

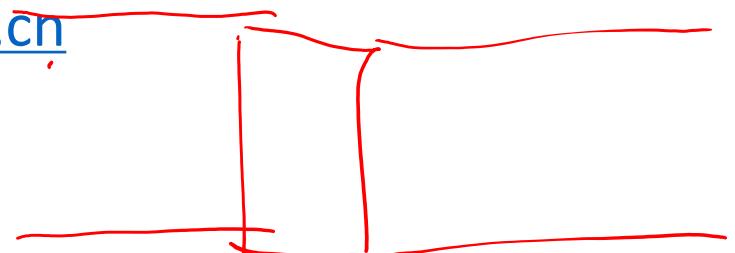
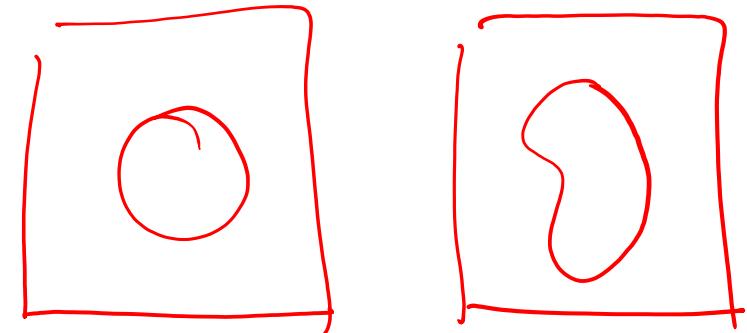
Lecture 27 Image inpainting

outpainting

Yuyao Zhang PhD

zhangyy8@shanghaitech.edu.cn

SIST Building 2 302-F



Problem to solve

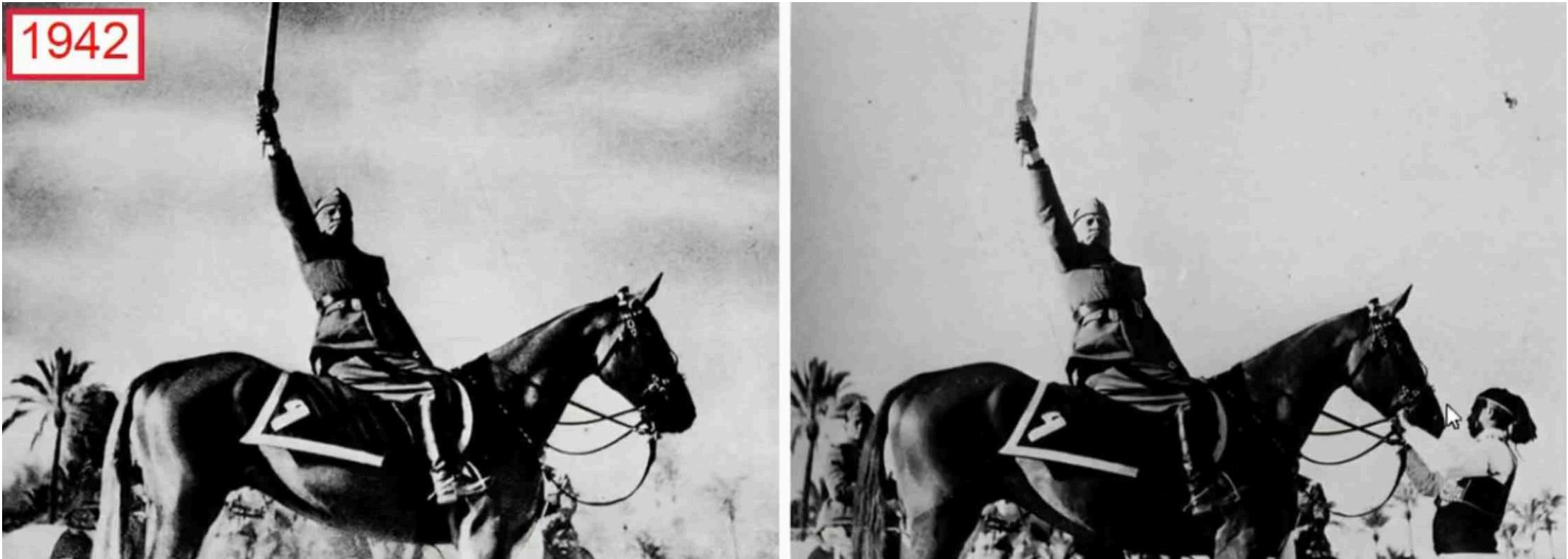
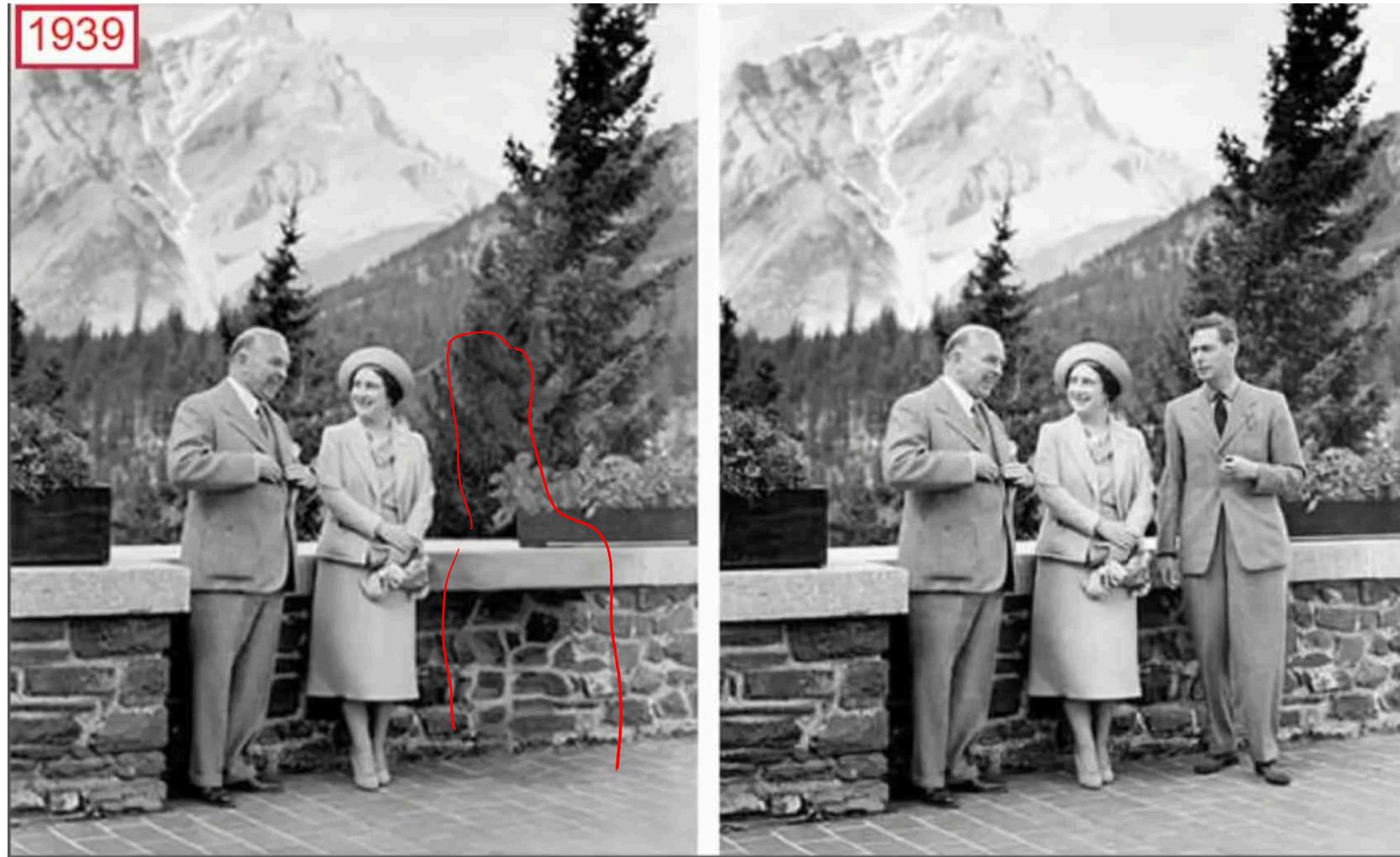


