Ukrainian Catholic University

Image Inpainting

Presented by Mykola Vysotskyi, Severyn Shykula, Oleksandr Ivaniuk

Image Inpainting

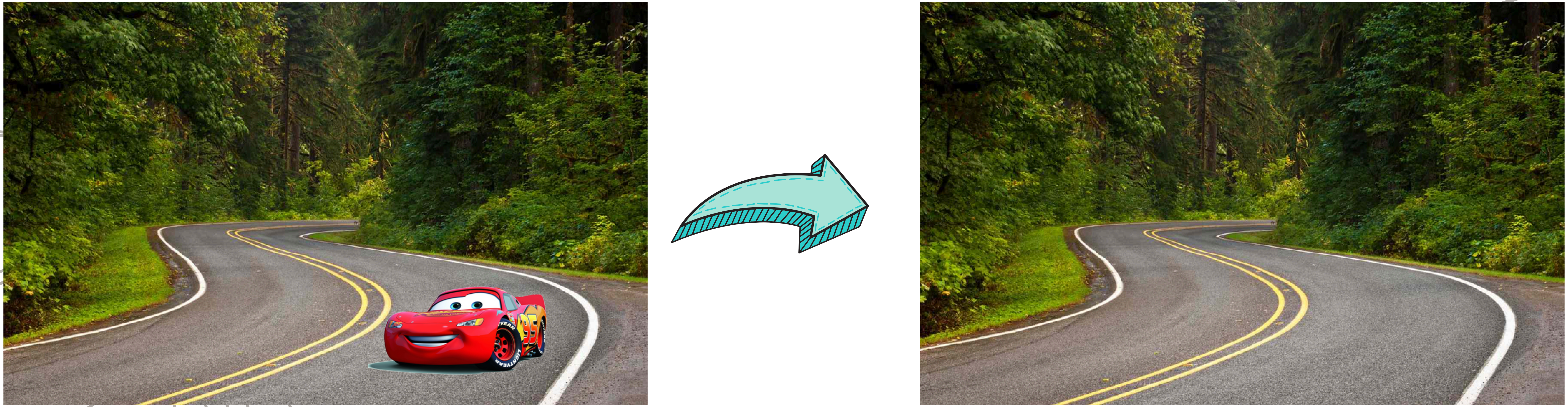
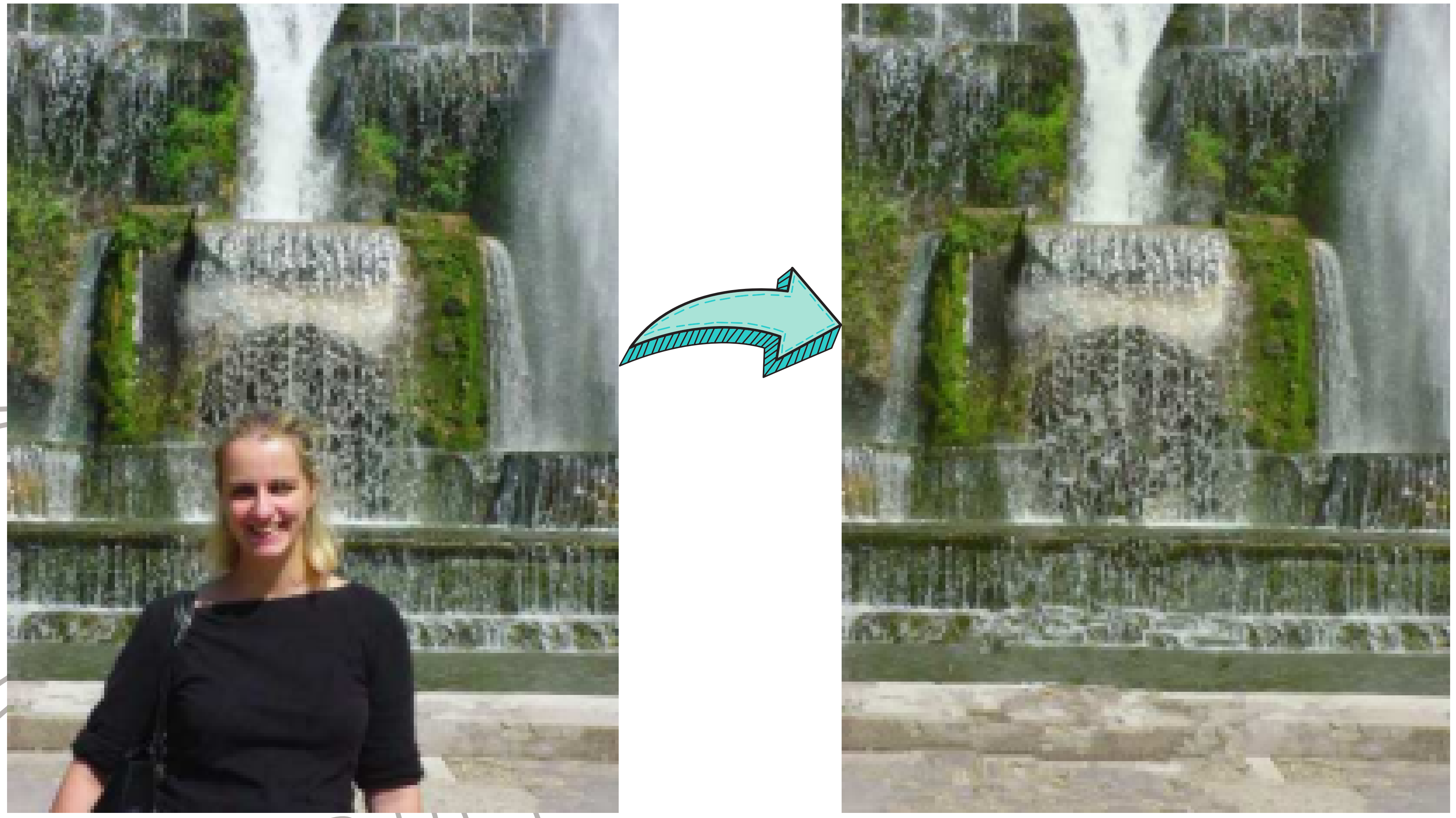


