CSC320 A2

Image inpainting

Today’s tutorial

- Image inpainting problem
- The algorithm:
  - Patch priorities: intuition
  - Patch priorities: computation
  - Finding the best match
  - Estimating image gradients (time allowing)

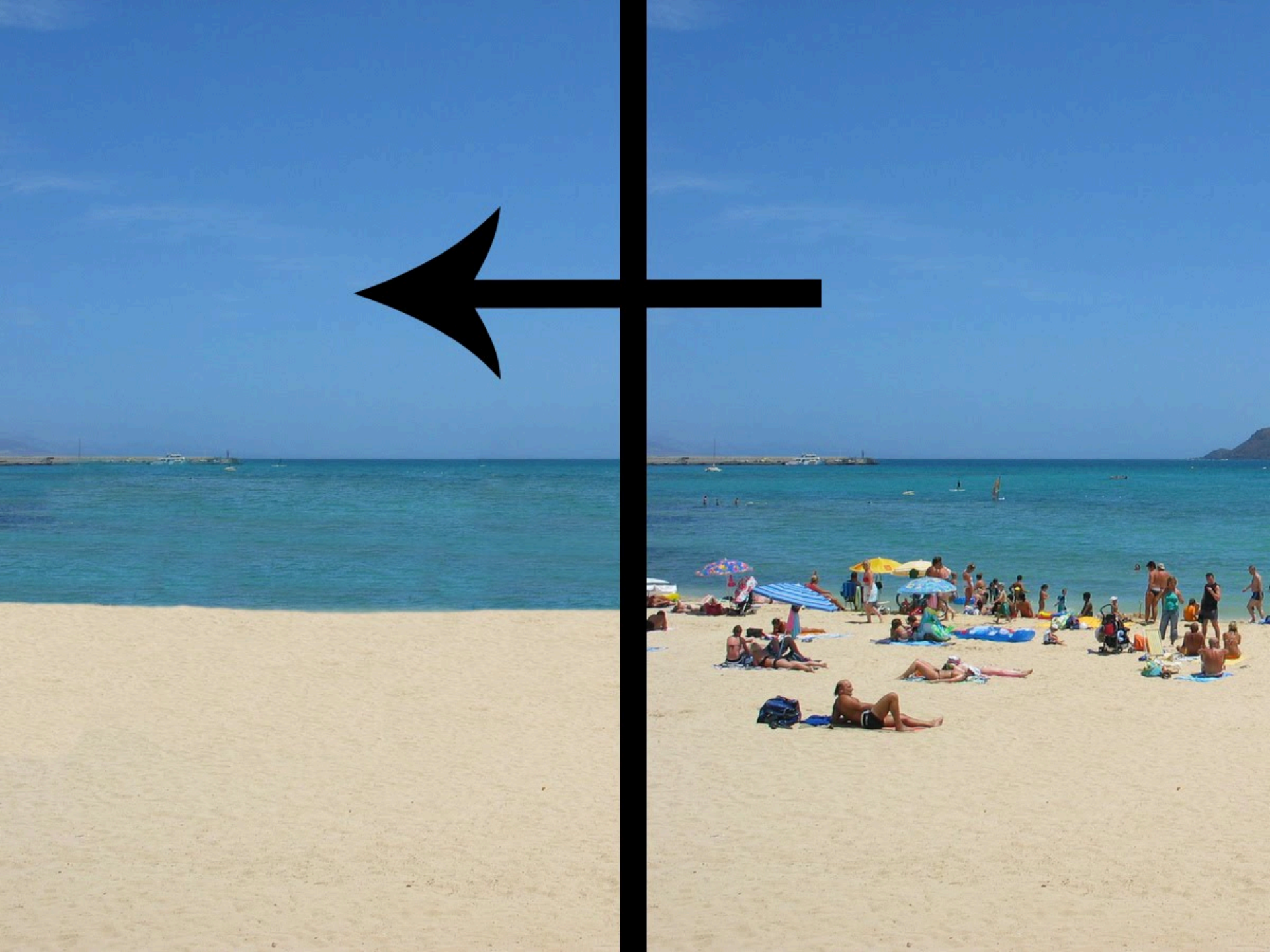
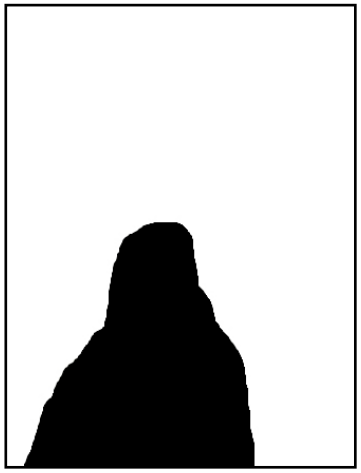


Image inpainting problem

- Inputs:
  - the original image
  - an image mask representing the area to be filled in (not necessarily a rectangle!)

