Image Inpainting

Digital signal and image processing Project

Cantarini Giorgio

a.a. 2017/2018

Summary

1. Problem
2. Algorithm
3. Results

1

Problem

What is image inpainting?

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

Goals and applications

Some examples:

- restoration of damaged paintings and photographs
- removal/replacement of selected objects
- film restoration
- removal of red-eye
- removal of the stamped date from photographs
- removal of logos in videos

2

Algorithm

Notation

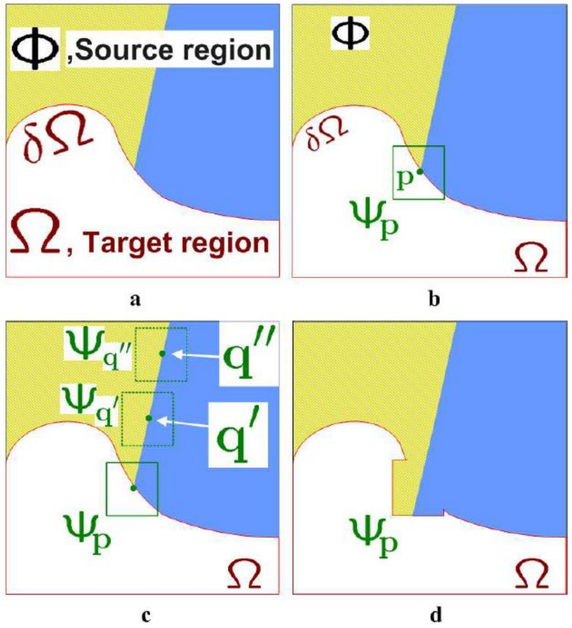


Image I

Region to be inpainted Ω

Source region $(I - \Omega) = \Phi$

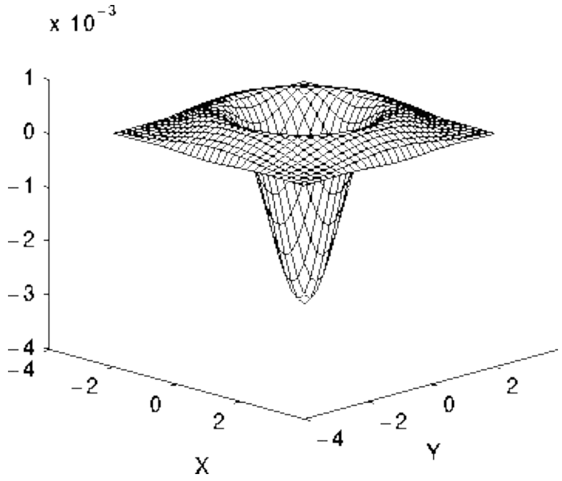
Boundary of target region $\delta\Omega$

The patch centred on the point p Ψ_p

Finding boundary of target region

Convolving the image with Laplacian filter we find edge

1	1	1
1	-8	1
1	1	1



“Region Filling and Object Removal by Exemplar-Based Image Inpainting” A.Criminisi, P. Pèrez

Given an input image, the user selects a target region to be removed and filled.

Then the user select the size of the template window.

The algorithm iterates the following **three steps** until all pixels have been filled:

1) Computing patch priorities
2) Propagating texture and structure information
3) Updating confidence values

1) Computing patch priorities

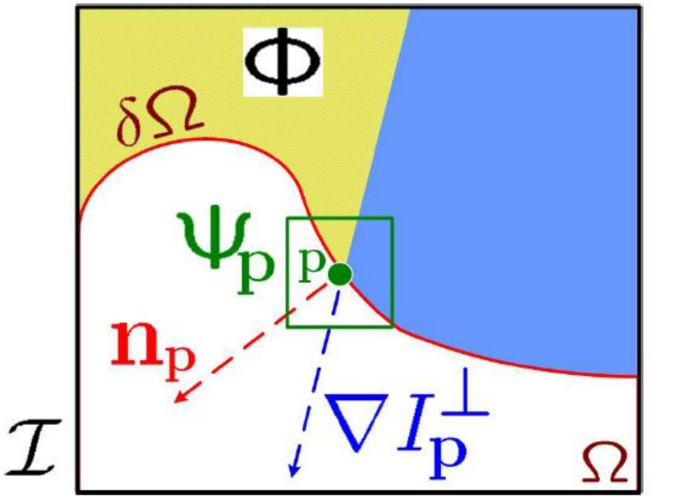
Given a patch Ψ_p centred at the point p for some $p \in \delta\Omega$, we define its priority $P(\mathbf{p})$

$P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p})$ where

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

$\mathbf{n}_{\mathbf{p}}$ is unit vector orthogonal to the front $\delta\Omega$

$\nabla I_{\mathbf{p}}^{\perp}$ is the isophote(direction and intensity)



2) Propagating texture and structure information

Once all priorities on the fill front have been computed, the patch Ψ_p with highest priority is found.