Image Inpainting



Main aim:

- Learn how to apply LA to image processing
- Implement classical image inpainting algorithm
- Compare results with OpenCV method

Implementation of paper

Region Filling and Object Removal by Exemplar-Based Image Inpainting

A. Criminisi, P. Perez and K. Toyama ´ Microsoft Research, Cambridge (UK) and Redmond (US)

This paper:

- Introduces an algorithm designed for removing large objects from digital images and convincingly filling the resultant gaps.
- Combines texture synthesis and inpainting techniques to handle both texture and structural details in images.
- Ensures computational efficiency by employing a block-based sampling process.
- Demonstrates the algorithm's effectiveness and robustness using both real and synthetic image examples, capable of removing objects and repairing thin scratches.
- Shows adaptability to various shapes and sizes of target regions in images.

Φ -- known image part

Ω -- unknown image part

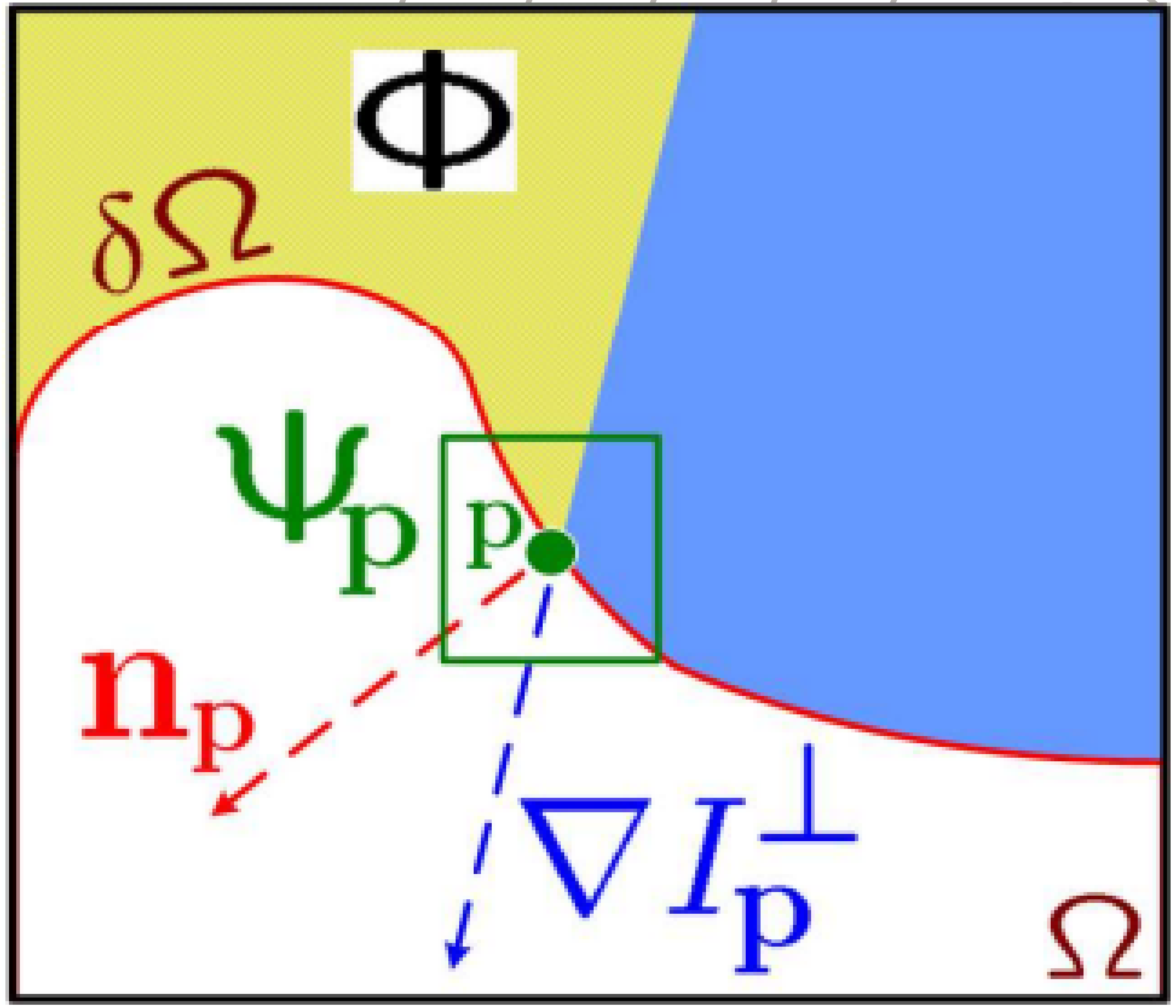
$\delta\Omega$ -- border between known and unknown parts(front)