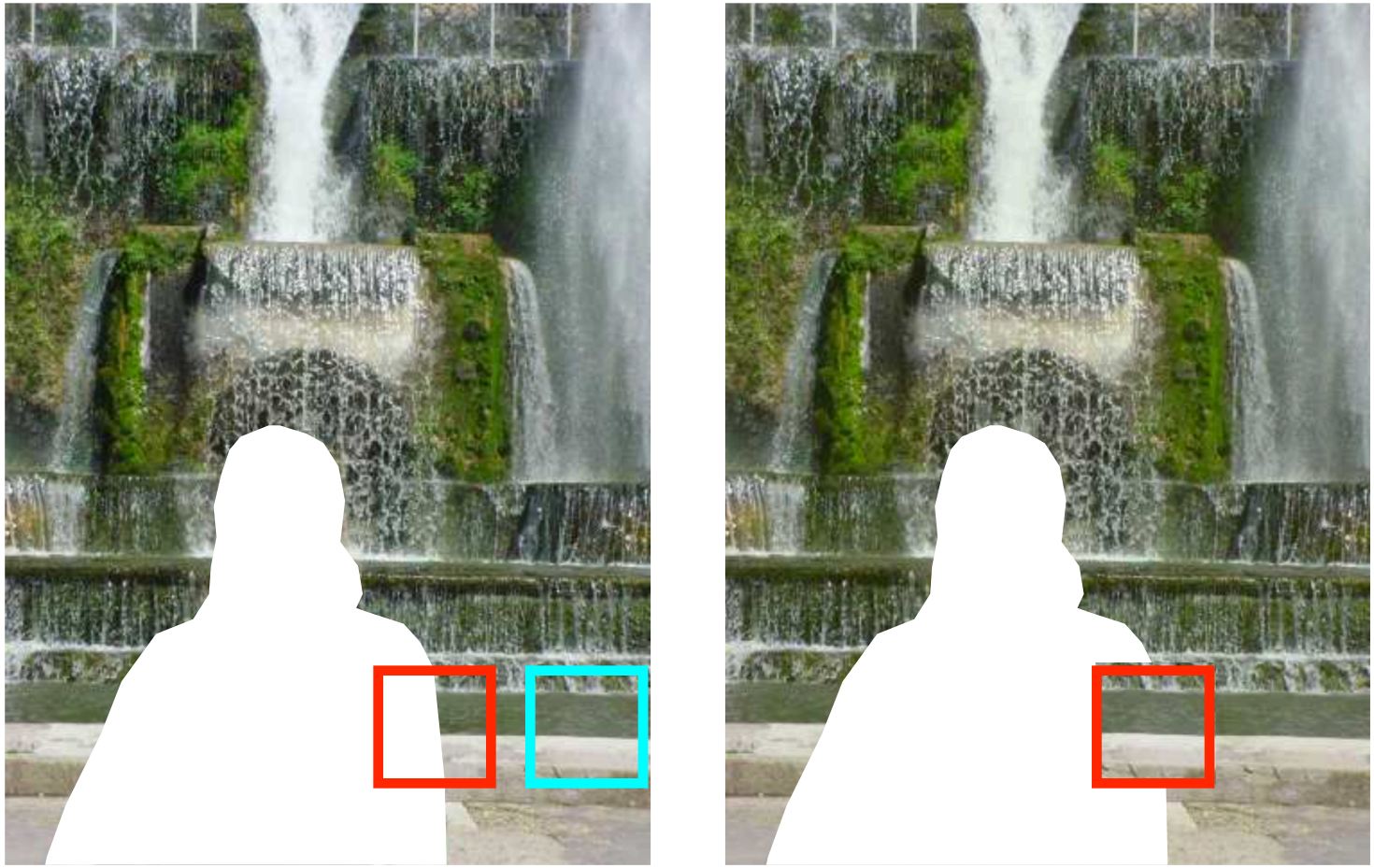


- Output:
  - a modified image with the area filled in

Image inpainting algorithm

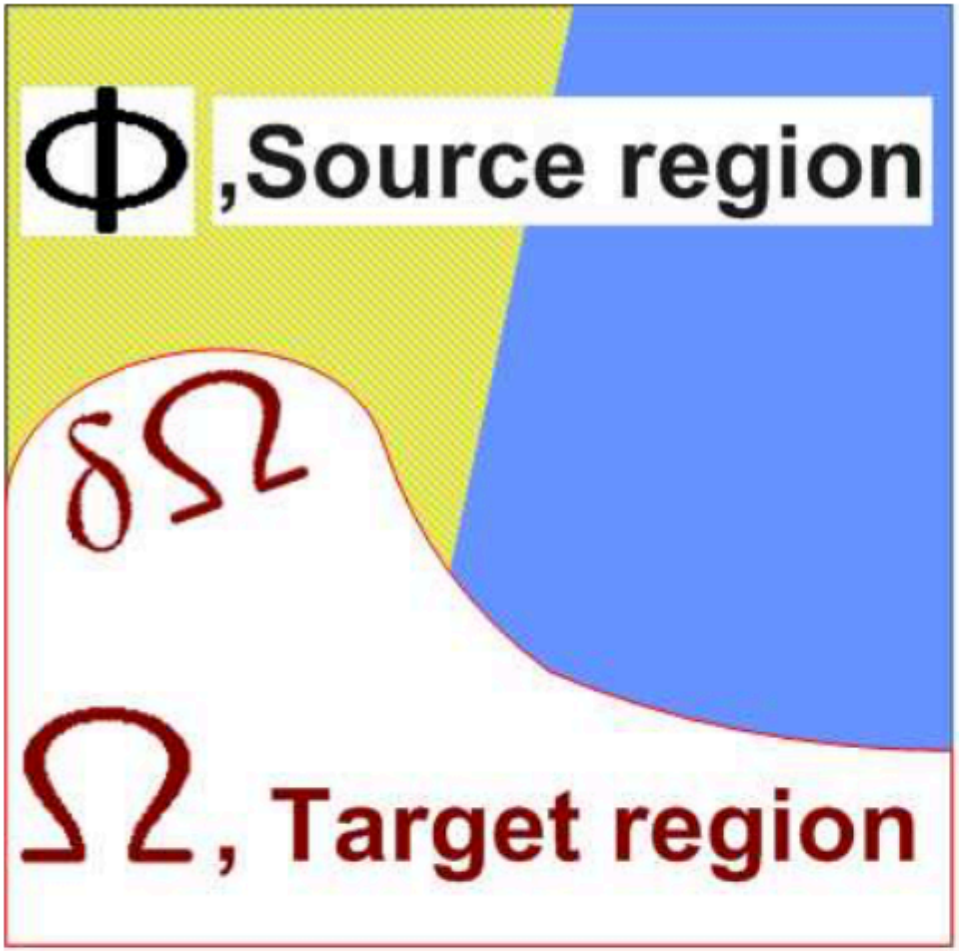
```
Split the masked region into smaller patches.
While the image is not filled:
  Pick a patch.
  Find its closest match in the image.
  Fill the patch with its match.
```

Power of patches



Exemplar-based synthesis

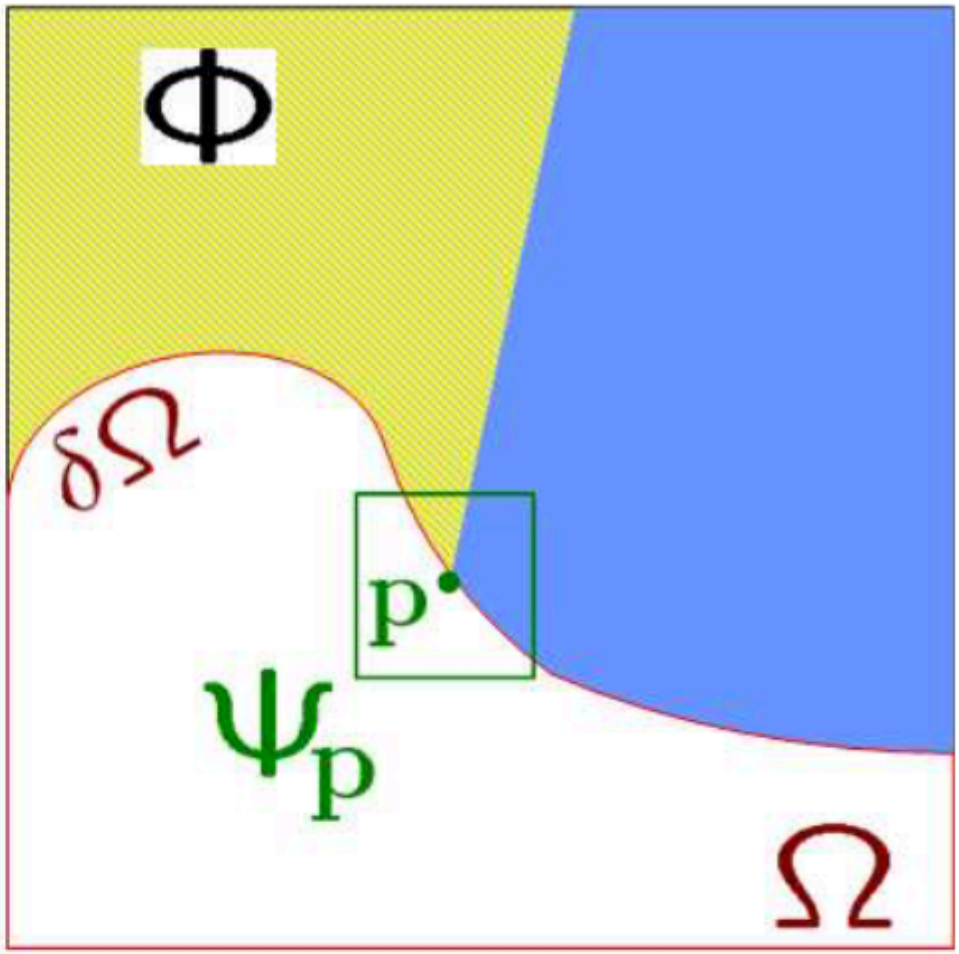
- target = Ω
- contour = $\delta\Omega$
- source = Φ
- square template Ψ_p



