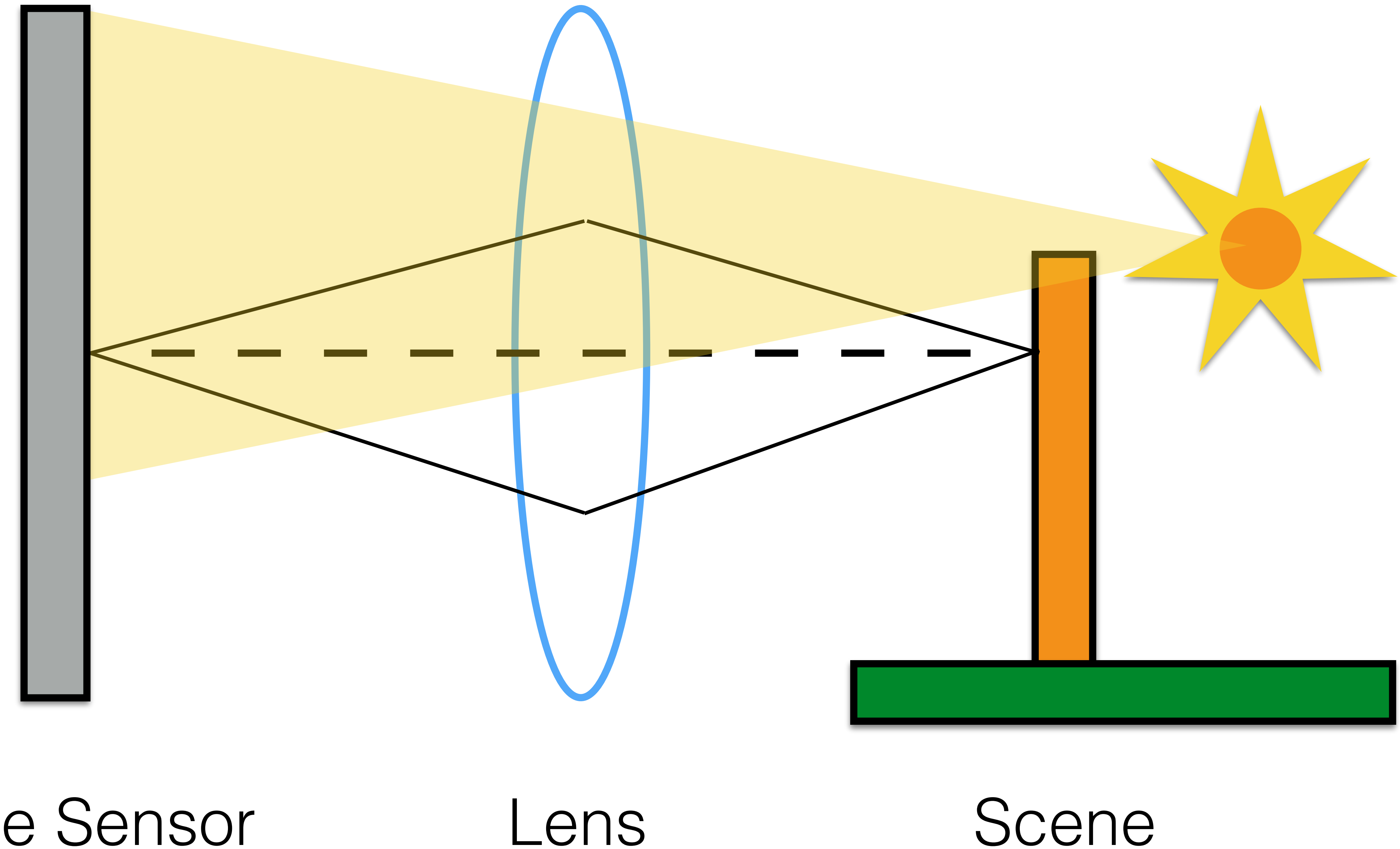
Veiling Glare

- Camera optics, lenses, are generally designed for:
  - 2-3 orders of magnitude
  - 24-bit sensors or 35mm film

Veiling Glare



Veiling Glare



Veiling Glare

- OK, we have more light that should be there… what is the real problem?
    - Reducing the dynamic range of the scene!