$\mathbf{p}$ -- point on front

$\nabla I_p^{\perp}$ -- direction and intensity at point

$\mathbf{n_p}$ -- normal direction to the front point

Ψ_p -- patch centred in front point

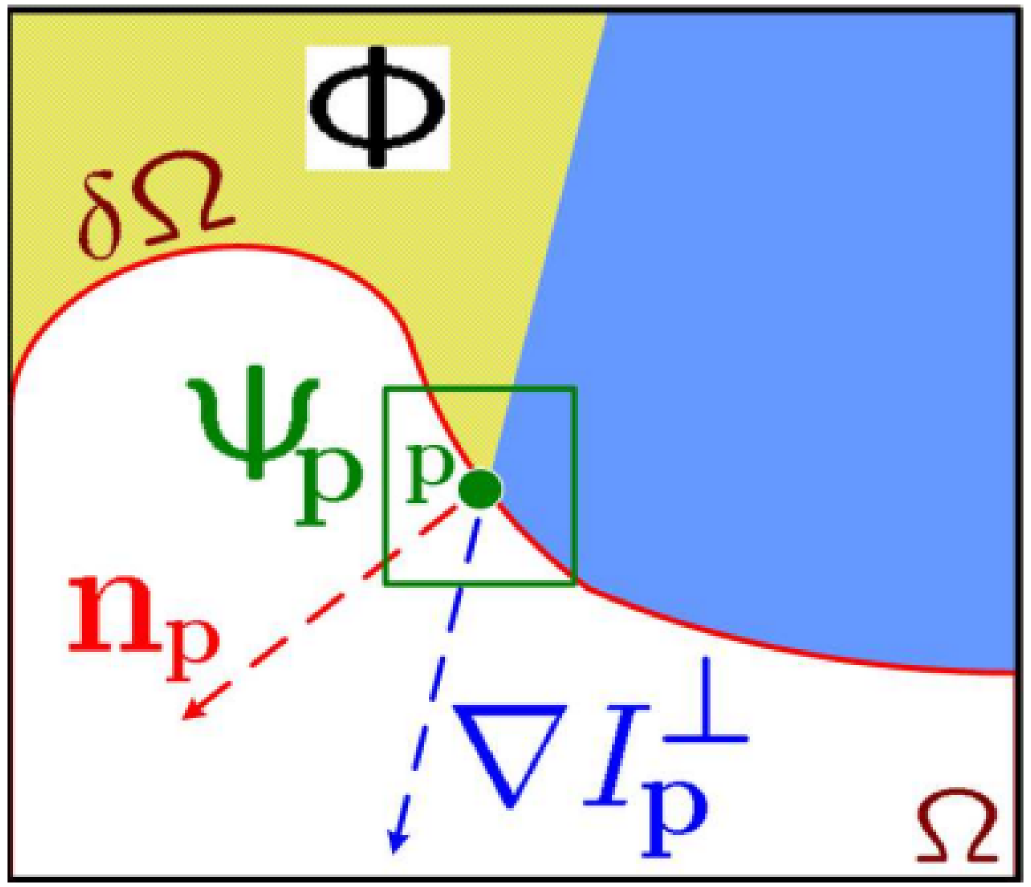


Confidence term

$$C(\mathbf{p}) = \frac{\sum_{q \in \Psi_p \cap (I - \Omega)} C(\mathbf{q})}{|\Psi_p|}$$