a

Exemplar-based synthesis

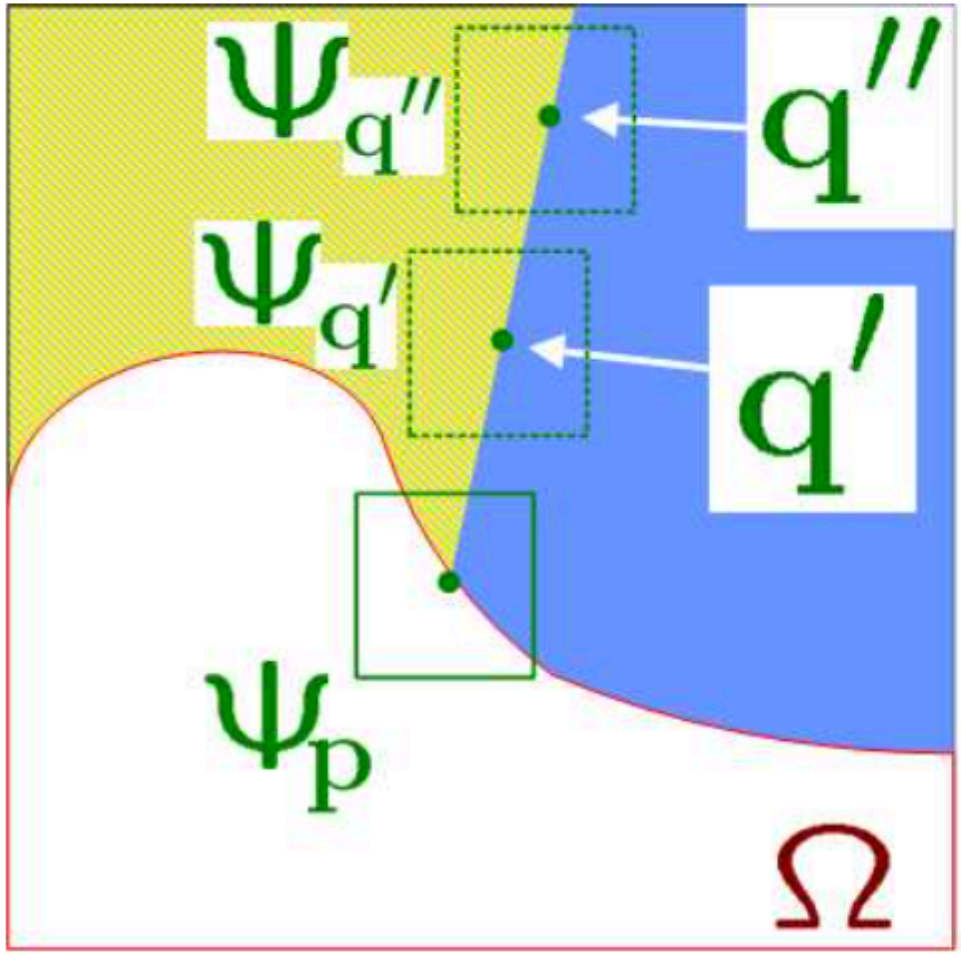
- target = Ω
- contour = δΩ
- source = Φ
- square template Ψ_p



b

Exemplar-based synthesis

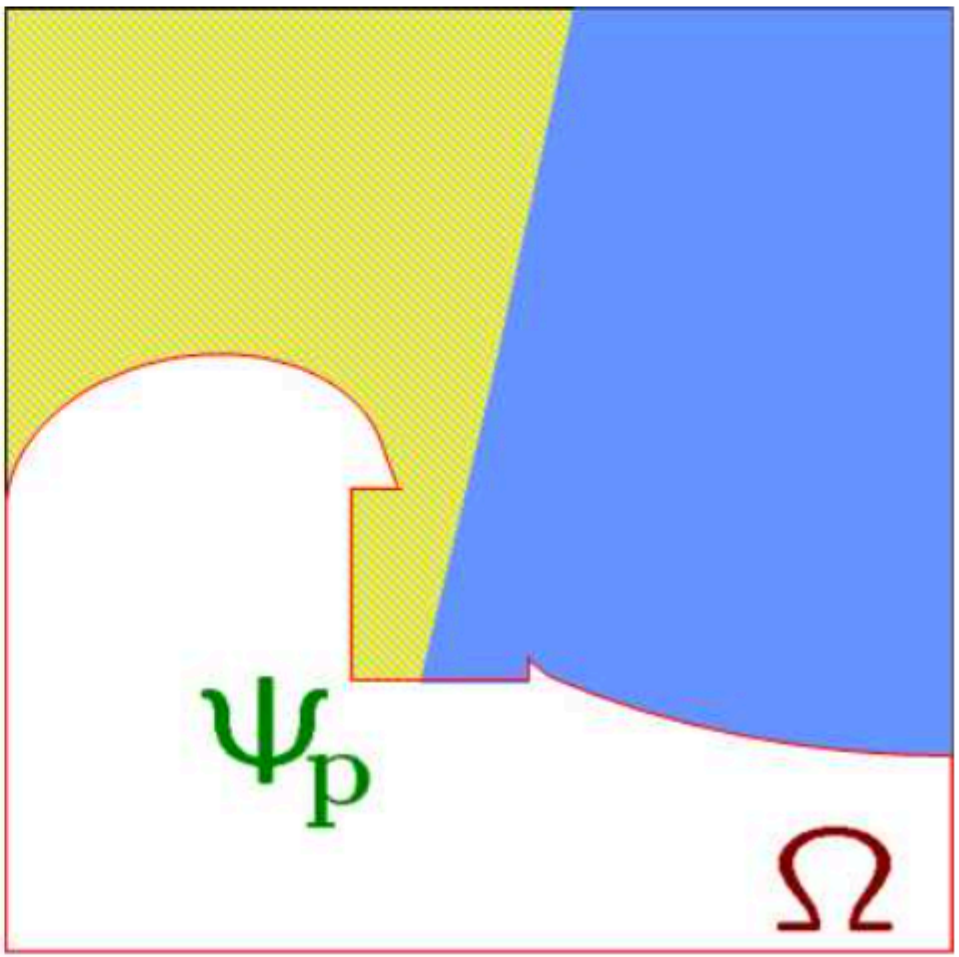
- target = Ω
- contour = $\delta\Omega$
- source = Φ
- square template Ψ_p a
- best matching Ψ_q in (excluding unfilled pixels)



c

Exemplar-based synthesis

- target = Ω
- contour = $\delta\Omega$
- source = Φ
- square template Ψ_p
- best matching Ψ_q in (excluding unfilled pixels)



d

Image Inpainting Algorithm

```
Split the masked region into smaller patches.
While the image is not filled:
  Pick a patch.
  Find its closest match in the image.
  Fill the patch with its match.
```

Image Inpainting Algorithm

```
Split the masked region into smaller patches.
While the image is not filled:
  Pick a patch.
  Find its closest match in the image.
  Fill the patch with its match.
```

Does it matter which patch you pick first?
Hint: yes.