Image inpainting



Outline

- Photomontage
- PDE based image inpainting
- Patch based image inpainting

Image photomonatge

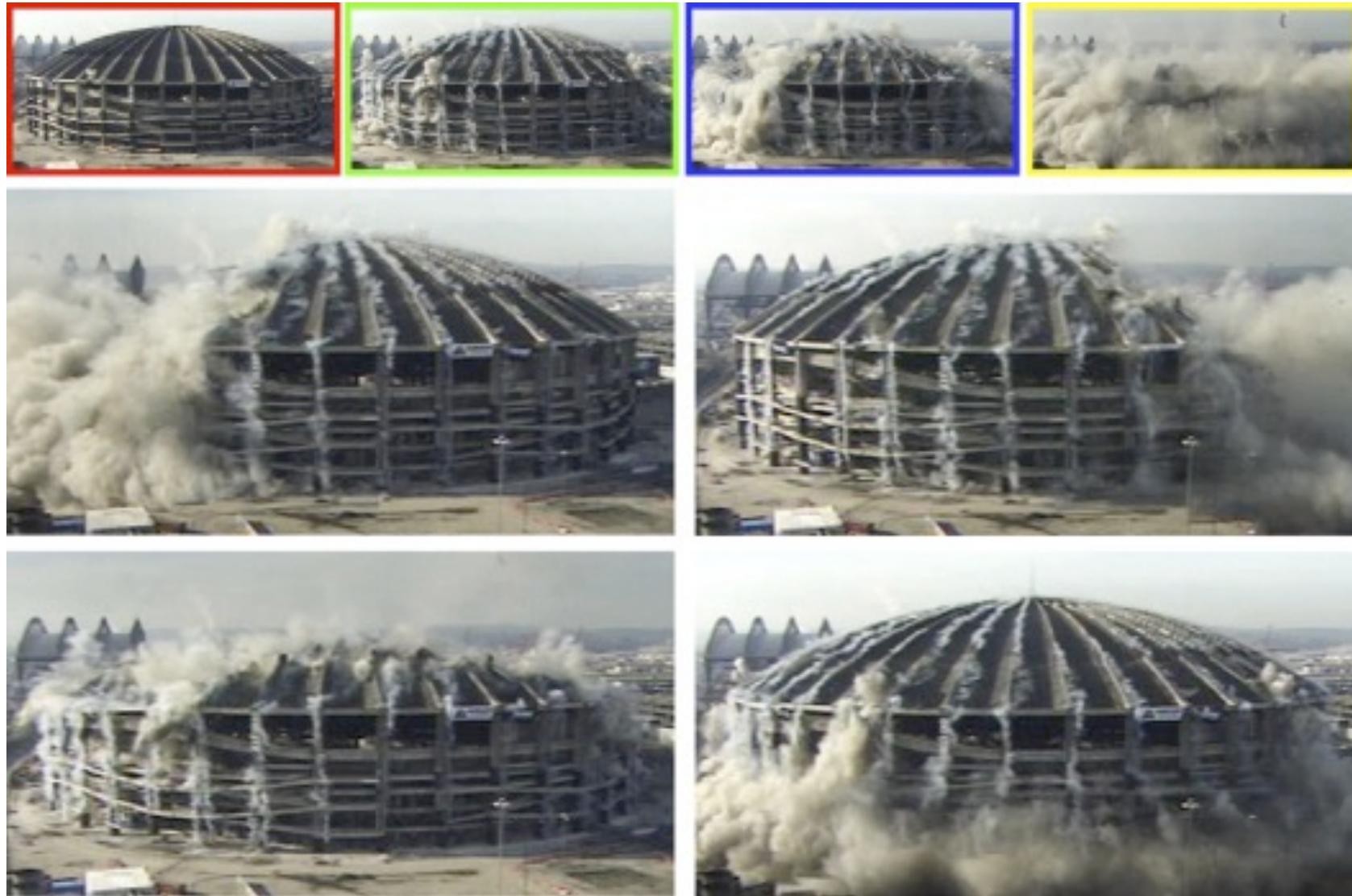


Image photomontage

- Key problem : find a good seam (dividing line) between 2 images with similar scene, so that the intensity difference across seam is imperceptible.