- measure of the amount of reliable information surrounding the pixel **p**
- sum of conficences of all known pixels in patch with center p
- $|\Psi_p|$ - area of the patch

Data term



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

- **isophote** is a line on an image that connects points of equal brightness
- **data term** is strength of isophotes hitting the front
- α - normalization factor

Priority

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

- for each pixel **priority** term is defined as product **confidence** and **data** terms
- patches with central pixel with higher priority are filled first

Best patch choice

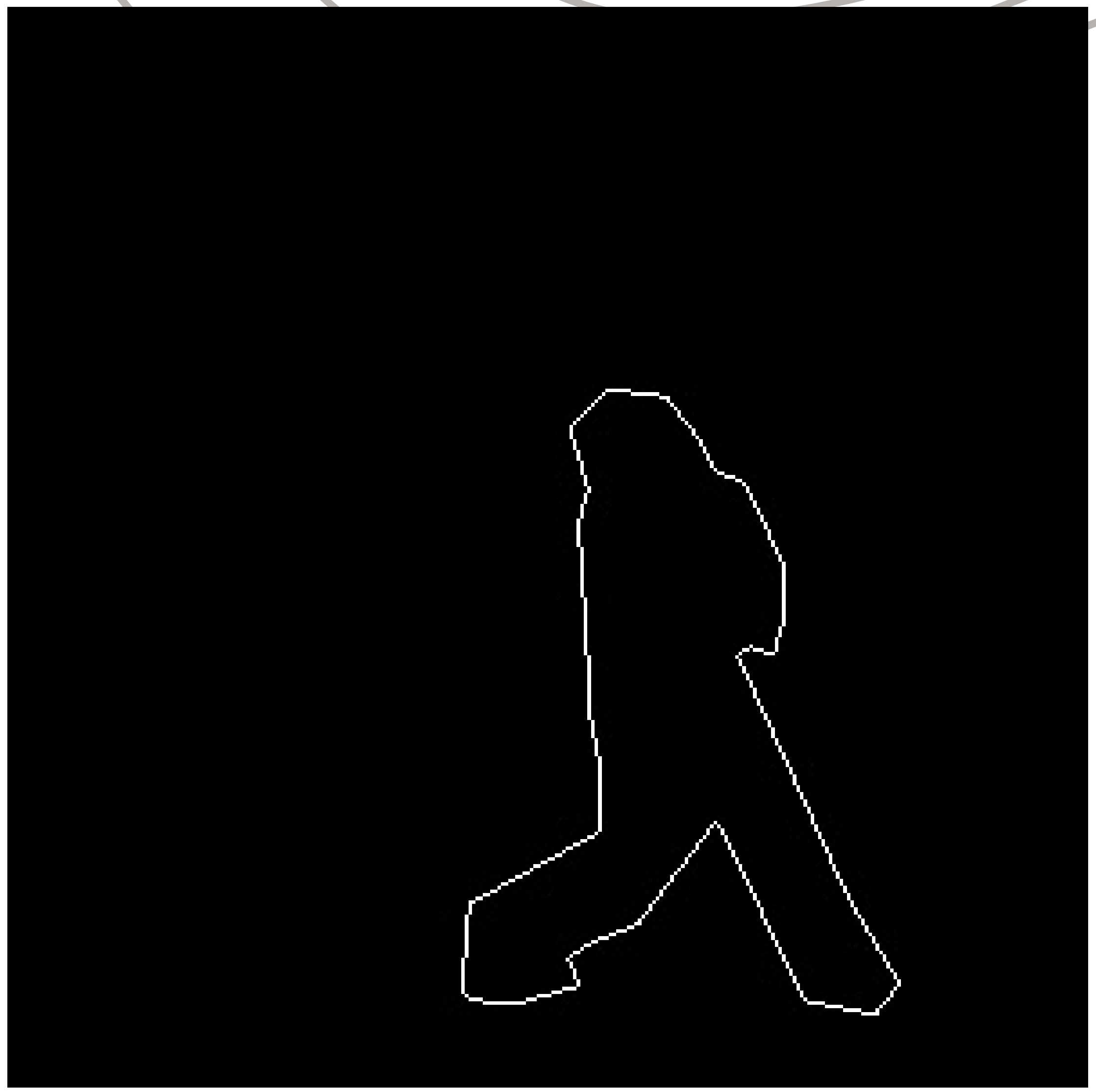
Patch error = color distance + patch coordinades distance

- after finding pixel with highest priority we need to find best patch from known part of image
- we do this by minimizing **patch error**

How we computed?

