Project 10: Image inpainting

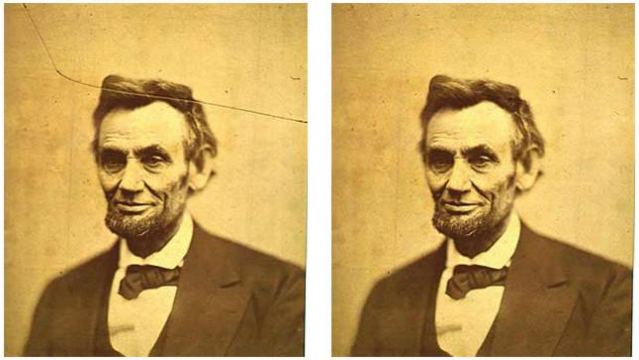
Pietro Corsi

Image inpainting

“**Inpainting** is the process of reconstructing lost or deteriorated parts of images and videos.”

Structural inpainting

- The idea is to propagate the isophotes (linear structure) that arrive at the boundary of the inpainting region, smoothly inside the region while preserving the arrival angle.
- Useful for small regions.



Textural inpainting

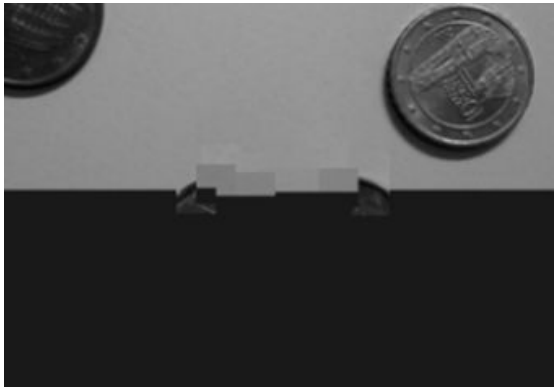
Textural inpainting

- We pick existing pixels with similar neighborhoods.
- Better with bigger regions.

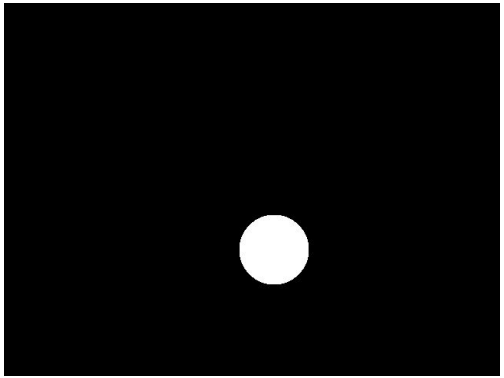
Starting with Textural inpainting

- Size window

Require the user to set it to be slightly larger than the largest distinguishable texture element.

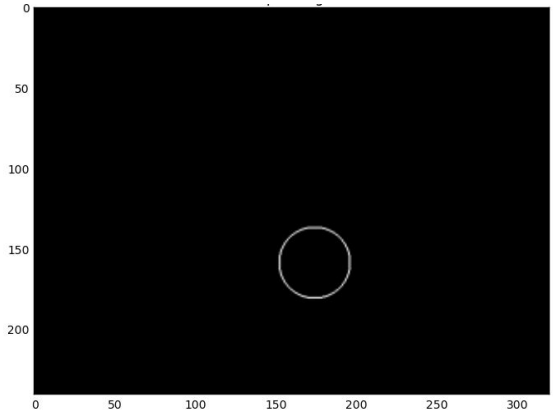


Starting with Textural inpainting



Implementation

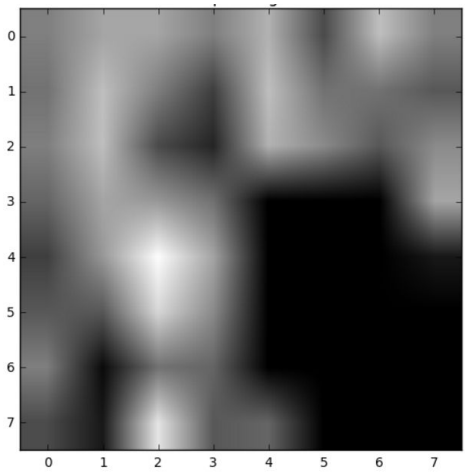
1. Find borders of the mask
2. Choose a random pixel of the border and extract its patch