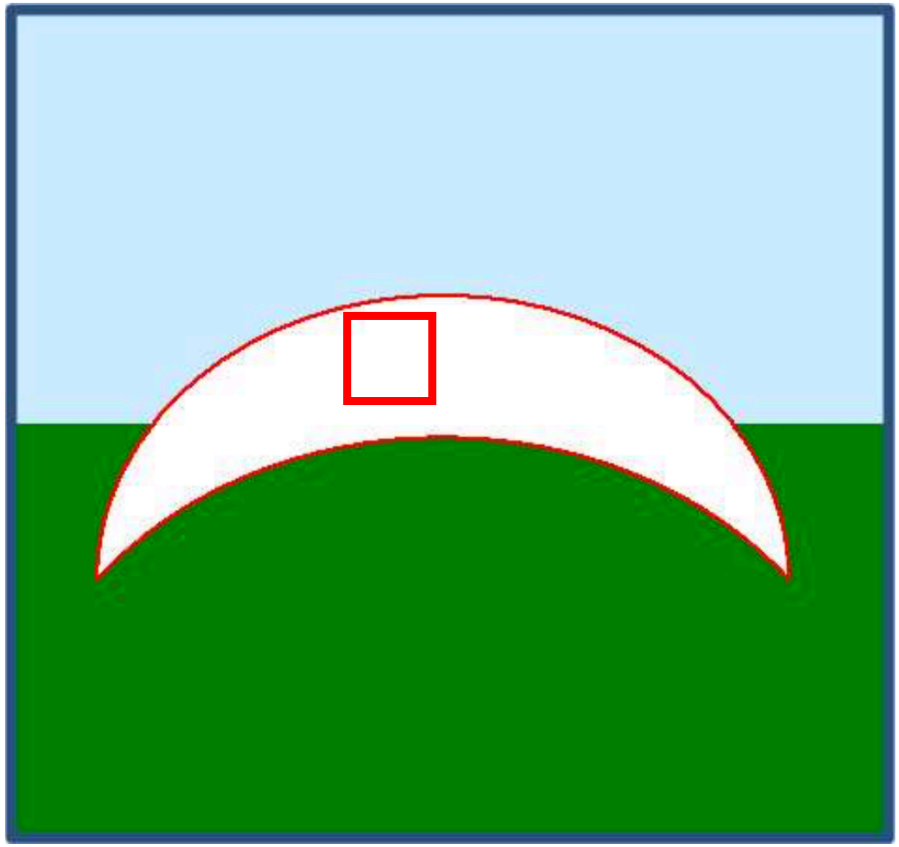
Patch priorities

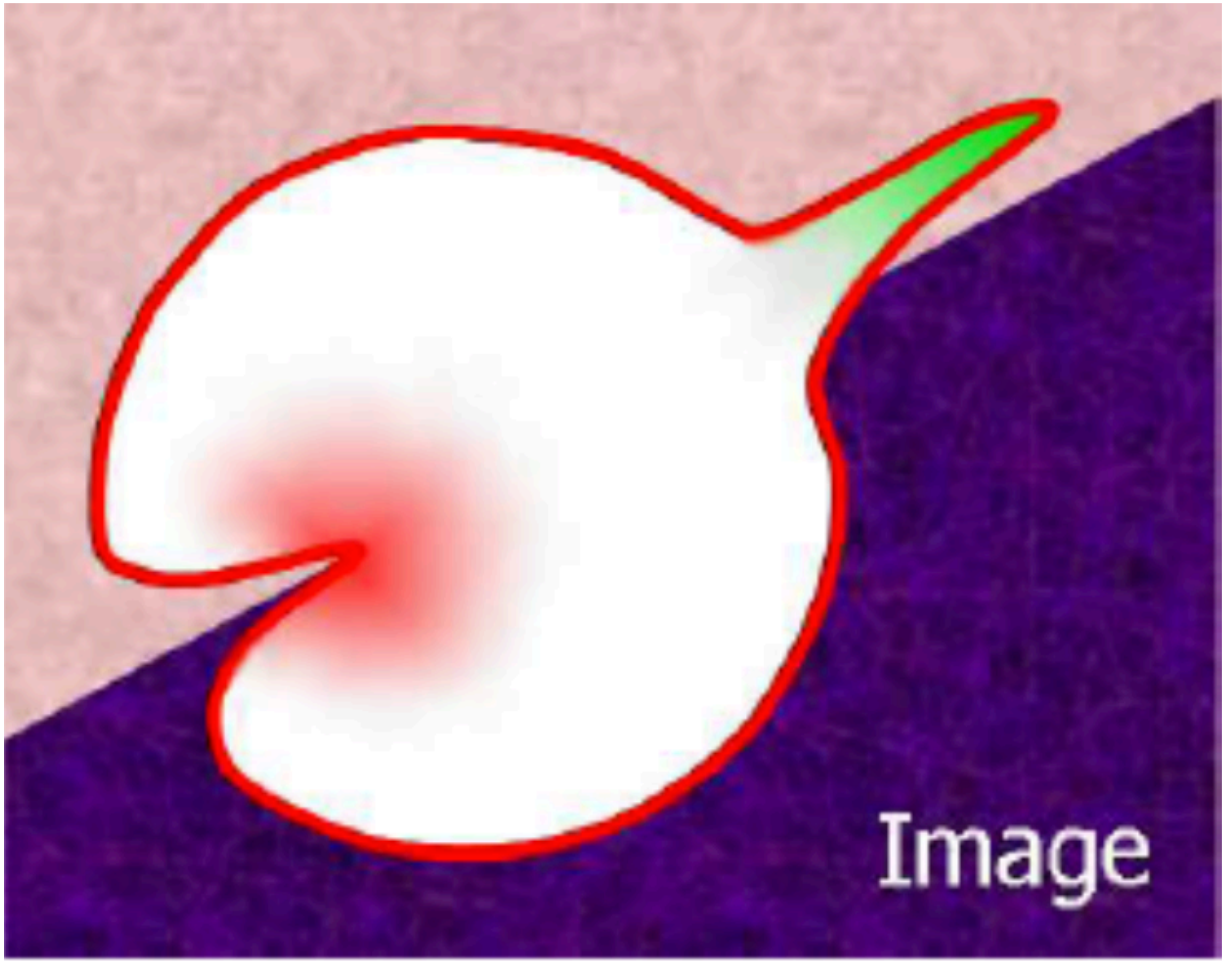
Intuition

Order Matters

Case 1: the patch is *completely unfilled*

Fix: Prioritize patches that overlap with filled regions

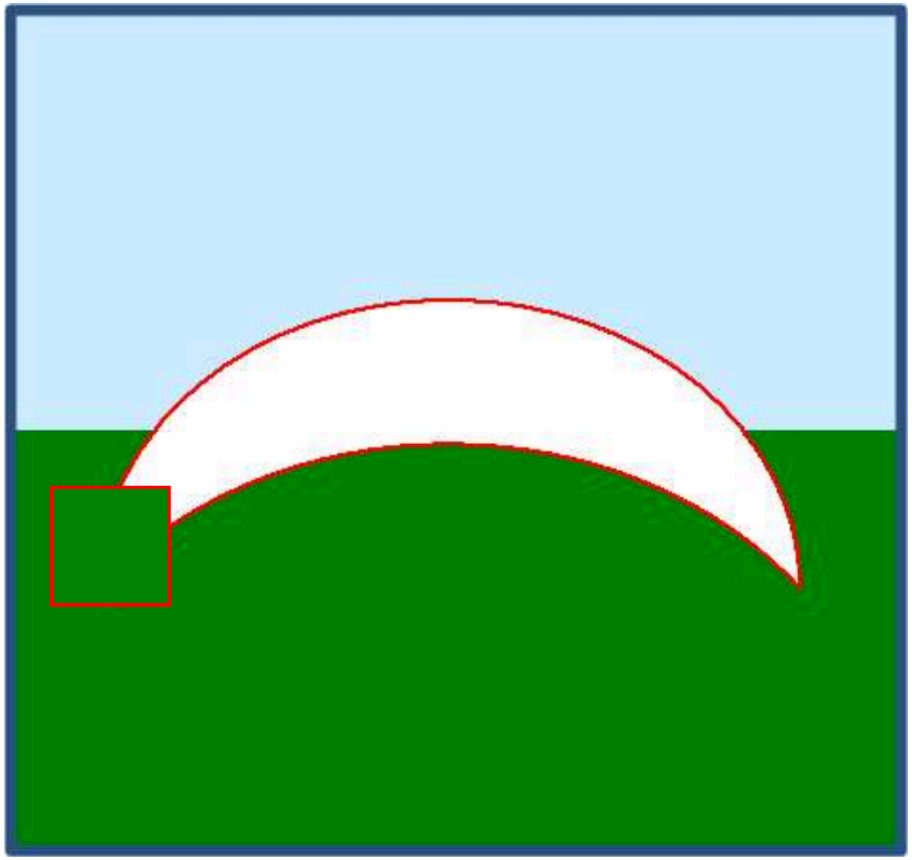