Laplacian filter

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



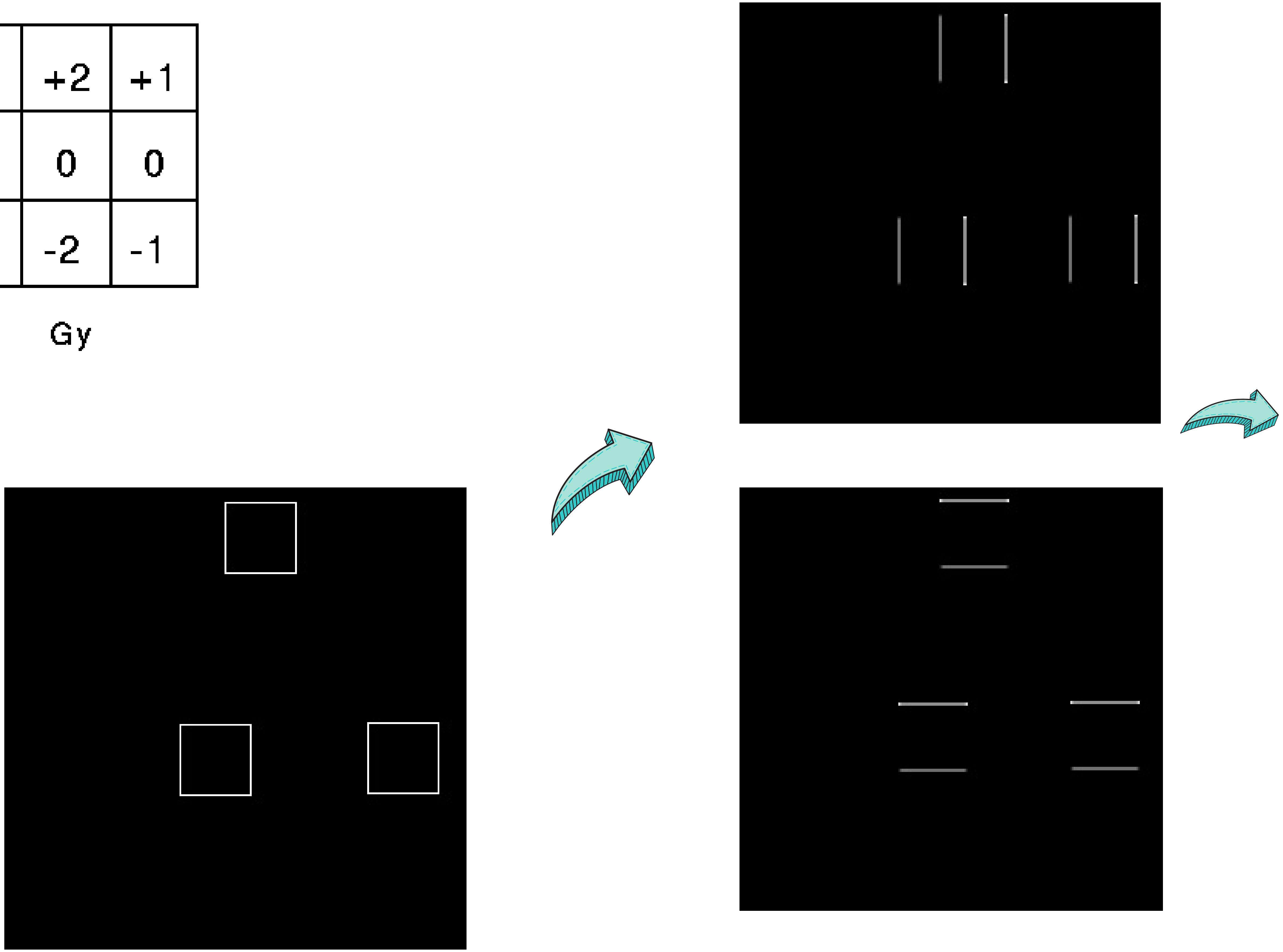
Sobel filter

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy



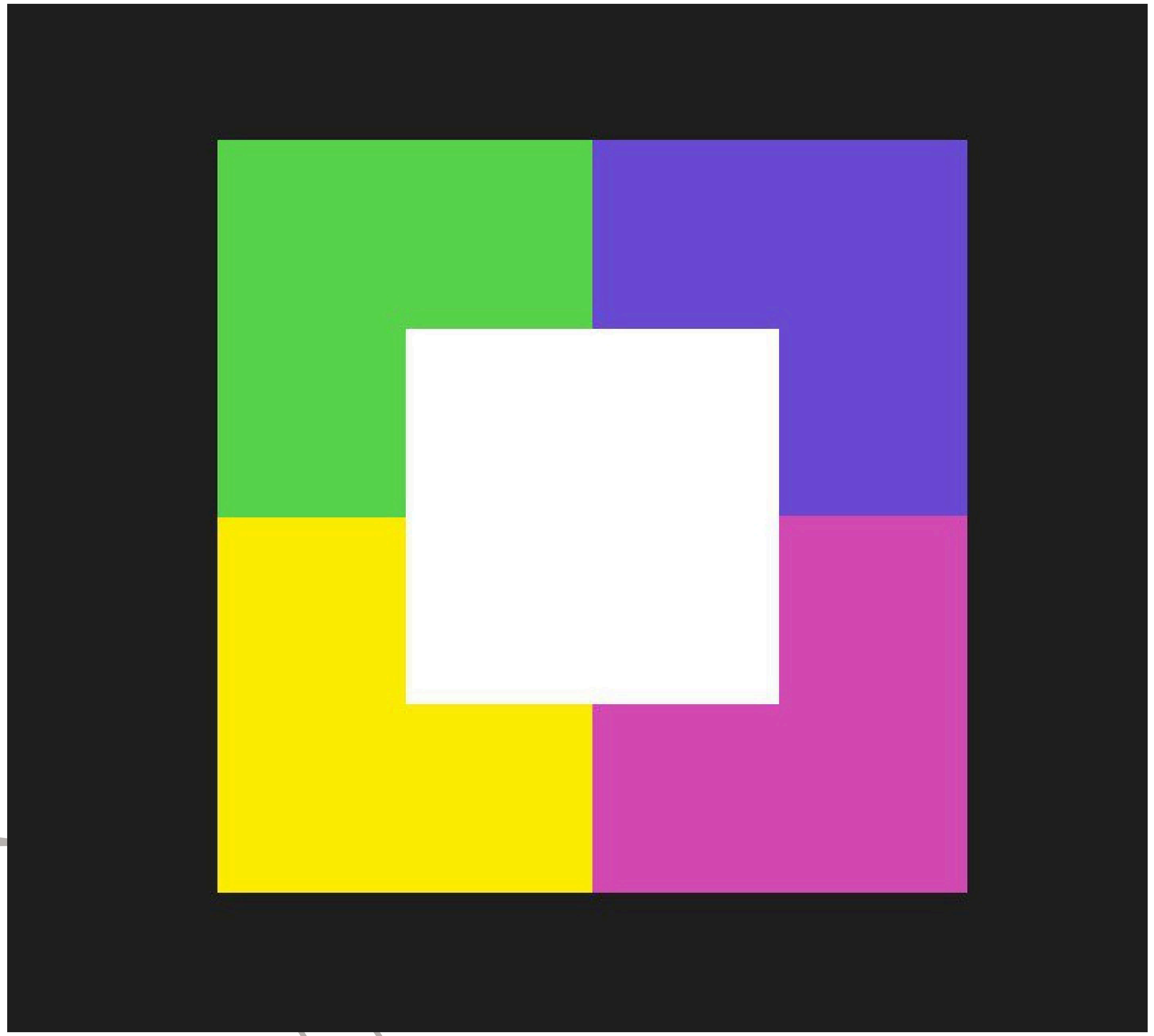
Normal direction

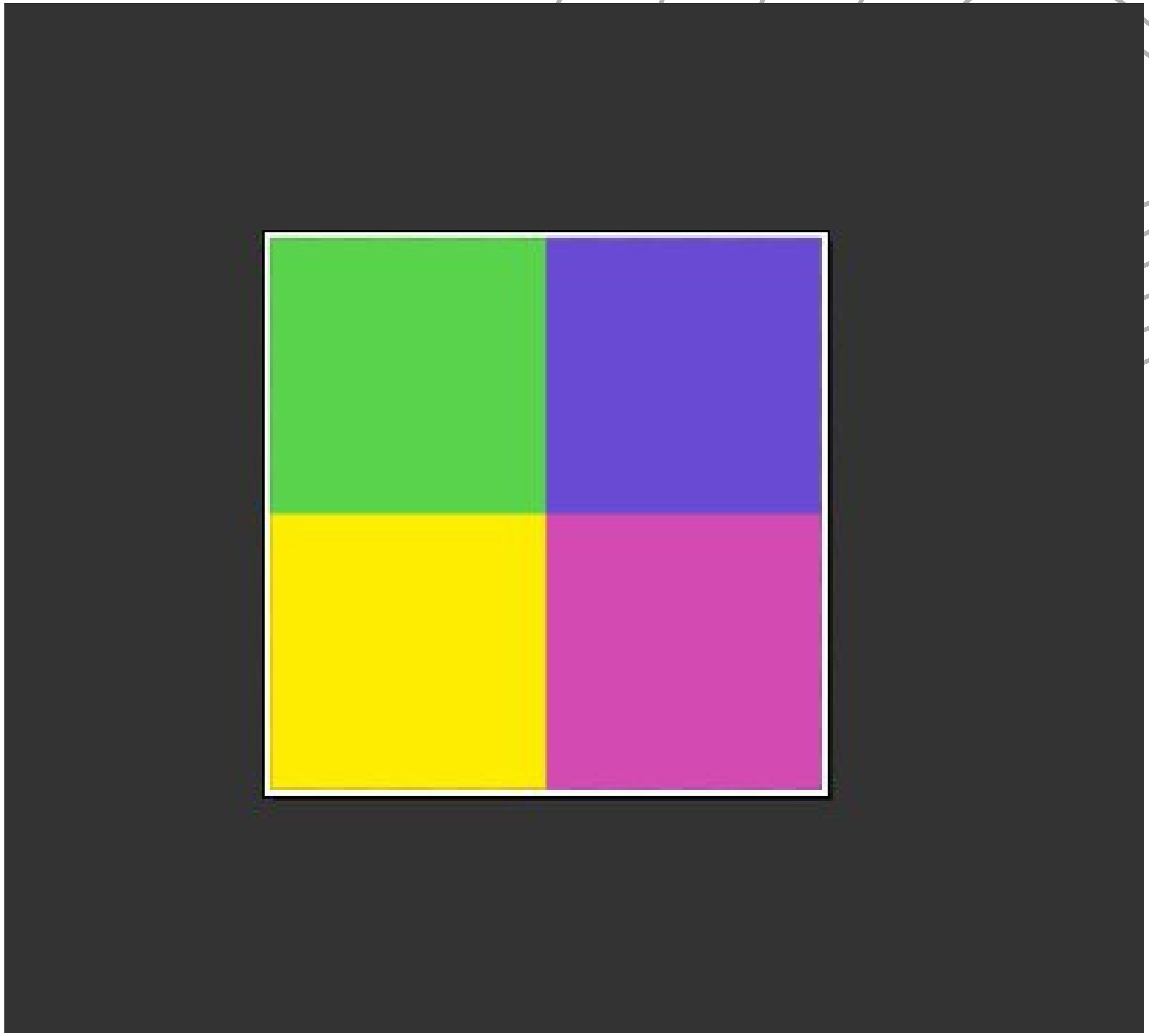
Our implementation

Python

- scipy
- numpy
- PIL
- OpenCV (for comparison with Telea’s method)

Our results





Our results

Testing

Metrics:

1. PSNR (Peak Signal-to-Noise Ratio)
2. SSI (Structural Similarity Index)
3. IQI (Image Quality Index)

Testing

PSNR (Peak Signal-to-Noise Ratio)

PSNR is a ratio between the maximum possible power of a signal (represented by pixel values). A higher PSNR value generally indicates that the degradation is less noticeable. Commonly, a PSNR in the range of 30-50 dB.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}|I(i,j) - K(i,j)|^2$$

$$PSNR = 10\log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

Testing

SSI (Structural Similarity Index)

SSIM considers luminance (*l*), contrast (*c*), and structural similarity between two images (*s*). Output is a value from -1 to 1, which represents the degree of similarity.

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \qquad c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \qquad s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}$$

$$\mathrm{SSIM}(x,y) = l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma}$$

x,y: two images being compared.
μx,μy: average luminance values of images x and y.
σ: contrast within the images.
*C*1,*C*2,*C*3: constants used to stabilize the division.
α, β, γ: weights.