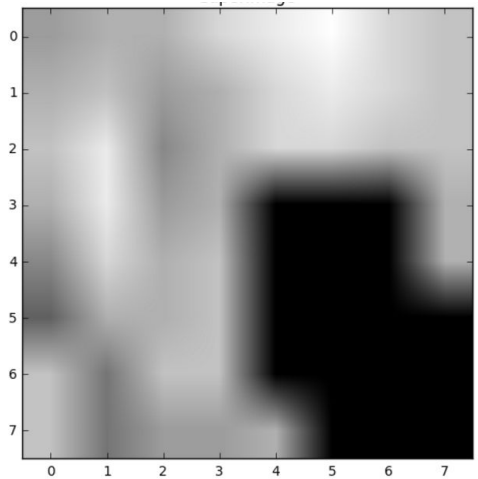


Implementation (2)

For every patch of the image:

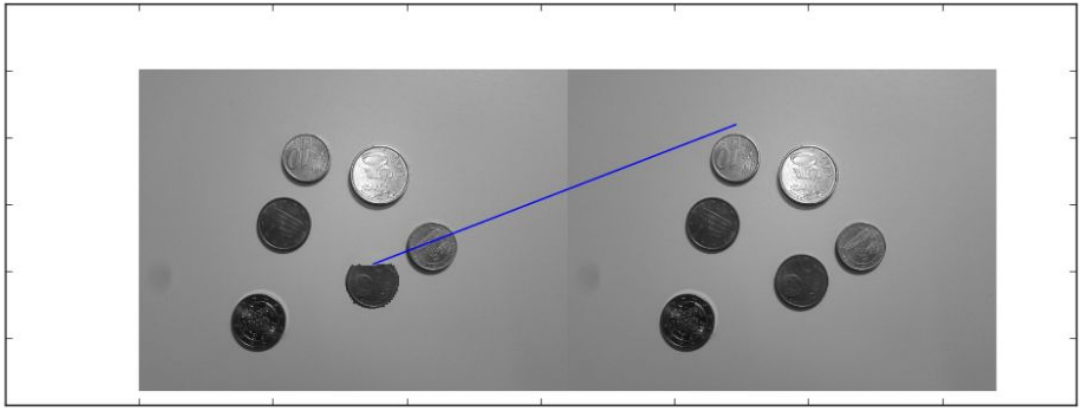
3. Multiply the patch of the mask with both the patches.
4. Distance between the patches (Euclidean distance / NCC)



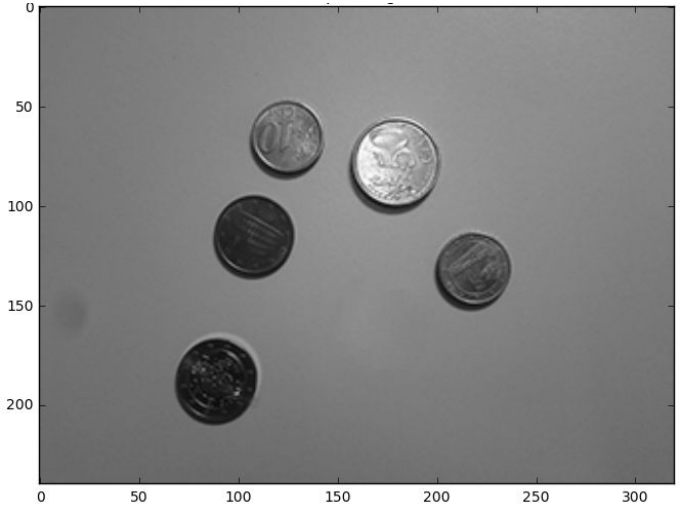


Implementation (3)