priority contributed by **confidence term**

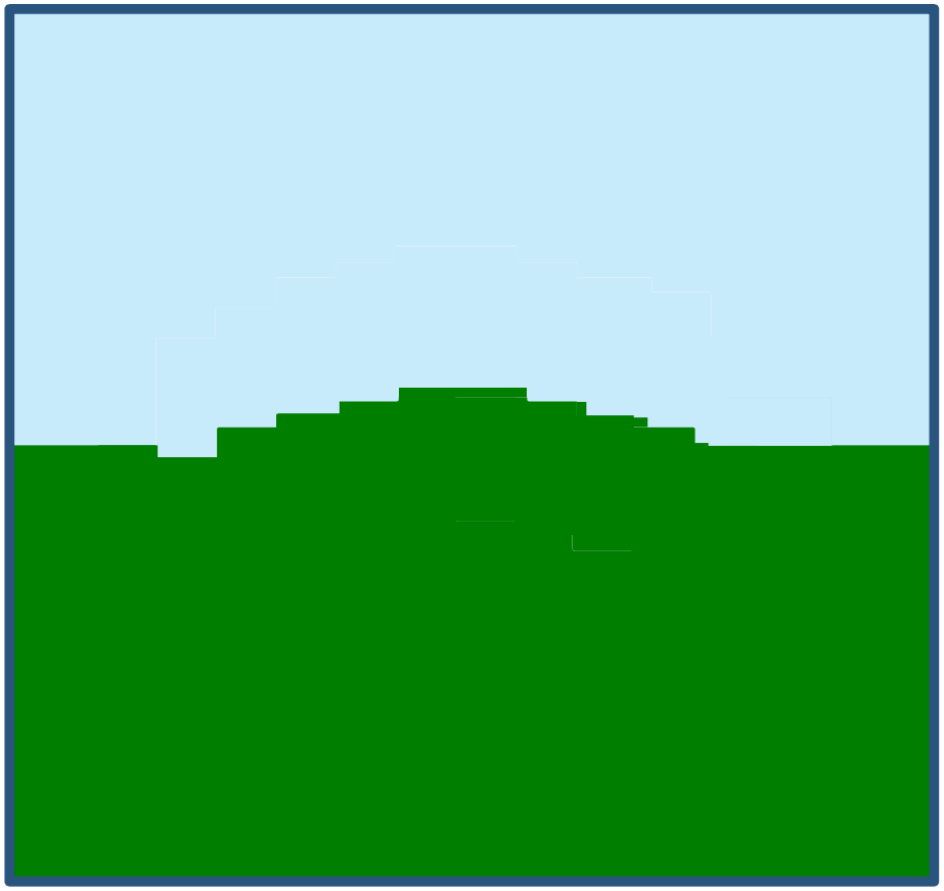
Order Matters

Case 2: the filled part of the patch is *constant / near-constant*



Order Matters

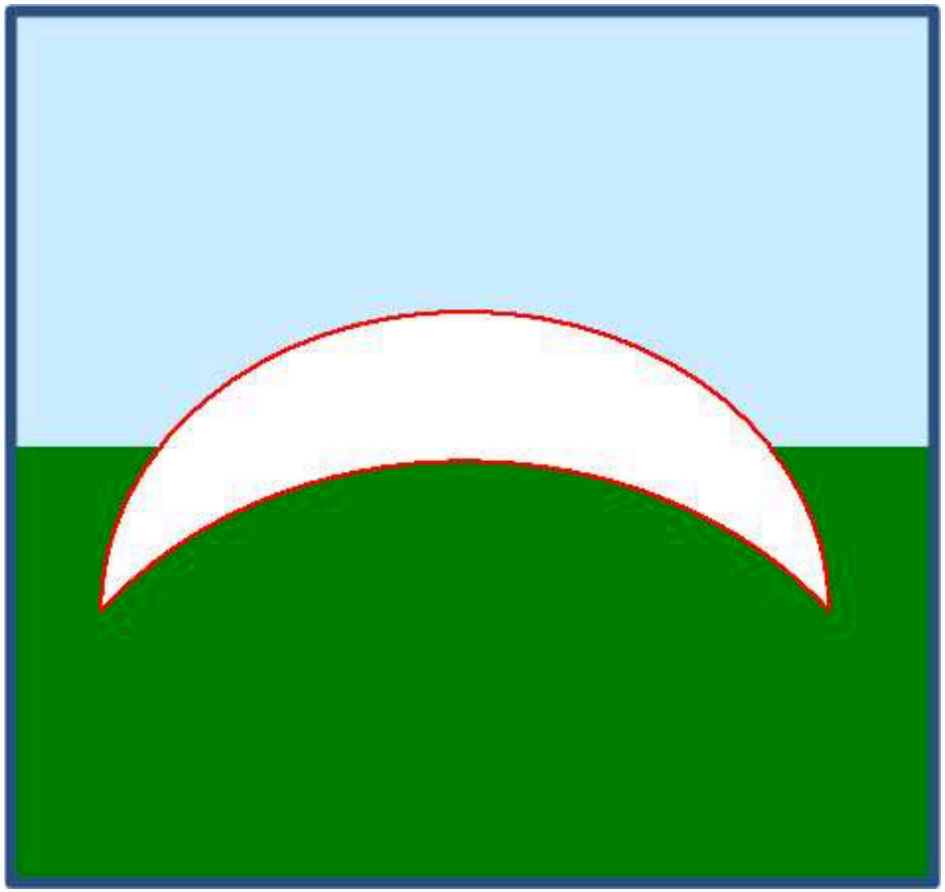
Case 2: the filled part of the patch is *constant / near-constant*