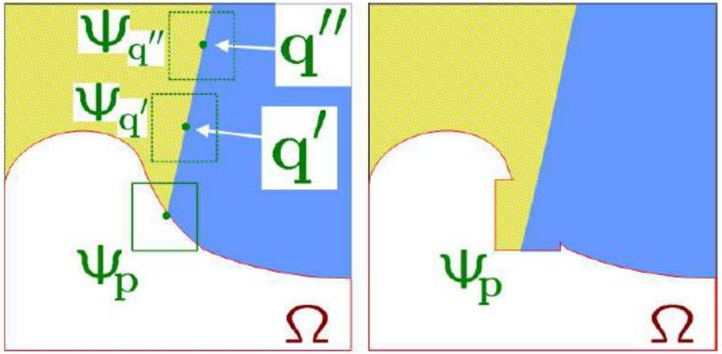
We search in the source region for that patch which is most similar to Ψ_p.

$$\Psi_{\hat{\mathbf{q}}} = \arg \min_{\Psi_{\mathbf{q}} \in \Phi} d(\Psi_{\hat{\mathbf{p}}}, \Psi_{\mathbf{q}})$$

2) Propagating texture and structure information

Once found the source exemplar $\Psi_{\hat{q}}$, the value of each pixel-to-be-filled, $\mathbf{p}' \;|\mathbf{p}' \in \Psi_{\hat{p}\cap\Omega}$ is copied from its corresponding position inside $\Psi_{\hat{q}}$.

This suffices to achieve the **propagation of both structure and texture information** from the source Φ to the target region Ω, one patch at a time.



3) Updating confidence values

After the patch $\Psi_{\hat{\mathbf{p}}}$ has been filled with new pixel values, the confidence C(**p**) is updated in the area delimited by $\Psi_{\hat{\mathbf{p}}}$ as follows:

$$C(\mathbf{p}) = C(\hat{\mathbf{p}}) \quad \forall \mathbf{p} \in \Psi_{\hat{\mathbf{p}}} \cap \Omega$$

As filling proceeds, confidence values decay, indicating that we are less sure of the colour values of pixels near the centre of the target region.

Pseudocode

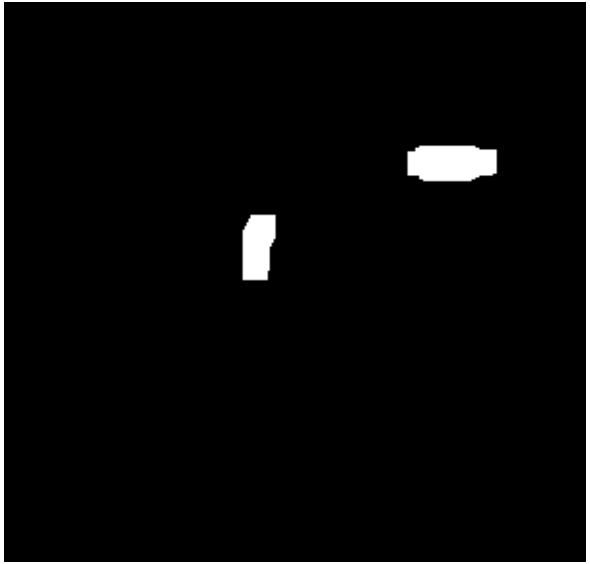
- Extract the manually selected initial front $\delta\Omega^0$.
- Repeat until done:

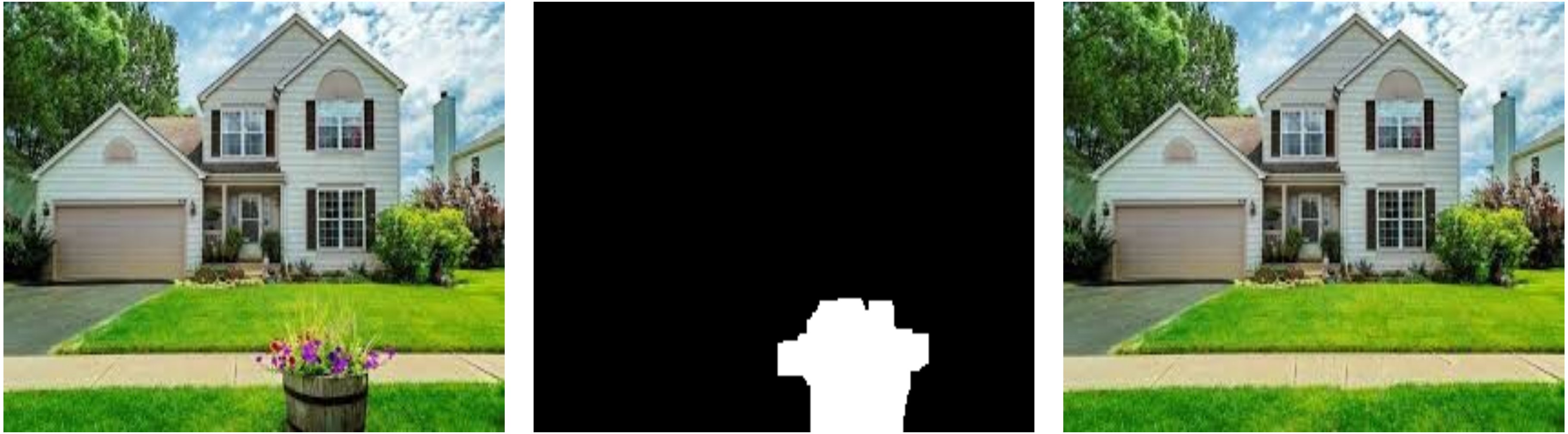
1a. Identify the fill front $\delta\Omega^t$. If $\Omega^t = \emptyset$, exit.
1b. Compute priorities $P(\mathbf{p}) \quad \forall \mathbf{p} \in \delta\Omega^t$.
2a. Find the patch $\Psi_{\hat{\mathbf{p}}}$ with the maximum priority,
i.e., $\hat{\mathbf{p}} = \arg\max_{\mathbf{p} \in \delta\Omega^t} P(\mathbf{p})$.
2b. Find the exemplar $\Psi_{\hat{\mathbf{q}}} \in \Phi$ that minimizes $d(\Psi_{\hat{\mathbf{p}}}, \Psi_{\hat{\mathbf{q}}})$.
2c. Copy image data from $\Psi_{\hat{\mathbf{q}}}$ to $\Psi_{\hat{\mathbf{p}}}$ $\forall \mathbf{p} \in \Psi_{\hat{\mathbf{p}}} \cap \Omega$.
3. Update $C(\mathbf{p}) \quad \forall \mathbf{p} \in \Psi_{\hat{\mathbf{p}}} \cap \Omega$

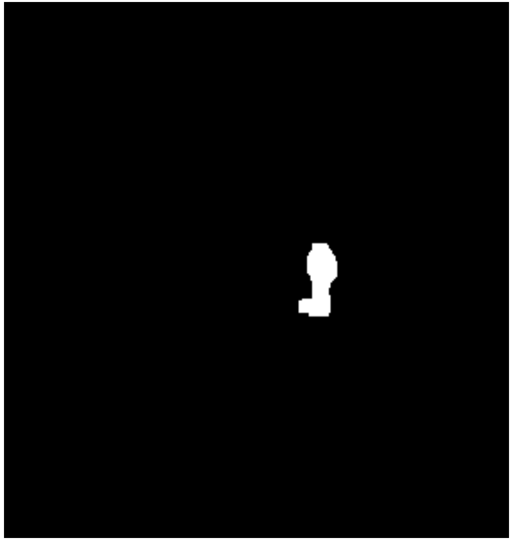
3

Results

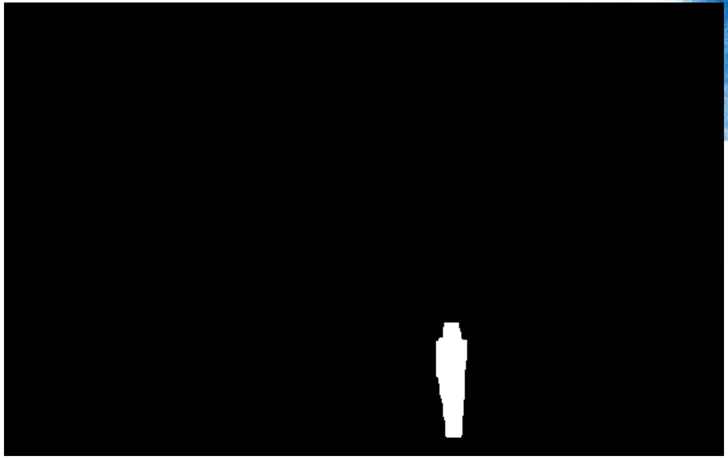




CHO59


CHO59


WILDLIFE
CONSERVATION
SOCIETY
BRONX ZOO




WILDLIFE
CONSERVATION
SOCIETY
BRONX ZOO

Thanks!

Any questions?