Image photomontage

- Key problem : find a good seam (dividing line) between 2 images with similar scene, so that the intensity difference across seam is imperceptible.



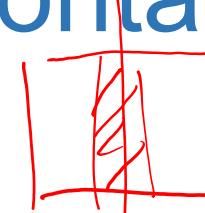
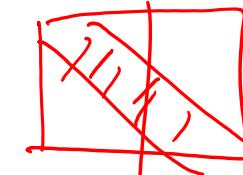
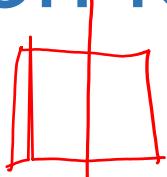
Penalty function for image photomontage



- Key problem : find a good seam (dividing line) between 2 images with similar scene, so that the intensity difference across seam is imperceptible.
- Idea: “cost” of drawing line between pixels i and j . Where $S(i)$ denote a pixel in the source image; $T(i)$ denote a pixel in the target image

$$C(i, j) = \|S(i) - T(i)\| - \|S(j) - T(j)\|$$

Penalty function for image photomontage

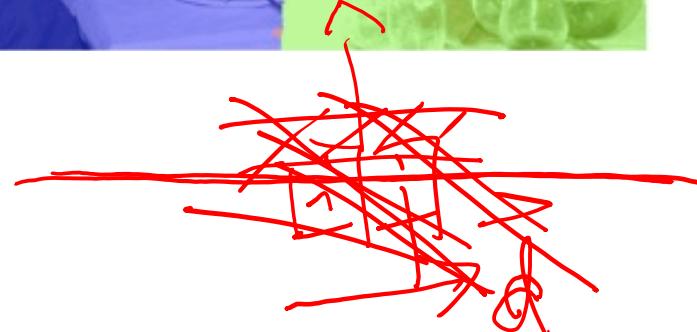
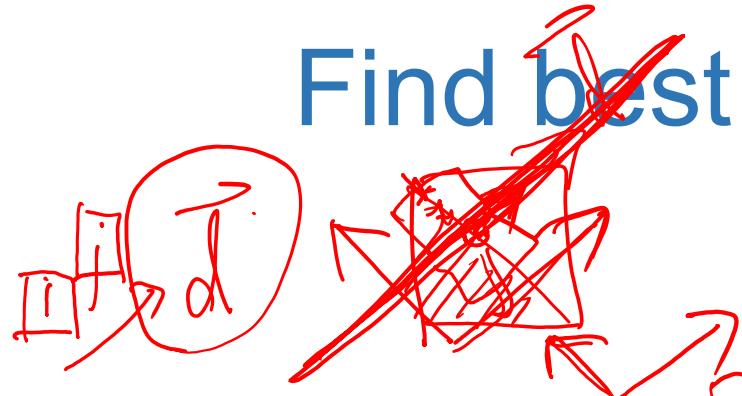


- Key problem : find a good **seam (dividing line)** between 2 images with similar scene, so that the intensity difference across seam is imperceptible.
- Idea: “cost” of drawing line between pixels i and j . Where $S(i)$ denote a pixel in the source image; $T(i)$ denote a pixel in the target image

Source Target

$$C(i, j) = \underbrace{\|S(i) - T(i)\|}_{\text{Source}} - \underbrace{\|S(j) - T(j)\|}_{\text{Target}}$$

Find best seam via graph cut



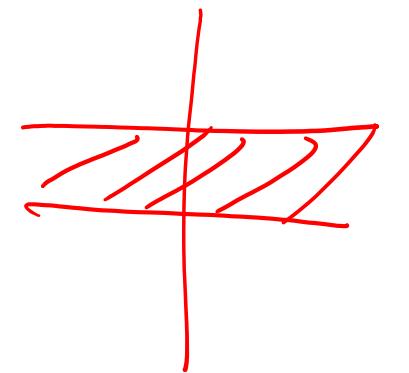
- We could modify this with image gradient:

$$w_{ij} = \frac{\|S(i) - T(i)\| - \|S(j) - T(j)\|}{|\vec{d} \cdot \nabla S(i)| + |\vec{d} \cdot \nabla T(i)| + |\vec{d} \cdot \nabla S(j)| + |\vec{d} \cdot \nabla T(j)|} \rightarrow \infty$$

Numerator: pixels color should be the same on either side. ∇ large $\rightarrow 0$

Denominator: \vec{d} is the vector pointing from i to j . Weight will be small if seam passed through gradient (parallel to edge). Weight will be large if seam passed along gradient (perpendicular to edge). Smooth areas will be in-between.

- Idea: “scribble” on regions of images that we want to keep in the final composite, let graph cut algorithm decide what to do with pixels.



Find best seam via graph cut

- We could modify this with image gradient:



$$w_{ij} = \frac{\|S(i) - T(i)\| - \|S(j) - T(j)\|}{|\vec{d} \cdot \nabla S(i)| + |\vec{d} \cdot \nabla T(i)| + |\vec{d} \cdot \nabla S(j)| + |\vec{d} \cdot \nabla T(j)|}$$

Numerator: pixels color should be the same on either side.