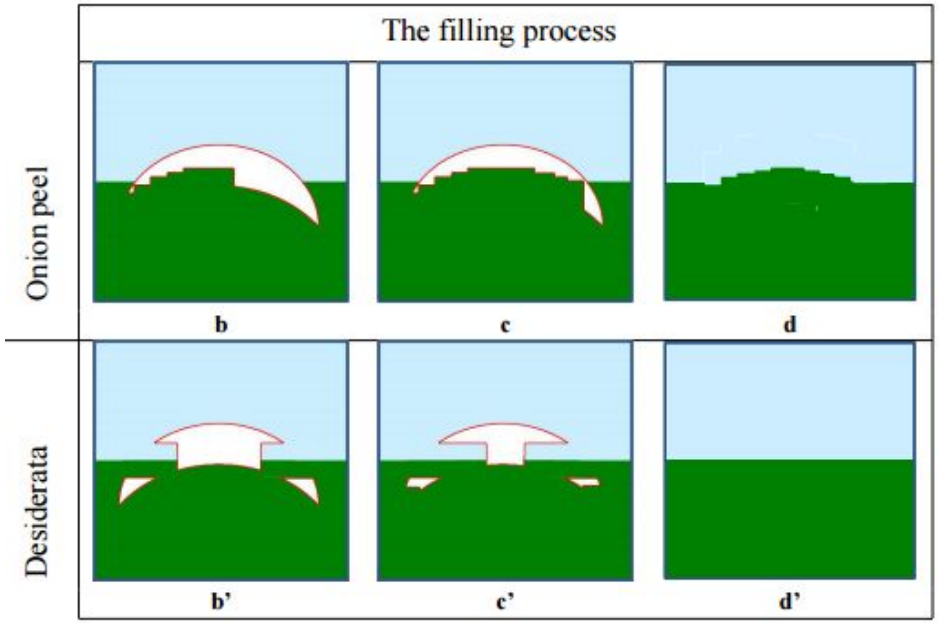
5. Find the patch closest to our pixel’s neighborhood
6. Substitute the patch
7. Update the mask
8. Back to step 1



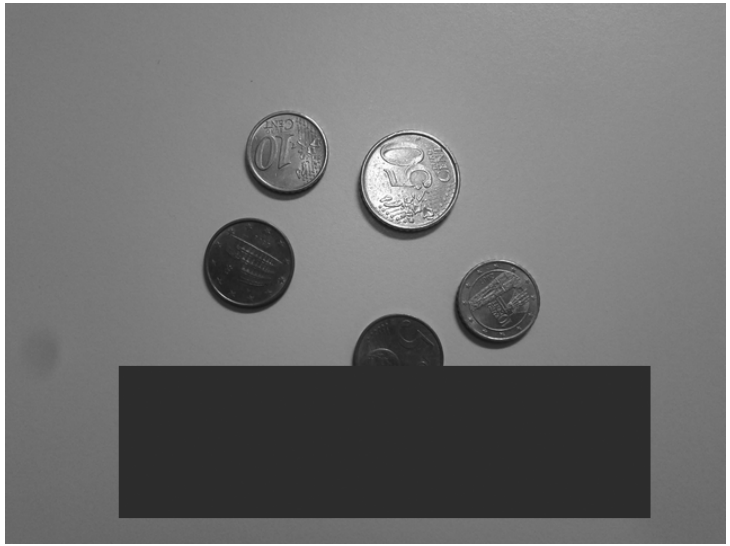
Result

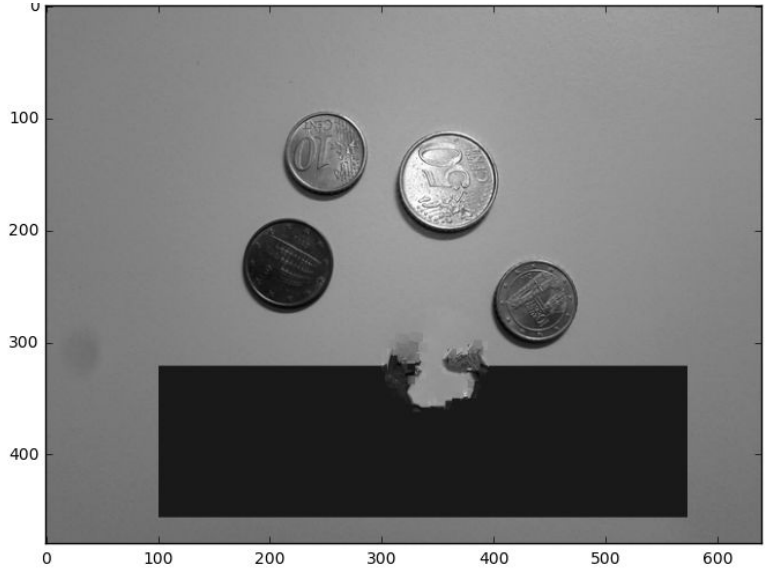


The main problem



The main problem





Criminisi Algorithm

$$P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p}). \qquad C(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\Psi_\mathbf{p}\cap(\mathcal{I}-\Omega)} C(\mathbf{q})}{|\Psi_\mathbf{p}|}, \qquad D(\mathbf{p}) = \frac{|\nabla I_\mathbf{p}^\perp \cdot \mathbf{n}_\mathbf{p}|}{\alpha}$$