Order Matters

Case 2: the filled part of the patch is *constant / near-constant*

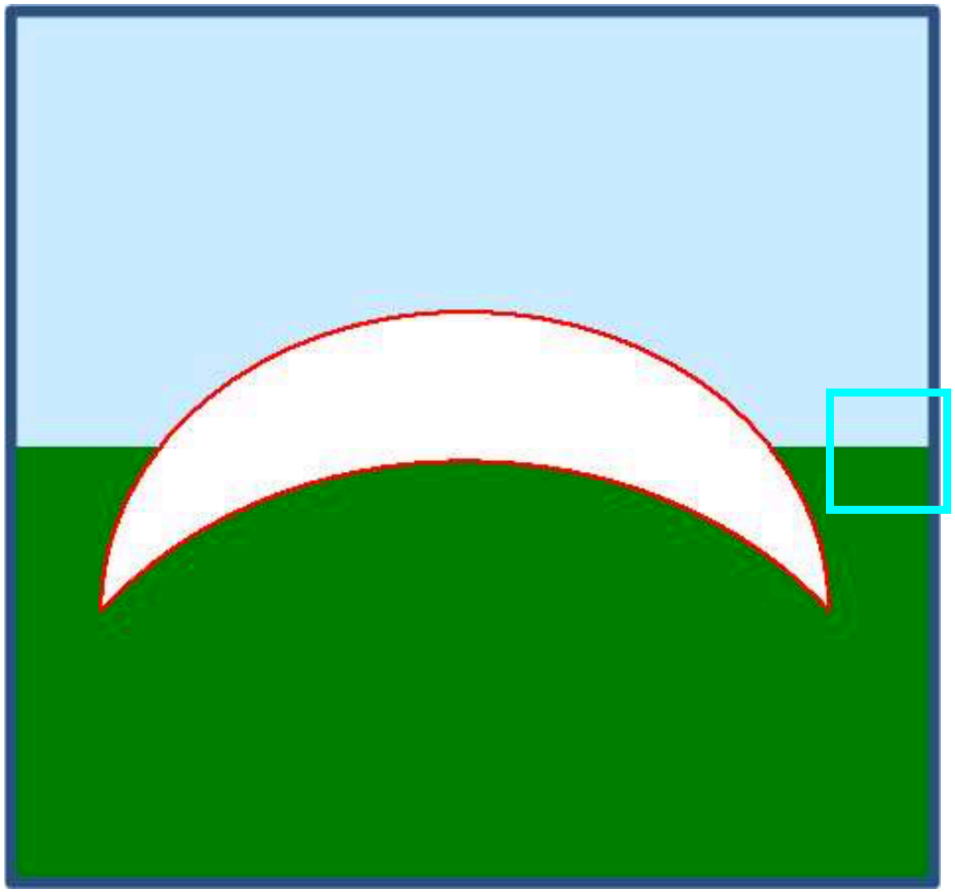
Fix: prioritize patches that *propagate structure*



Order Matters

Case 2: the filled part of the patch is *constant / near-constant*

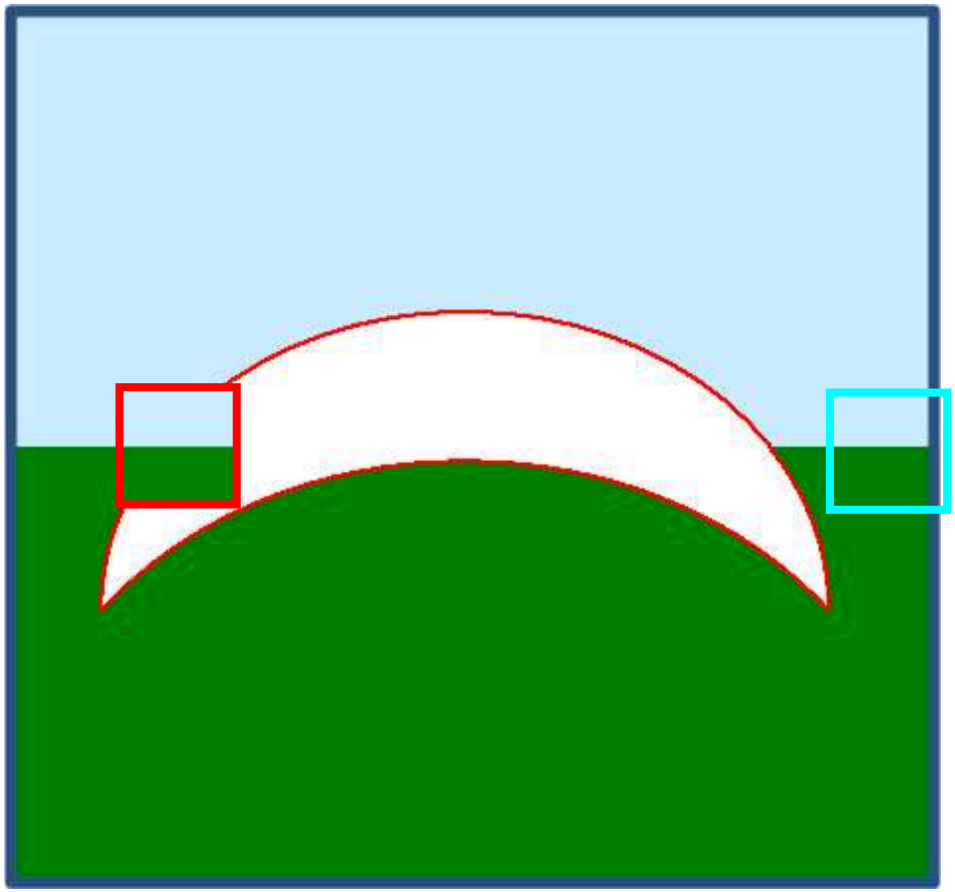
Fix: prioritize patches that *propagate structure*