Denominator: \vec{d} is the vector pointing from i to j . Weight will be small if seam passed through gradient (parallel to edge). Weight will be large if seam passed along gradient (perpendicular to edge). Smooth areas will be in-between.

- Idea: “scribble” on regions of images that we want to keep in the final composite, let graph cut algorithm decide what to do with pixels.

Image in-painting

- E.g. wire removal, removing artifact/creases from old portraits.
- Two easiest approaches:
- Partial Differential Equations (PDE).
- Patch-based methods.

PDE in-painting method

- A differential equation that tries to “punch” good colors from the boundary into the hole.
- The most important thing about image: image edges.
- Edges are particular important, we want to make sure they continue into the hole.
- Key equation:

$$\nabla L \cdot \vec{N} = 0$$

L : information we want to propagate; \vec{N} direction.



PDE in-painting method

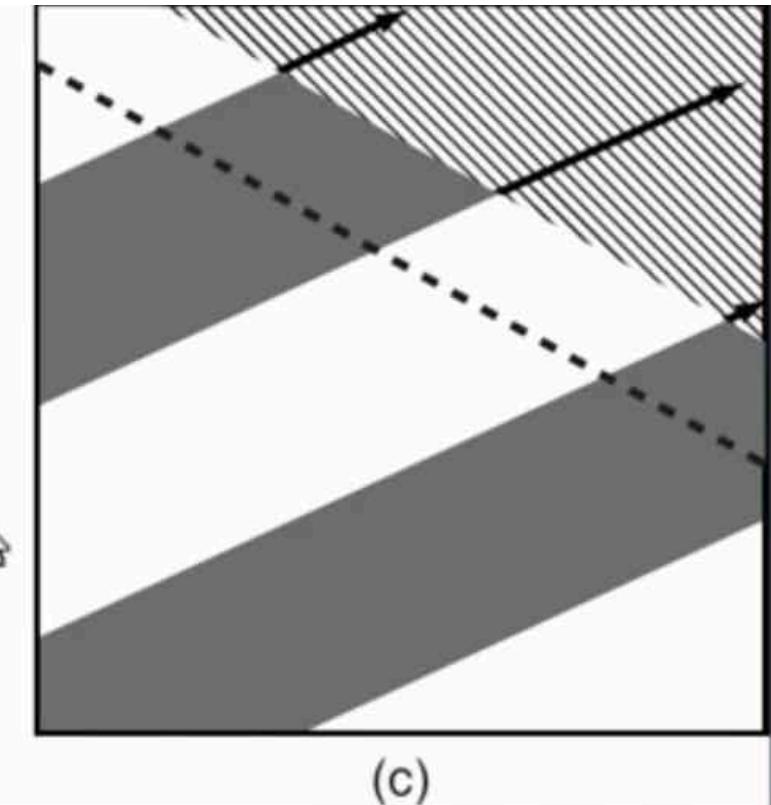
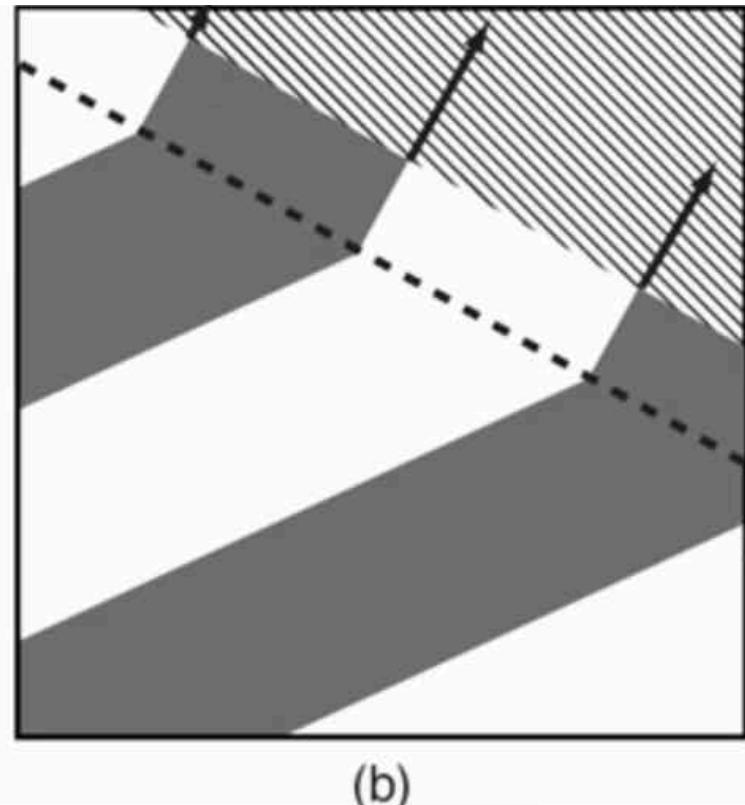
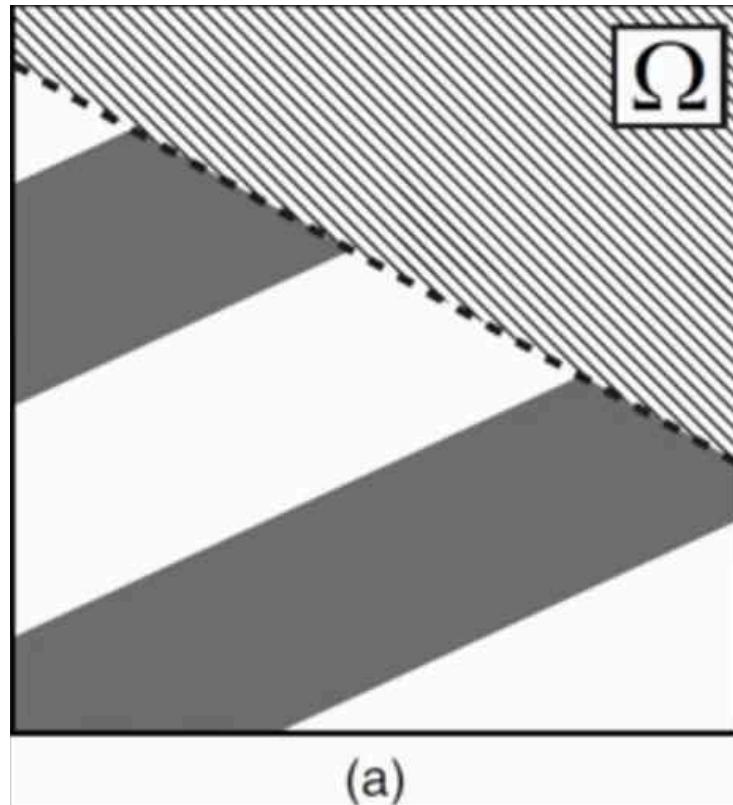
- A differential equation that tries to “punch” good colors from the boundary into the hole.
- The most important thing about image: image edges.
- Edges are particular important, we want to make sure they continue into the hole.
- Key equation:

$$\nabla L \cdot \vec{N} = 0$$

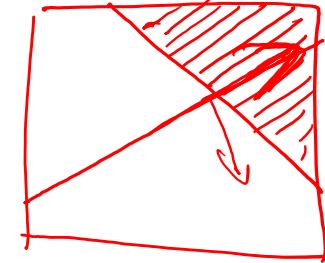
L : information we want to propagate; \vec{N} direction.



PDE inpainting method



PDE inpainting method



➤ E.G.: We define L=smoothness estimator; \vec{N} = isophote direction:

$$\nabla(\nabla^2 I(x, y)) \cdot \nabla^P I(x, y) = 0, (x, y) \in \Omega$$

~~$I(x, y)$~~

➤ Laplacian edge; the change in the edge; isophote direction (the direction going along the edge).

➤ Interpretation: change in Laplacian along isophote direction should be zeros. Want to preserve edge of ‘arrival’.

➤ Solution: $\frac{\partial I}{\partial t} = \nabla(\nabla^2 I(x, y)) \cdot \nabla^P I(x, y)$

General algorithm

$$\frac{\partial I}{\partial t} = \nabla(\nabla^2 I(x, y)) \cdot \nabla^P I(x, y)$$

edge color

➤ Iteratively

- Continue edge of surrounding area into hole;
- Continue color of surrounding area into hole;

$$I_i(x, y) = I_{i-1}(x, y) + \nabla(\nabla^2 I_{i-1}(x, y)) \cdot \nabla^P I_{i-1}(x, y)$$