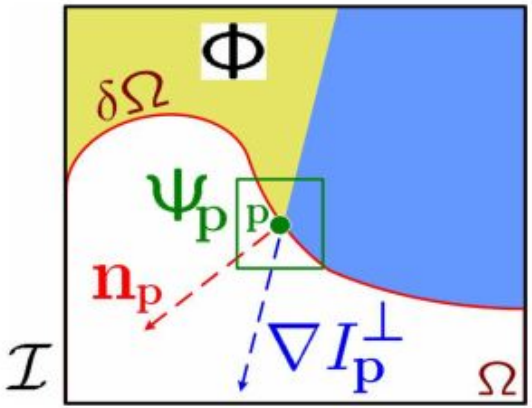
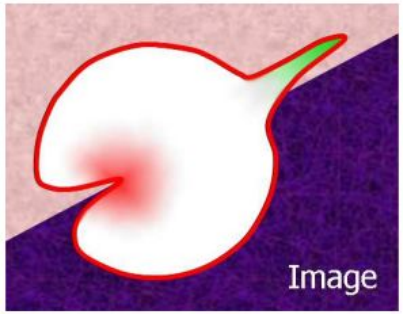
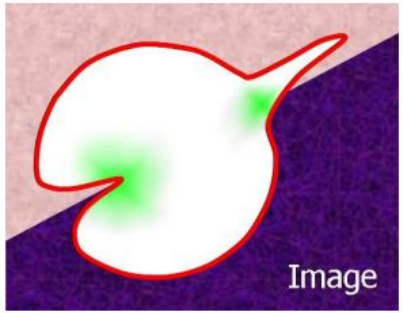


Fig. 5. **Notation diagram.** Given the patch $\Psi_\mathbf{p}$, $\mathbf{n}_\mathbf{p}$ is the normal to the contour $\delta\Omega$ of the target region Ω and $\nabla I_\mathbf{p}^\perp$ is the isophote (direction and intensity) at point $\mathbf{p}$. The entire image is denoted with $\mathcal{I}$.