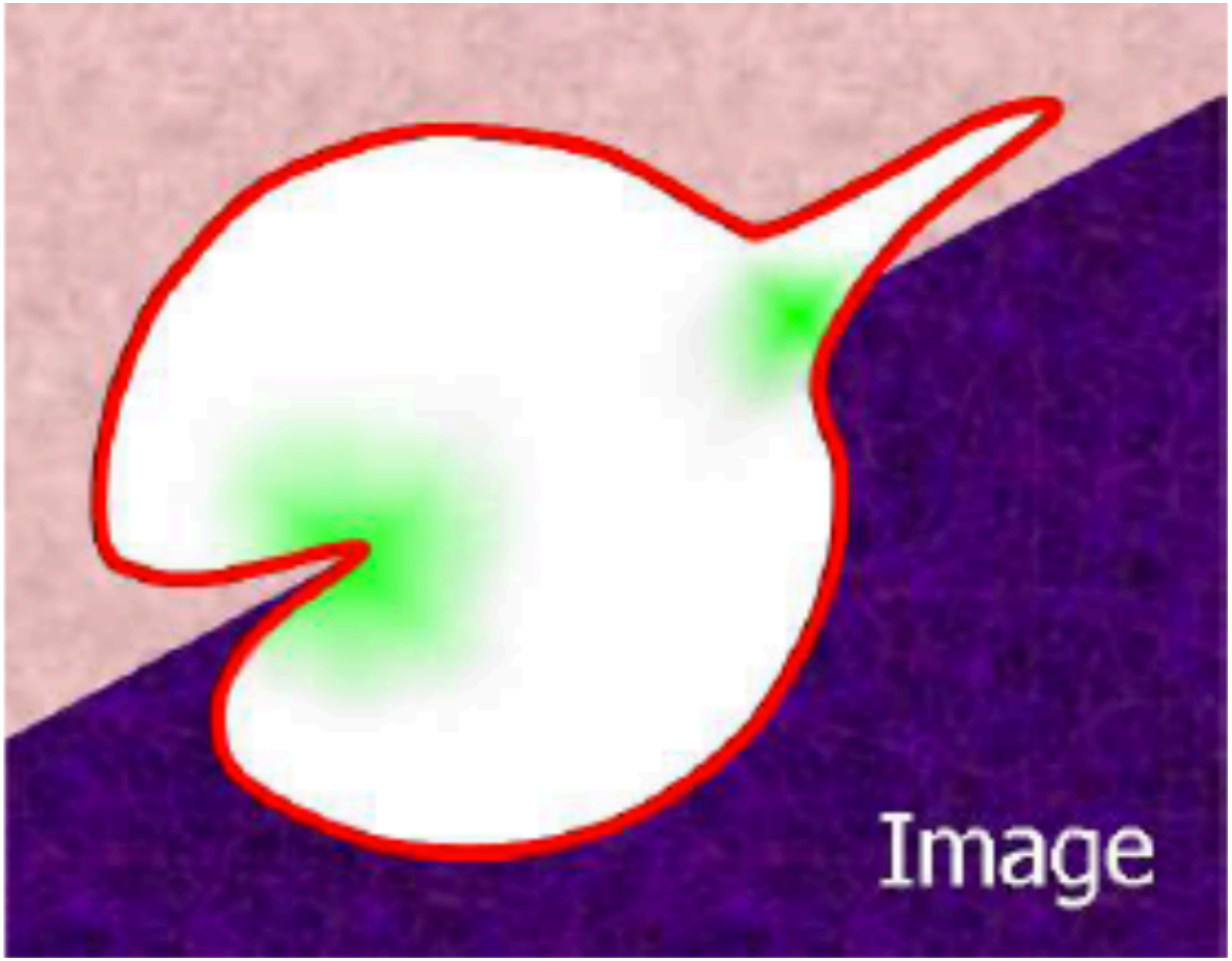


Order Matters

Case 2: the filled part of the patch is *constant / near-constant*

Fix: prioritize patches that *propagate structure*





priority contributed by **data term**

Updated algorithm

```
While the image is not filled:
  Calculate the priority for each pixel/patch
  that lies on the boundary of the mask.
  Get the patch with the highest priority.
  Find its closest match in the image.
  Fill the (unknown part of the) patch with its
  match.
  Update the boundary of the mask.
```

Patch priorities

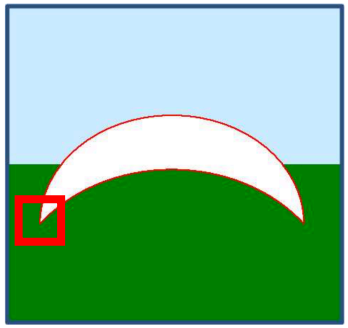
Computation

Calculating priorities: the gist of it

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

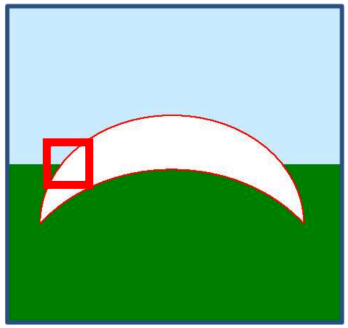
Confidence

is high if we know a lot about the patch centred around pixel *p*.



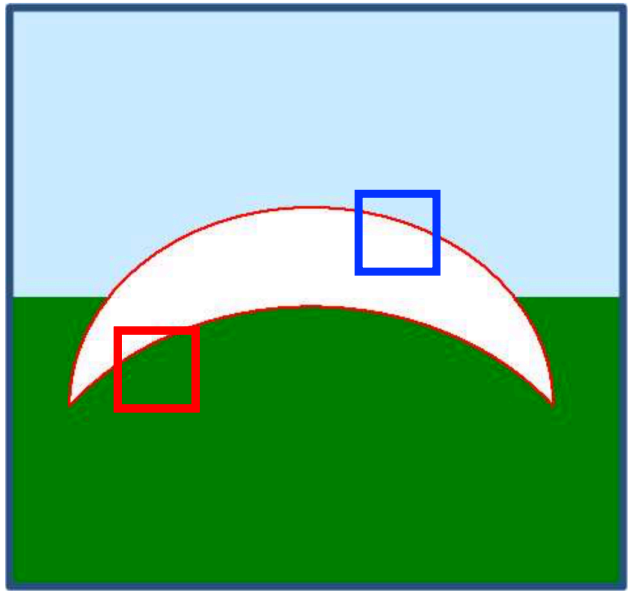
Data

is high if the patch centred around pixel *p* propagates lines/edges.



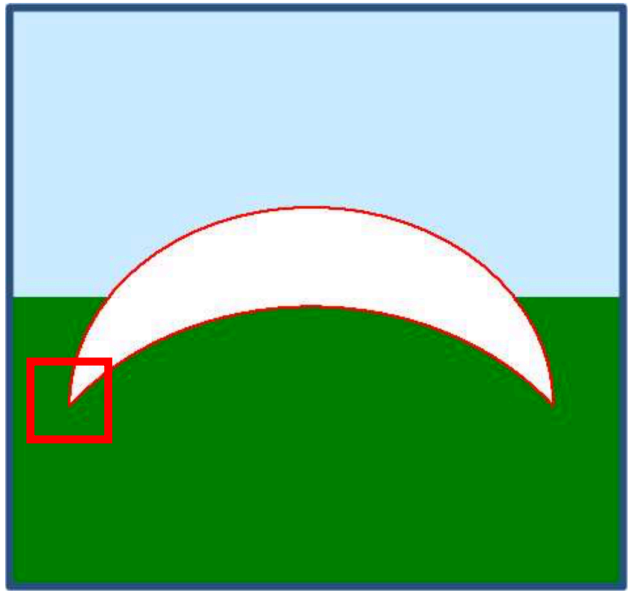
Calculating confidence term: initialization

$$C(p) = \begin{cases} 1 & \text{if } p \text{ is filled} \\ 0 & \text{otherwise} \end{cases}$$



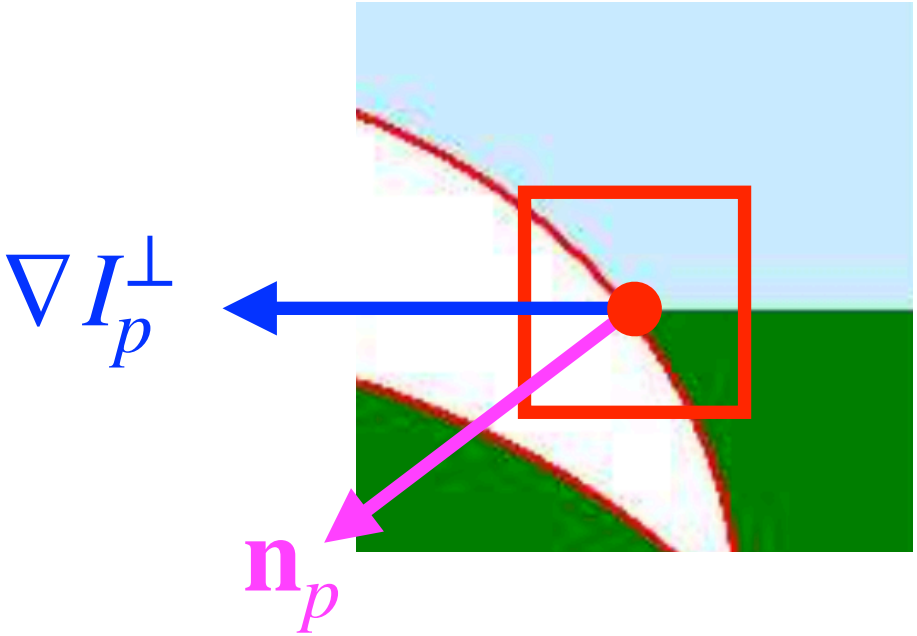
Calculating confidence term

$$C(p) = \frac{\sum_{q \in \text{filled pixels in patch}} C(q)}{\text{size of patch}}$$



Calculating data term

$$D(p) = \frac{|\nabla I_p^{\perp} \cdot \mathbf{n}_p|}{\alpha}$$



About $\nabla I_p^{\perp}$

- It's direction is perpendicular to the direction of the image gradient
- It's magnitude represents the intensity of the image gradient
- We'll revisit this if we have time…

Updated algorithm

```
Initialize C(p) for all pixels.
While the image is not filled:
  Calculate the priority for each pixel/patch
  that lies on the boundary of the mask.
  Get the patch with the highest priority.
  Find its closest match in the image.
  Fill the (unfilled part of the) patch with its
  match.
  Update the boundary of the mask.
```

Updated algorithm

```
Initialize C(p) for all pixels.
While the image is not filled:
  Calculate the priority for each pixel/patch
  that lies on the boundary of the mask.
  Get the patch with the highest priority.
  Find its closest match in the image.
  Fill the (unfilled part of the) patch with its
  match.
  Update the boundary of the mask.
```

The matching patch

What is a “close” match?