PDE method performance



PDE method performance



PDE method performance

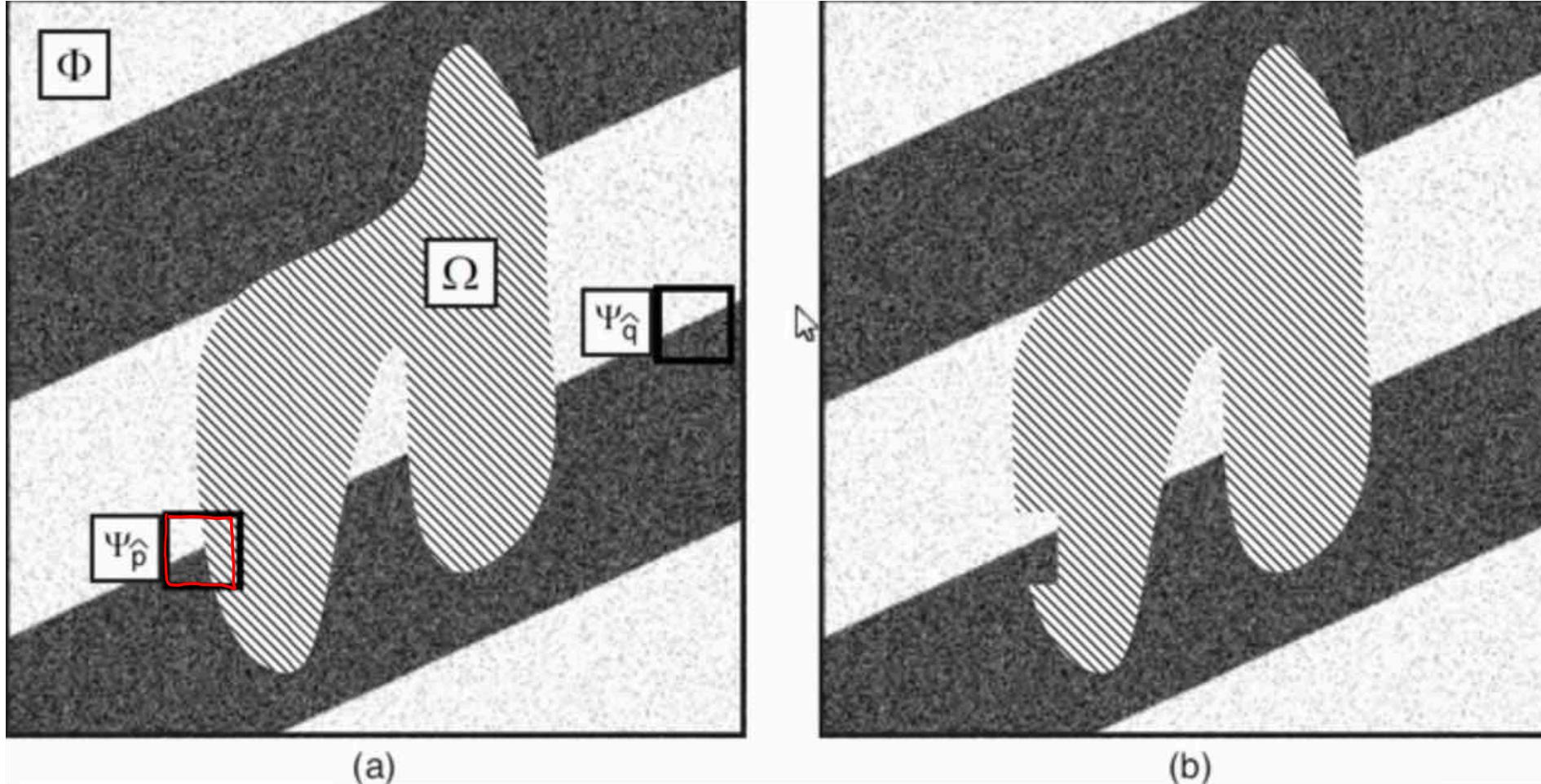


An example when PDE doesn't work



- PDE- based method works OK for very thin holes, and low-texture regions. But can' t hallucinate high detail.