Veiling Glare

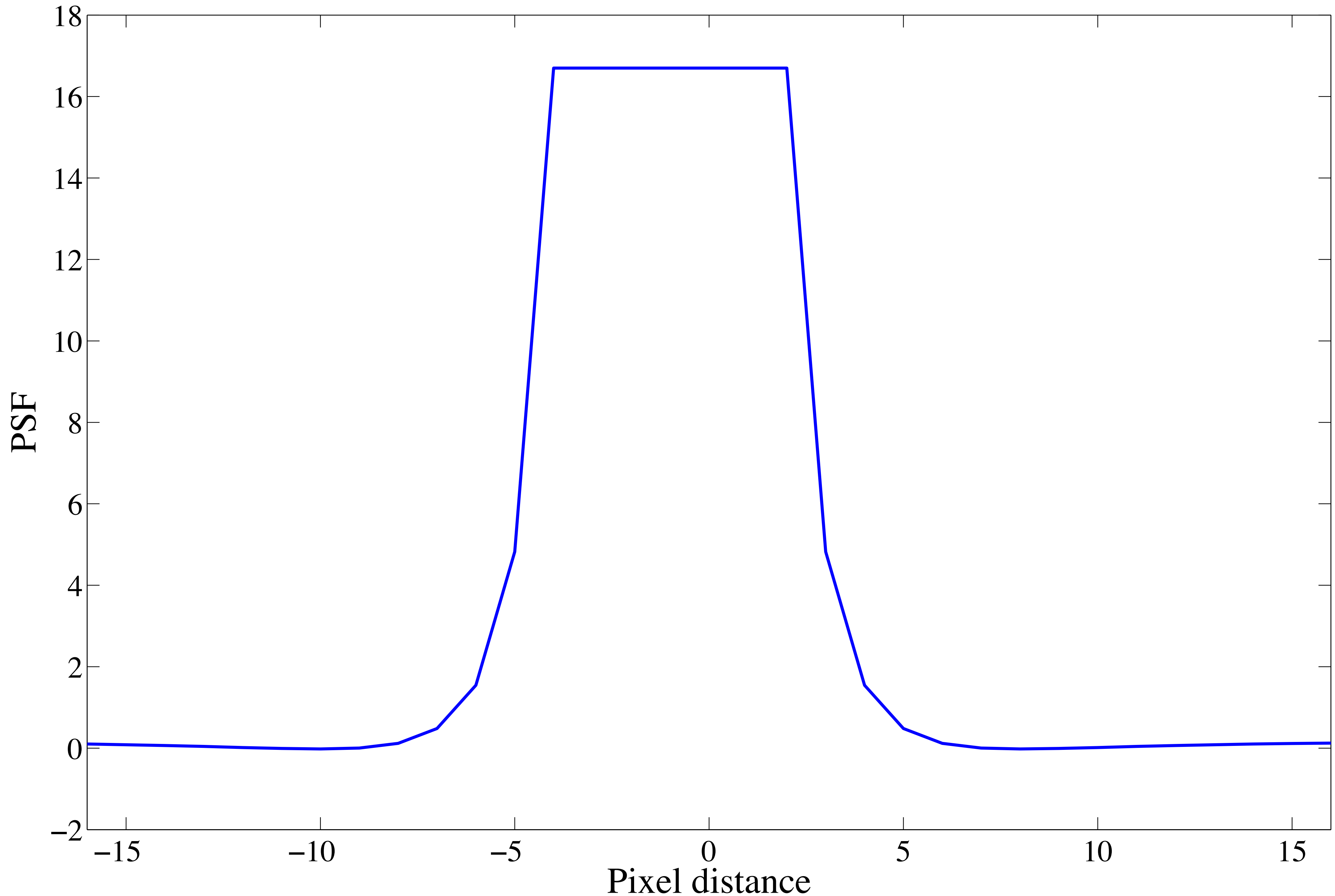
Veiling Glare:
A Capturing Approach

- Characterization of the glare of a particular camera
- Special glare capturing
- Glare removal

Veiling Glare: Characterization

- Measuring the glare of a camera at given aperture:
  - dark room
  - point light source; e.g. LED
  - capturing an HDR image