Criminisi Algorithm



a



b

Red: low priority
Green:high priority

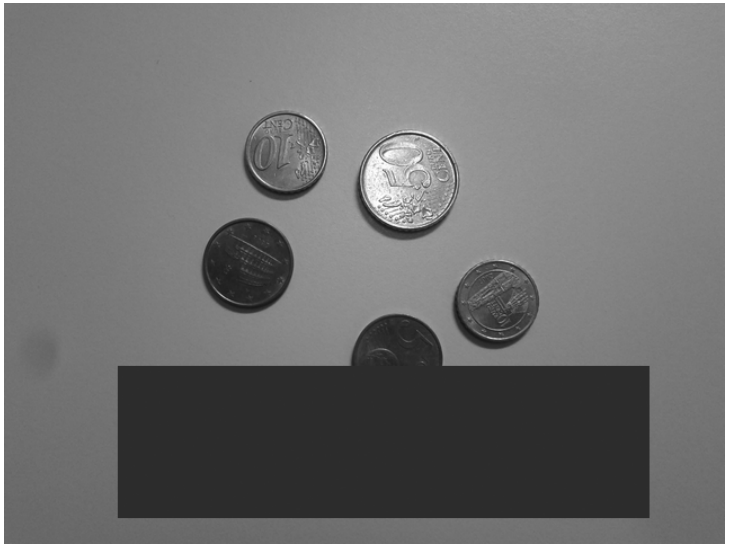
a. Confidence term

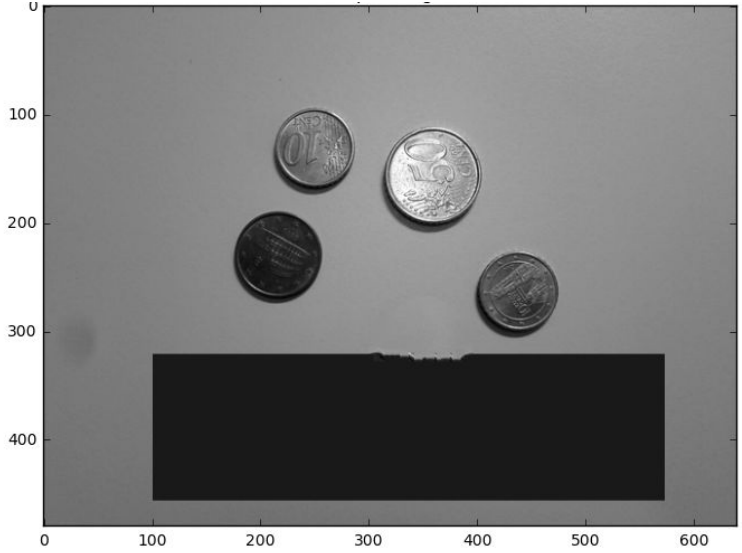
b. Data term

Criminisi Algorithm

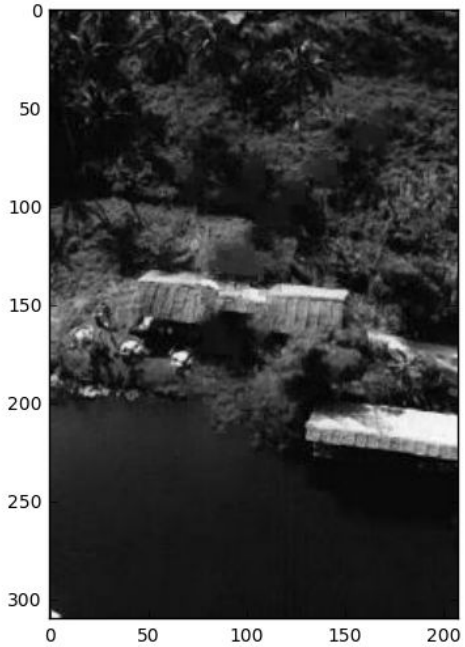
- Repeat until done:
  - 1a. Identify the fill front border B
  - 1b. Compute priorities P(p) $\forall p \in B$. P=D*C
  - 2a. Find the pixel with the maximum priority and the patch p1.
  - 2b. Find the patch p2 that minimizes the distance between p1 and p2.
  - 2c. Copy image data from p2 to p1.
  - 3. Update C(p) $\forall$ pixels of the border.

New Results

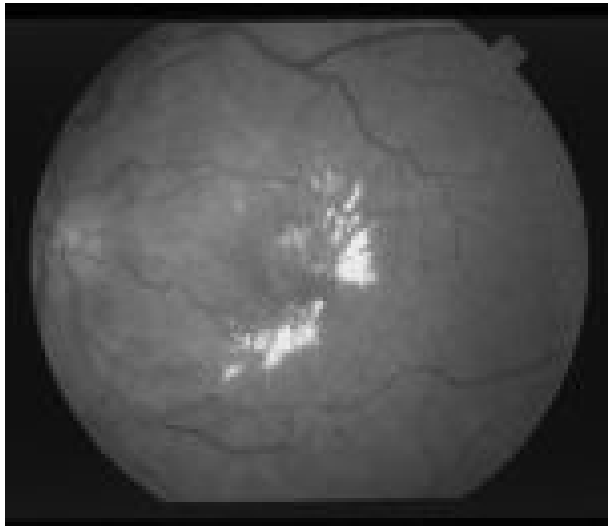


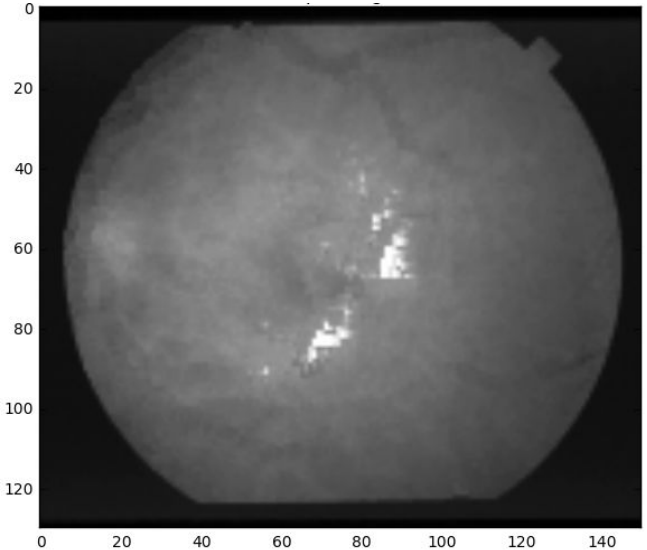


New Results



New Results





New Results