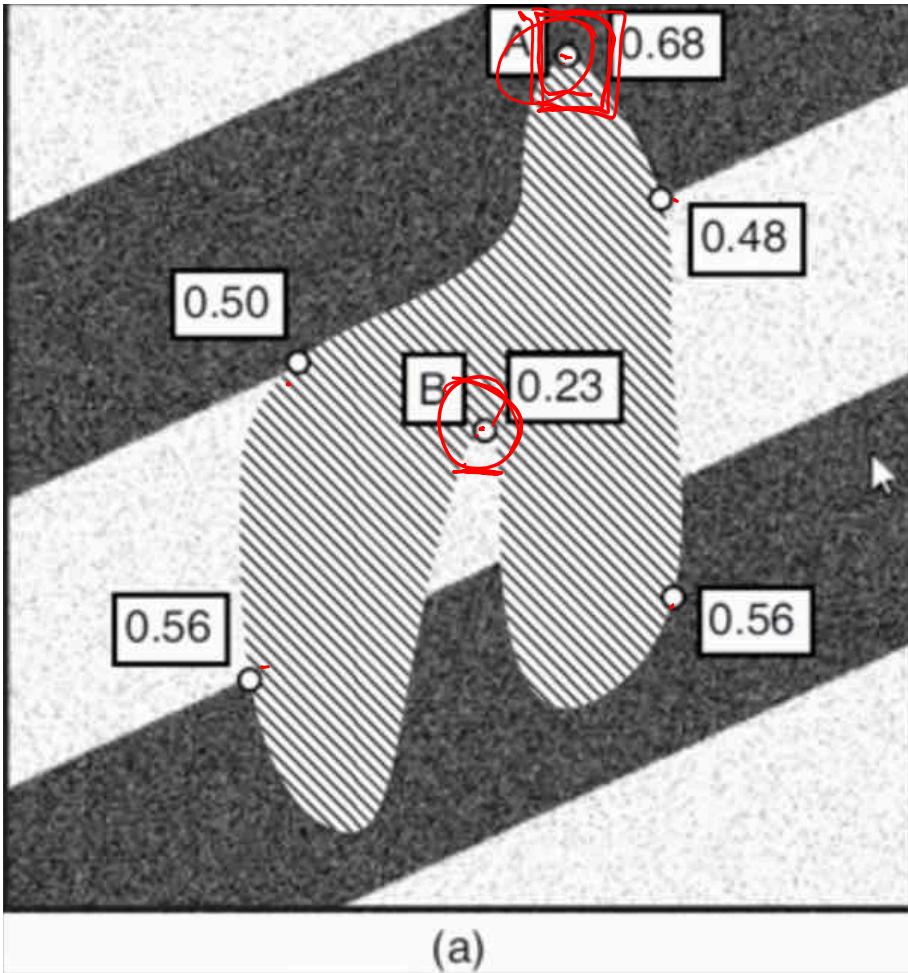
Patch-based method



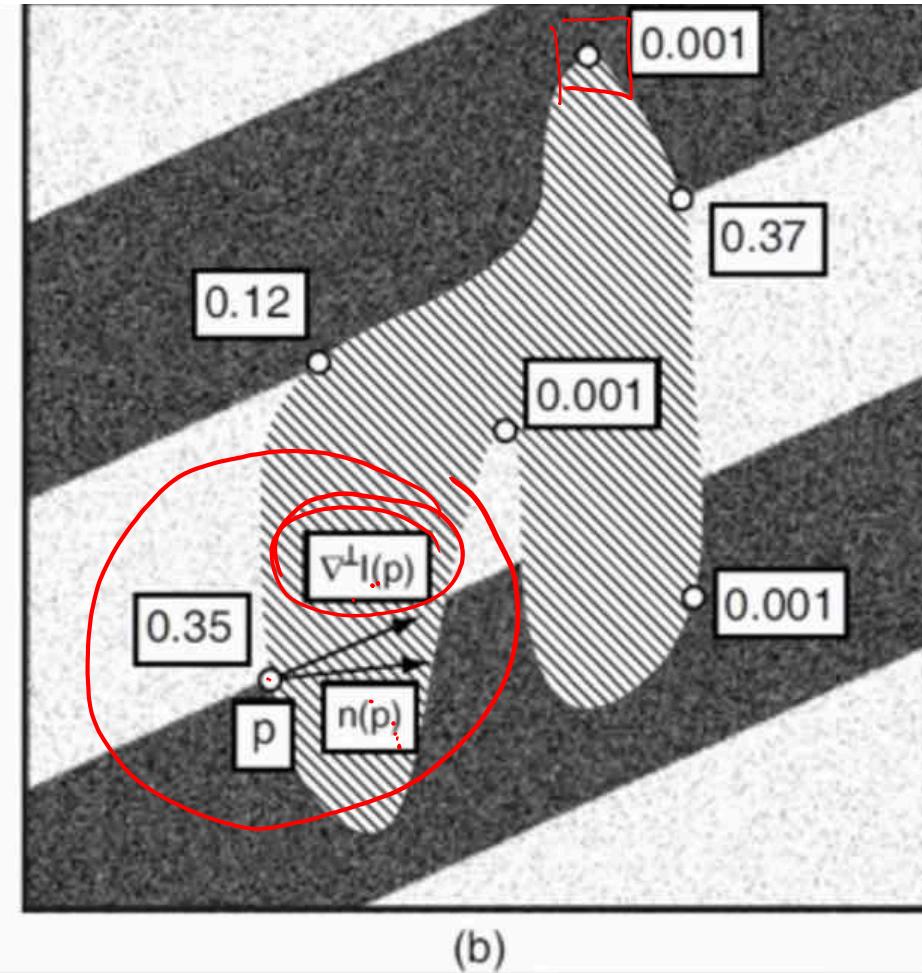
Patch-based method

- Determine a priority for each pixel on the hole boundary.
- Select patch around pixel p with highest priority, Ψ_p .
- Search remainder of image for the best matting patch Ψ_q .
- Overwrite hole pixels in Ψ_p with corresponding pixel from Ψ_q ,
shrink the hole, update priority.

Patch-based method



(a)



(b)

Patch-based method

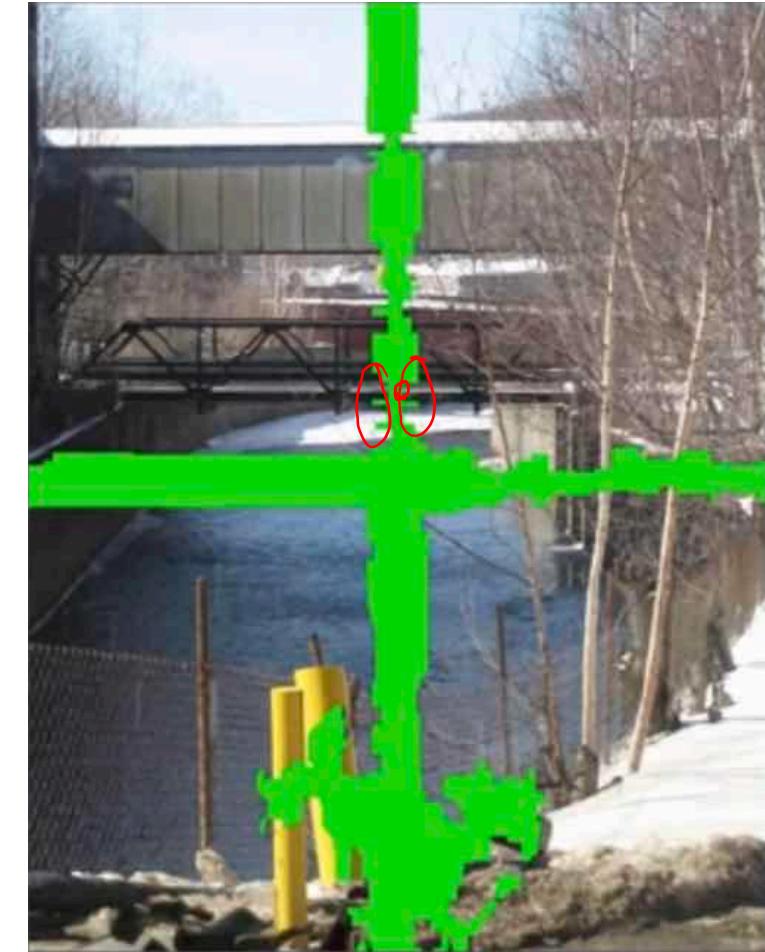
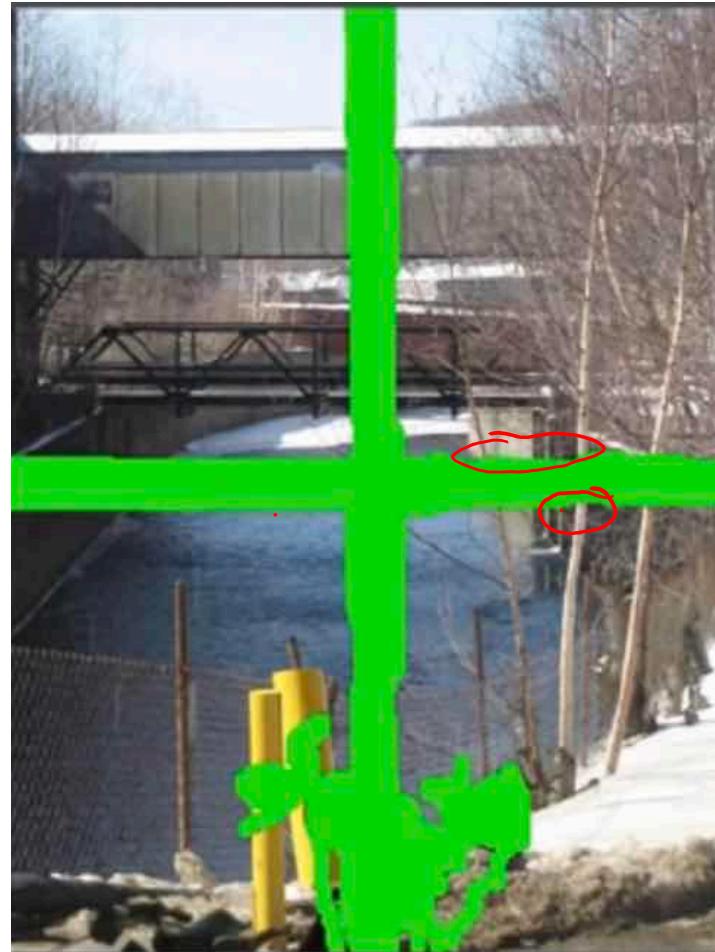
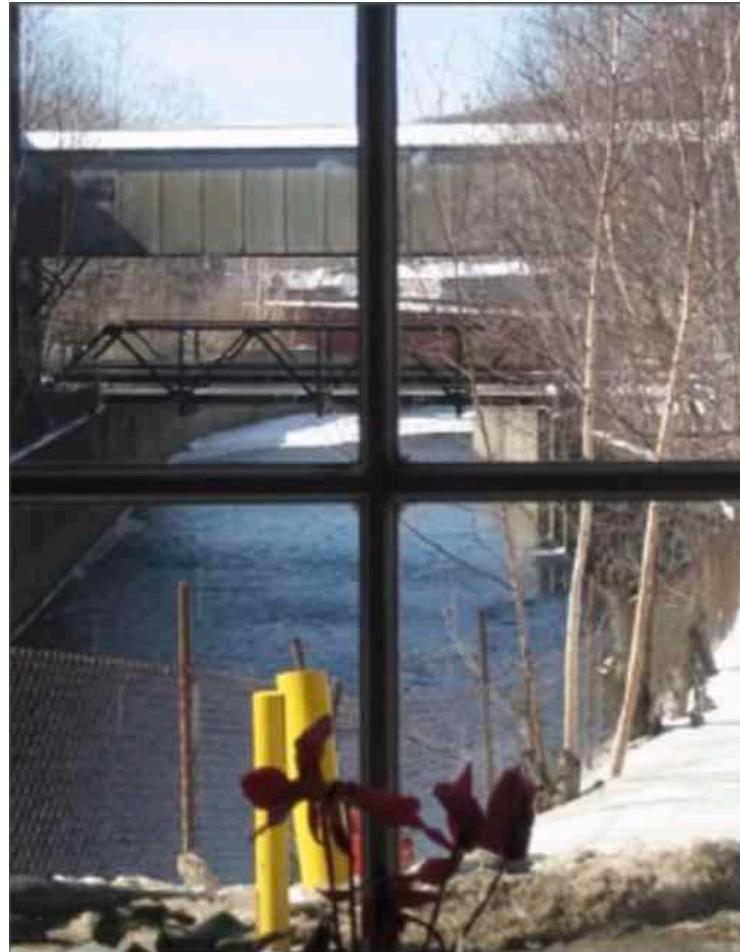
- Which pixel should have highest priority?
- 1) confidence term: high confidence if pixel is surrounded by known pixels $C(p)$.
- Date term: highest strong edges from outside the hole hit the hole at right angles.