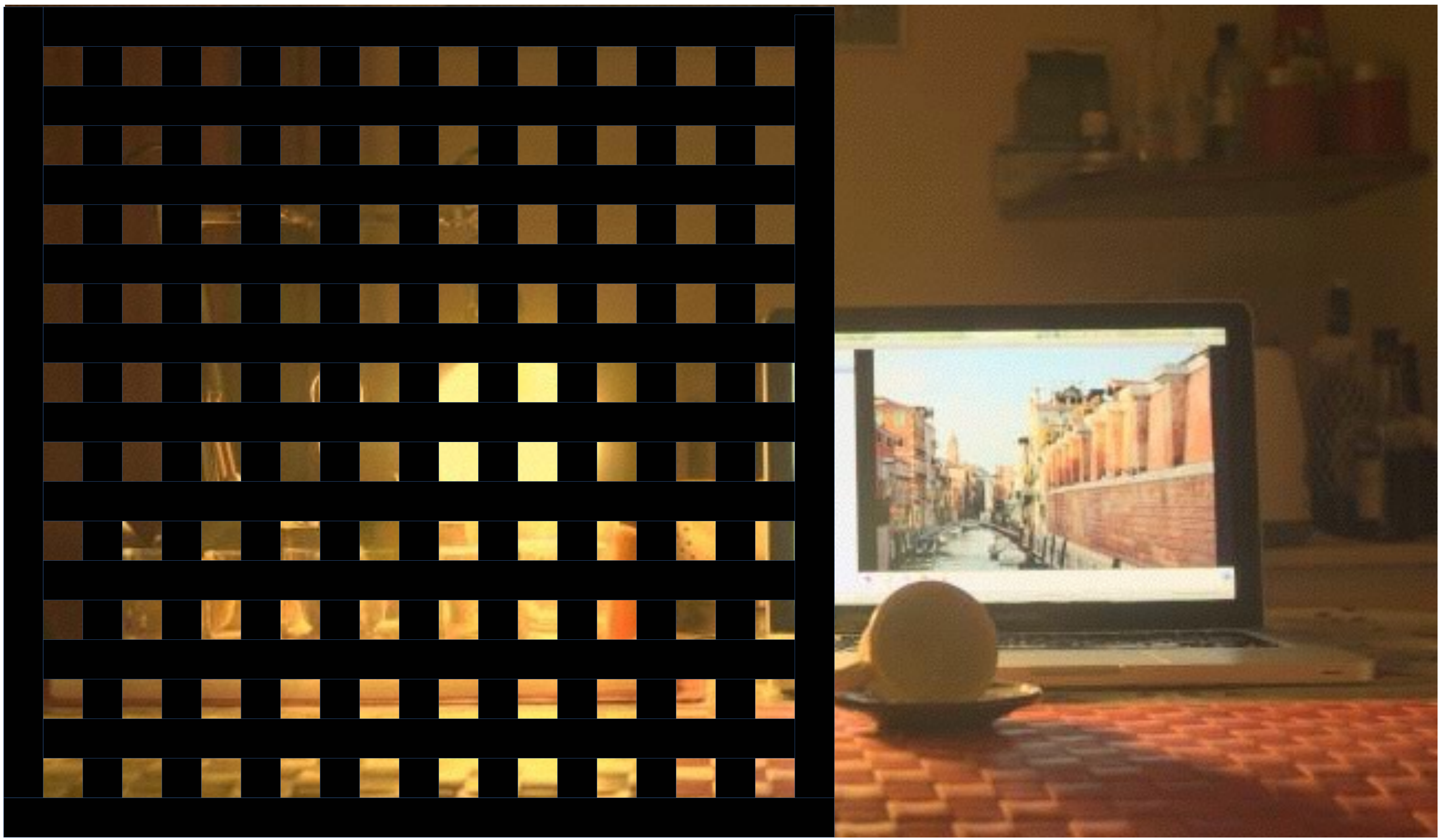
Veiling Glare: Characterization



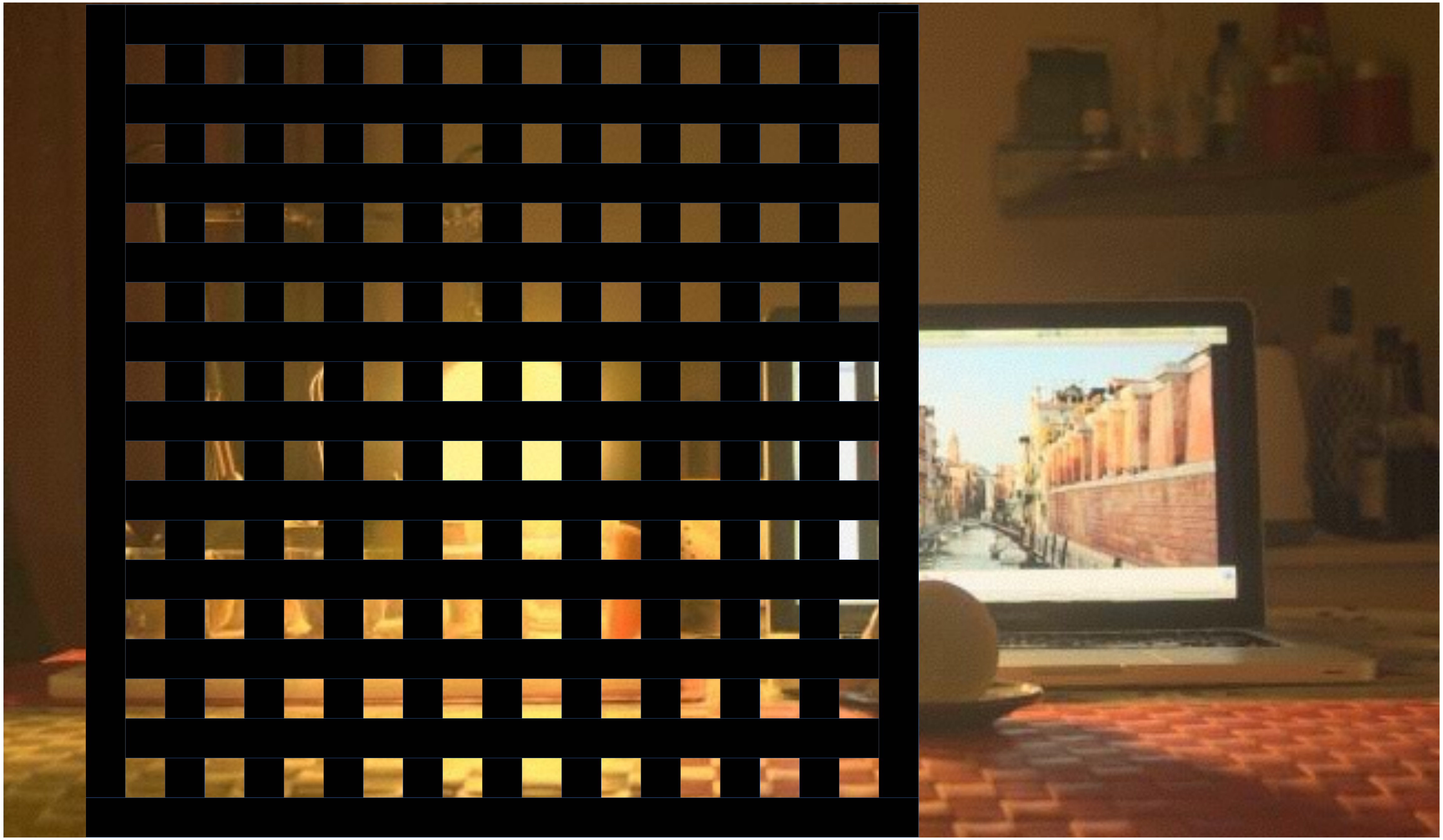
Veiling Glare: Acquisition

- Block glaring mask in front o the camera, e.g. a 30x30 mask
- Moving the mask in X and Y planes
- 6x6 HDR captures —> a lot of data!