We want to find the patch with the least difference from ours:

$$\text{patch}(q^*) = \arg\min_{\text{patch}(q)} d(\text{patch}(p), \text{patch}(q))$$

What is a “close” match?

We want to find the patch with the least difference from ours:

$$\text{patch}(q^*) = \arg \min_{\text{patch}(q)} d(\text{patch}(p), \text{patch}(q))$$

Let d be the **sum-squared difference** of the filled pixels in the two patches.

Image inpainting algorithm

```
Initialize C(p) for all pixels.
While the image is not filled:
  Calculate the priority for each pixel/patch
  that lies on the boundary of the mask.
  Get the patch with the highest priority.
  Find the matching patch that minimizes the SSD.
  Fill the (unfilled part of the) patch with its
  match.
  Update the boundary of the mask.
```

Estimating ∇I

Sobel filters

Estimating image gradient

- Formally, given an image I and a patch centred about (0, 0), we want to estimate:

$$\nabla I(0,0) = \begin{bmatrix} \frac{\partial I}{\partial x}(0,0) & \frac{\partial I}{\partial y}(0,0) \end{bmatrix}$$

Sobel filters

original image

$$\|\nabla I_p\|$$

Sobel filters

I(-1,-1)	I(-1,0)	I(-1,1)
I(0,-1)	I(0,0)	I(0,1)
I(1,-1)	I(1,0)	I(1,1)

$$\frac{\partial f}{\partial x} = f(x+1, y) - f(x-1, y)$$

(up to a constant)

Sobel filters

I(-1,-1)	I(-1,0)	I(-1,1)
I(0,-1)	I(0,0)	I(0,1)
I(1,-1)	I(1,0)	I(1,1)

(up to a constant)

Sobel filters

I(-1,-1)	I(-1,0)	I(-1,1)
I(0,-1)	I(0,0)	I(0,1)
I(1,-1)	I(1,0)	I(1,1)

$$\frac{\partial f}{\partial x} \approx f(x+1, y) - f(x-1, y)$$

(up to a constant)

$$est_1 = I(1, -1) - I(-1, -1)$$
$$est_2 = I(1, 0) - I(-1, 0)$$
$$est_3 = I(1, 1) - I(1, -1)$$

$$\frac{\partial f}{\partial x} = \text{weighted_average}(est_1, est_2, est_3)$$

Sobel filters

I(-1,-1)	I(-1,0)	I(-1,1)
I(0,-1)	I(0,0)	I(0,1)
I(1,-1)	I(1,0)	I(1,1)

(up to a constant)

$$est_1 = I(1, -1) - I(-1, -1)$$
$$est_2 = I(1, 0) - I(-1, 0)$$
$$est_3 = I(1, 1) - I(1, -1)$$

$$\frac{\partial f}{\partial x} = \text{weighted_average}(est_1, est_2, est_3)$$

Sobel filters

I(-1,-1)	I(-1,0)	I(-1,1)
I(0,-1)	I(0,0)	I(0,1)
I(1,-1)	I(1,0)	I(1,1)

(up to a constant)

$$\frac{\partial f}{\partial x} = \text{weighted_average}(est_1, est_2, est_3)$$

Sobel filters

1. Recall (up to a const.) $\frac{\partial f}{\partial x} = f(x+1, y) - f(x-1, y)$
2. Apply across each column to get *n* approximations.
3. Weigh approximations closer to centre more heavily.
4. Get weighted average.
5. Repeat for partial-f partial-y.

Sobel filters

1. Recall (up to a const.)
2. Apply across each column to get *n* approximations.
3. Weigh approximations closer to centre more heavily.
4. Get weighted average.
5. Repeat for partial-f partial-y.

Sobel filters

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * I$$

The * symbol just means element-wise multiply, then sum over everything.

Estimating ∇l

Taylor series

Taylor Series

$$f(x) = f(0) + xf'(0) + x^2\frac{f''(0)}{2!} + H.O.T$$

Taylor Series

$$I(x, y) \approx I(0, 0) + x\frac{\partial I}{\partial x}(0, 0) + y\frac{\partial I}{\partial y}(0, 0) + H\,.\,O\,.\,T$$

Taylor Series

unknown

unknown

$$I(x,y) - I(0,0) \approx x\frac{\partial I}{\partial x}(0,0) + y\frac{\partial I}{\partial y}(0,0) + H.O.T$$

known

known

known

Taylor Series

$$I(x, y) - I(0, 0) = x\frac{\partial I}{\partial x}(0, 0) + y\frac{\partial I}{\partial y}(0, 0) + H\,.\,O\,.\,T$$

Substitute (x, y) to be each of the eight neighbours about (0, 0) to get eight linear equations!

$$I(-1, -1) - I(0, 0) = -\frac{\partial I}{\partial x}(0, 0) - \frac{\partial I}{\partial y}(0, 0)$$

$$I(-1, 0) - I(0, 0) = -\frac{\partial I}{\partial x}(0, 0) + 0$$

$$I(-1, 1) - I(0, 0) = -\frac{\partial I}{\partial x}(0, 0) + \frac{\partial I}{\partial y}(0, 0)$$

$$\vdots$$

$$I(1, -1) - I(0, 0) = \frac{\partial I}{\partial x}(0, 0) - \frac{\partial I}{\partial y}(0, 0)$$

$$I(1, 0) - I(0, 0) = \frac{\partial I}{\partial x}(0, 0) + 0$$

$$I(1, 1) - I(0, 0) = \frac{\partial I}{\partial x}(0, 0) + \frac{\partial I}{\partial y}(0, 0)$$

Taylor Series

$$I(x, y) - I(0, 0) = x\frac{\partial I}{\partial x}(0, 0) + y\frac{\partial I}{\partial y}(0, 0) + H\,.\,O\,.\,T$$

$$X = \begin{bmatrix} -1 & -1 \\ -1 & 0 \\ -1 & 1 \\ \vdots & \vdots \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad d = \begin{bmatrix} \frac{\partial I}{\partial x}(0, 0) \\ \frac{\partial I}{\partial y}(0, 0) \end{bmatrix} \qquad Xd = I - I(0, 0)$$

Taylor Series: weighting equations

1 / distance-to-center

1	$\sqrt{2}$	1
$\sqrt{2}$		$\sqrt{2}$
1	$\sqrt{2}$	1

2D Gaussian

$$\Omega(i, j) = \sqrt{2\pi} e^{-\frac{i^2 + j^2}{2}}$$