$$D(p) = \|\nabla I(p)\| (\nabla^2 I(p) \cdot \vec{n}(p))$$

↑ grad ↑ edge direction ↓ normal to the hole

$$\varphi_p = C(p) \cdot D(p)$$

Patch-based method performance



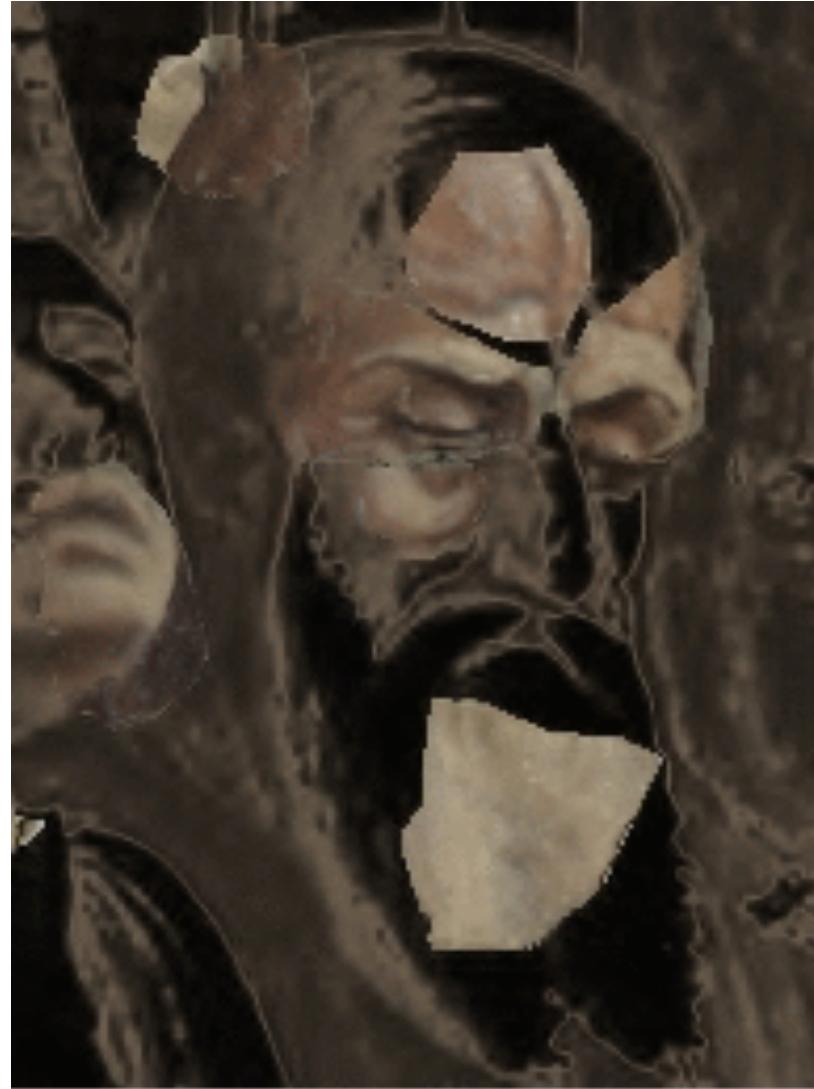
Patch-based method performance



Patch-based method performance



Patch-based method performance



Patch-based method performance