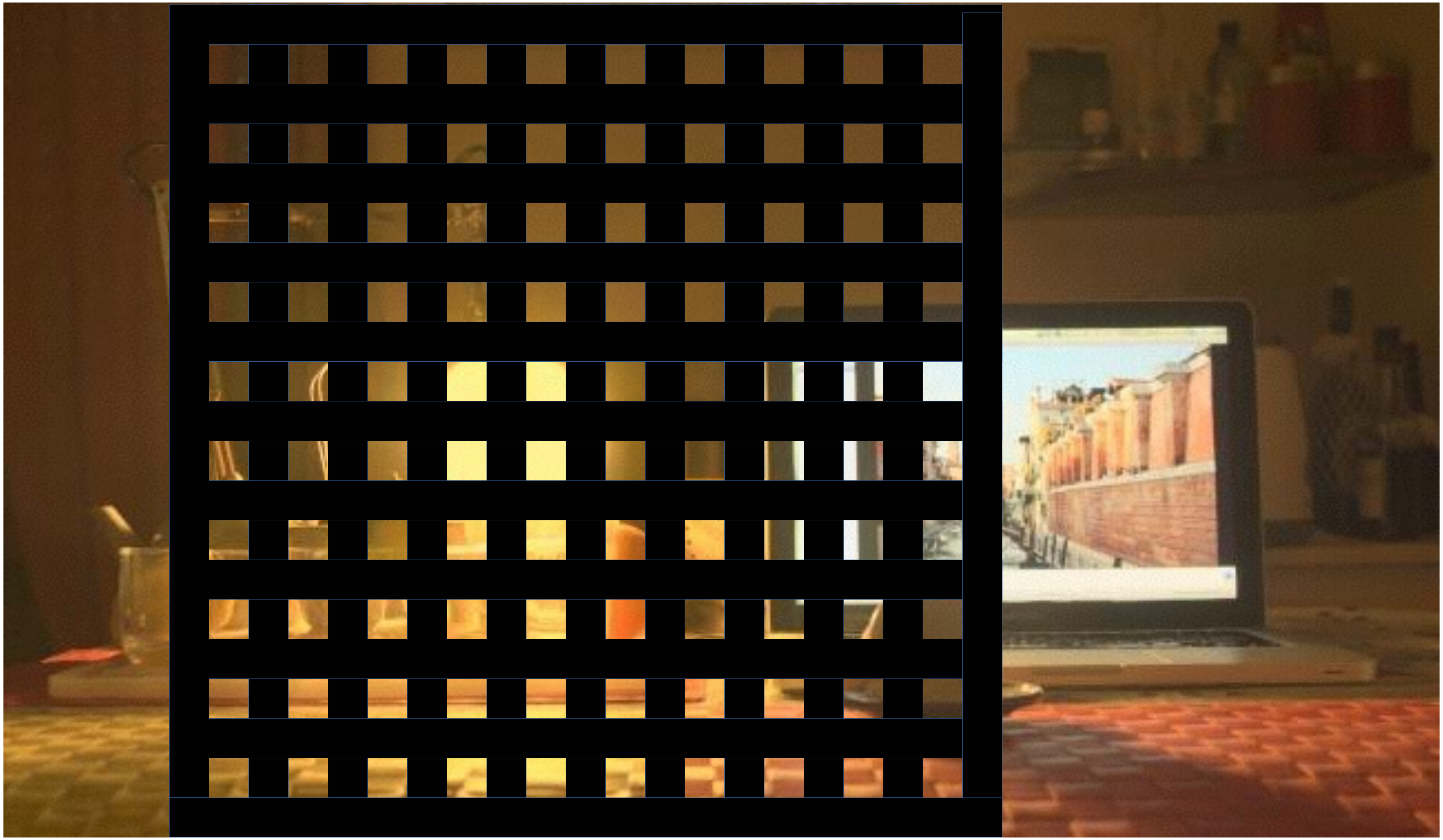
Veiling Glare: capturing approach



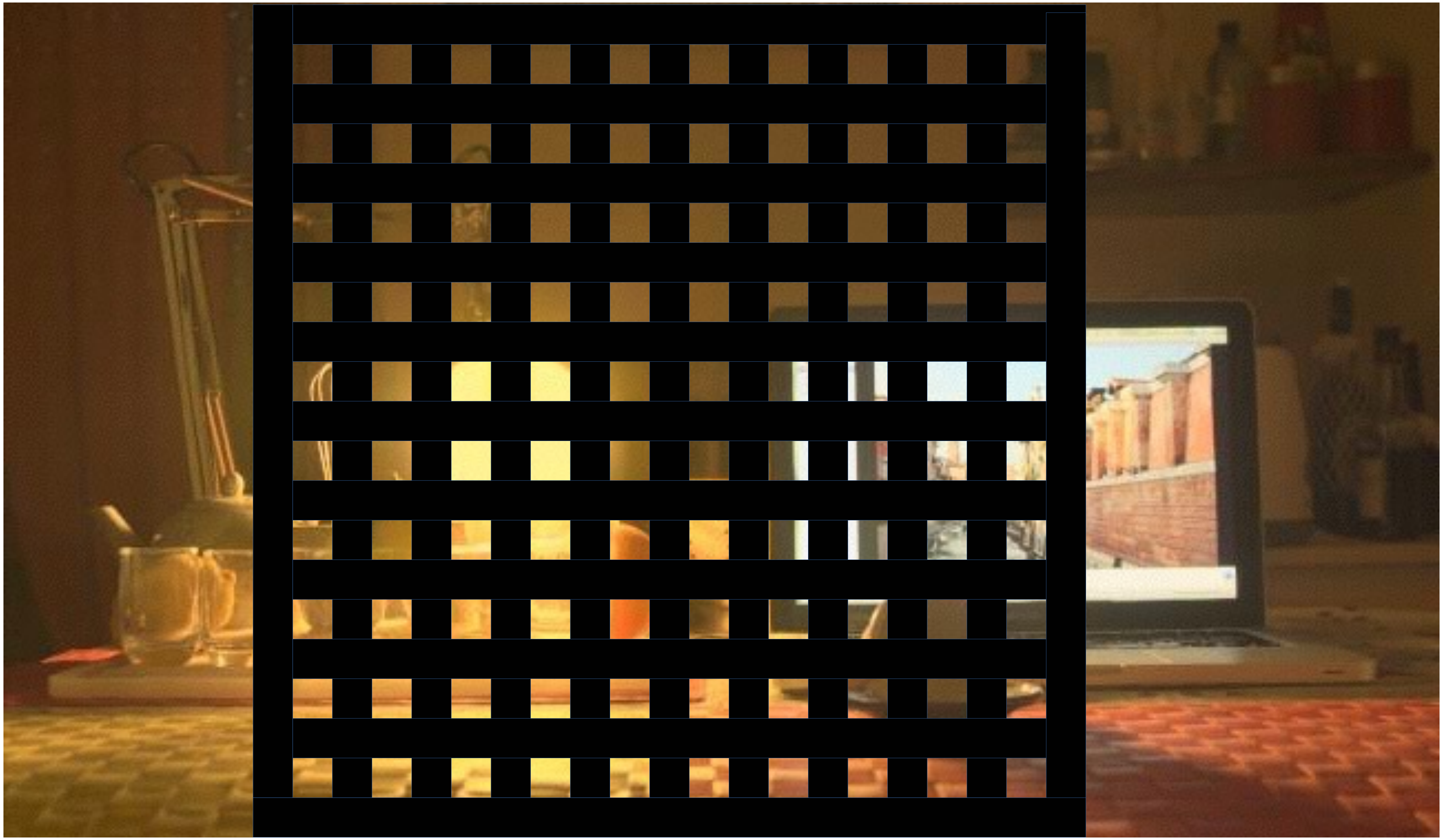
Veiling Glare: capturing approach



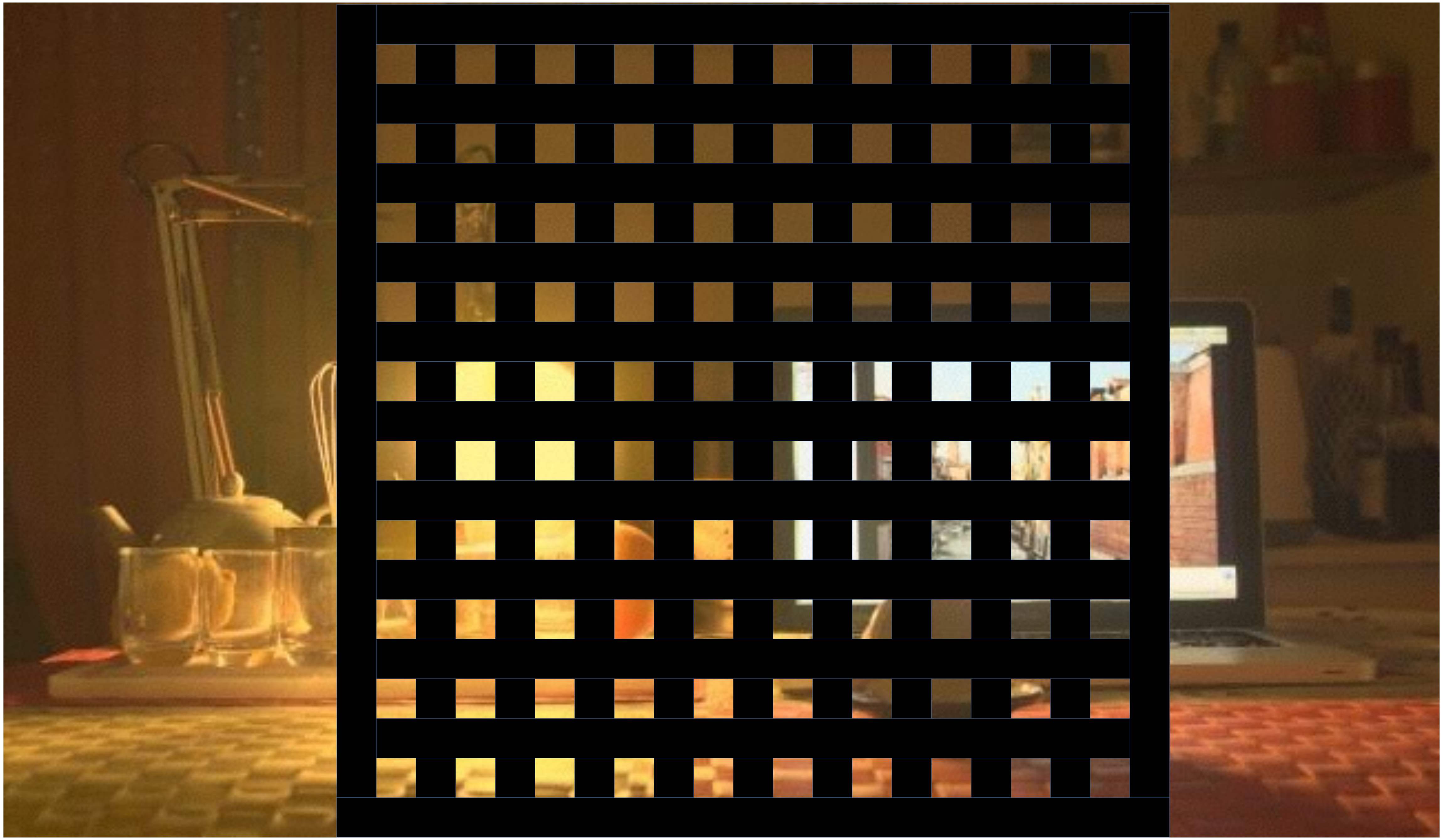
Veiling Glare: capturing approach



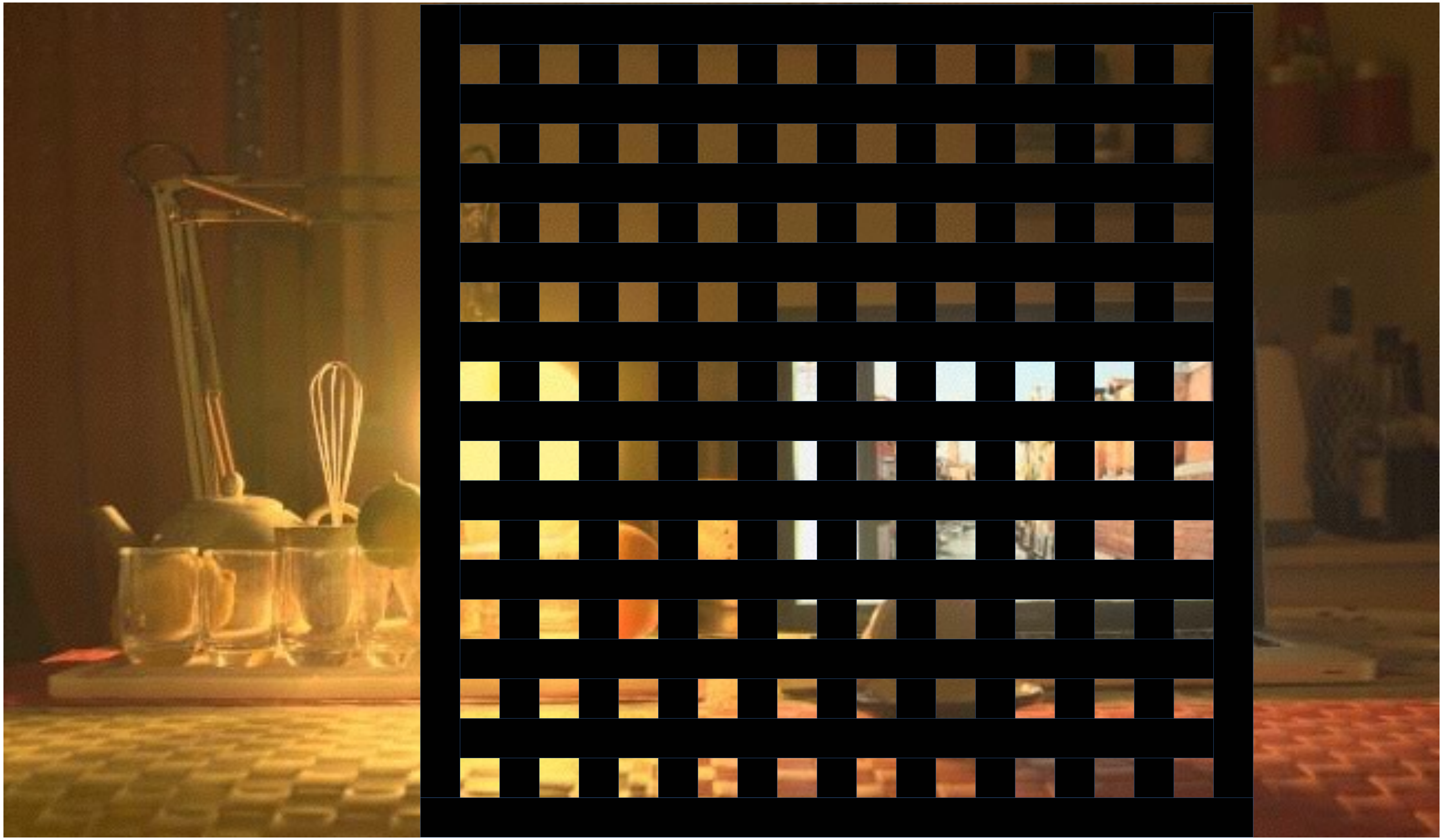
Veiling Glare: capturing approach



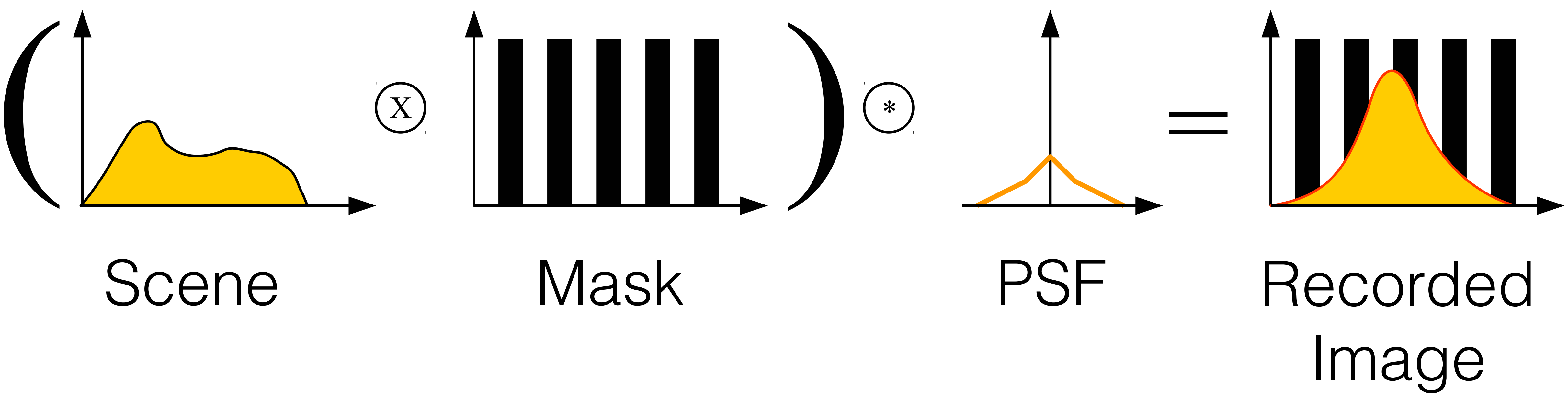
Veiling Glare: capturing approach



Veiling Glare: capturing approach



Veiling Glare: glare removal



For removing glare, this process has to be inverted!

Veiling Glare: results

from the paper “Veiling glare high dynamic range imaging”. Eino-Ville Talvala, Andrew Adams, Mark Horowitz, Marc Levoy. ACM SIGGRAPH 2007 Papers Program.

Veiling Glare: a post-processing approach

- The previous method produces high quality results!
- There are some disadvantages:
    - Many pictures to take
    - The scene has to be static
    - Characterization of the PSF of the camera

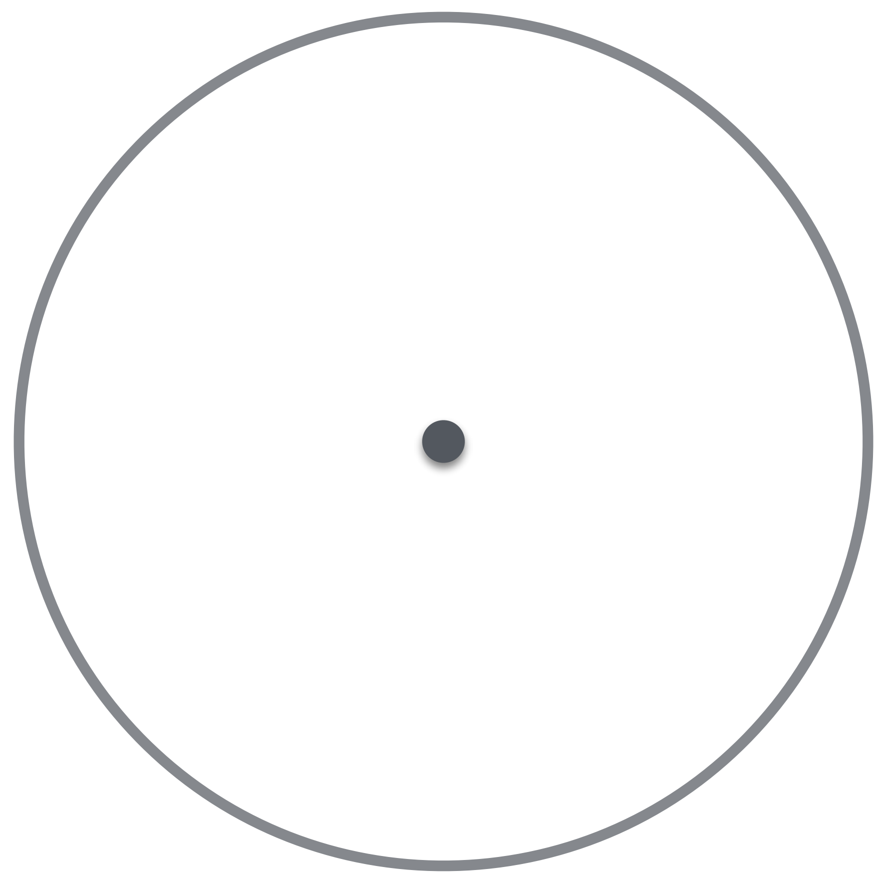
Veiling Glare: a post-processing approach

- Main steps:
  - Estimate the PSF
  - Generate the glare image
  - Remove the glare image

Veiling Glare: PSF Estimation

- Compute image luminance, L
- Threshold L to identify:
  - hot pixels (bright ones); source of glare
  - dark pixels (dark ones); “veiled”

Veiling Glare:
PSF Estimation



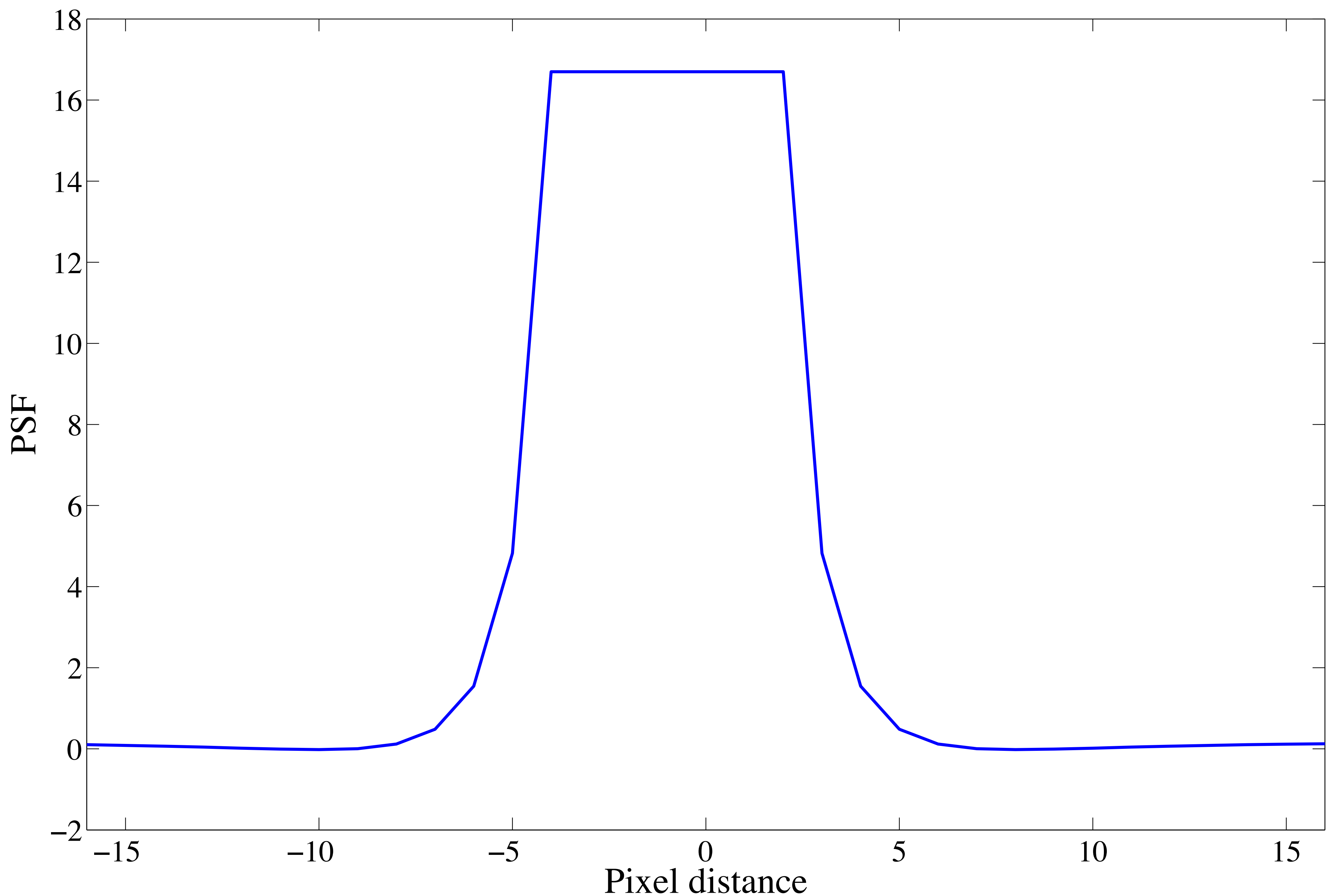
Veiling Glare:
PSF Estimation

$$P_i = \sum_j P_j \left(C_0 + \frac{C_1}{r_{ij}} + \frac{C_2}{r_{ij}^2} + \frac{C_2}{r_{ij}^3} \right)$$

- where r_{ij} is the distance between the hot pixel P_j and the minimum pixel P_i.

$$P_i = C_0 \sum_j P_j + C_1 \sum_j \frac{P_j}{r_{ij}} + C_2 \sum_j \frac{P_j}{r_{ij}^2} + C_3 \sum_j \frac{P_j}{r_{ij}^3}$$

Veiling Glare: PSF Estimation



Veiling Glare: Removing Glare

- Input: I_{cr} (image with glare), PSF
- Output: I_{out} (image glare-free)
- Algorithm:
  - Create a black image, F_{cr}
  - For each hot pixel in $I_{cr,}$ multiply by PSF and add the contribution to F_{cr}
  - $I_{out} = I_{cr} - F_{cr}$

Veiling Glare:
Glare Image

Veiling Glare: Removing Glare

Questions?