Testing

UQI (Universal Quality Index)

The Universal Quality Index is calculated based on the structural similarity between two images. The formula typically involves comparing the luminance and contrast of corresponding pixels in the images. Output from 0 to 1 and represents similarity.

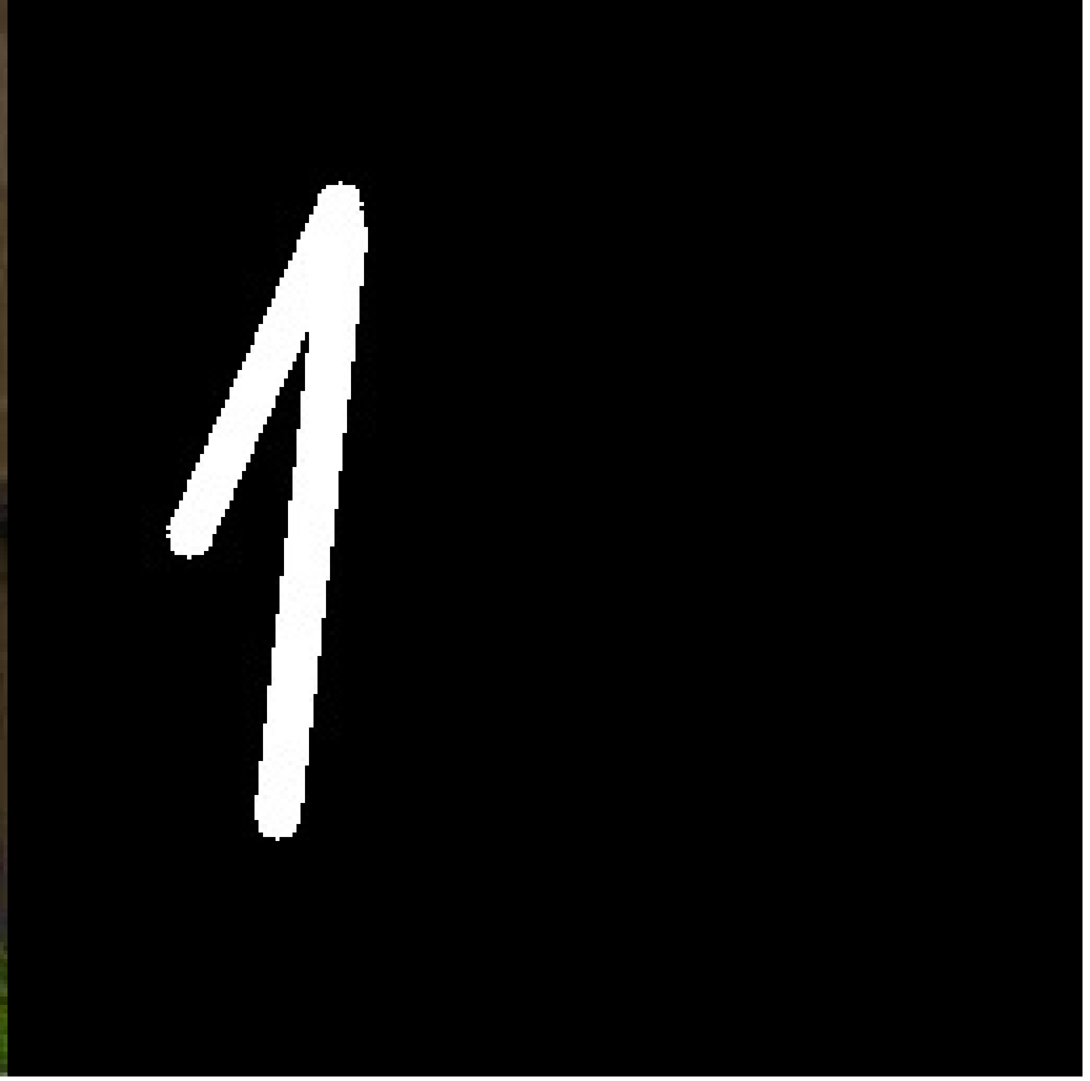
$$UQI(X,Y) = \frac{4 \times \sigma_{xy} \times \mu_x \times \mu_y}{(\sigma_x^2 + \sigma_y^2) \times (\mu_x^2 + \mu_y^2)}$$

σxy: covariance of the pixel intensities of images *X* and *Y*.
*σx*2, *σy*2: variances of the pixel intensities of images *X* and *Y*
μx, *μy*: means of the pixel intensities of images *X* and *Y*.

Data

Places2 dataset

Comparison

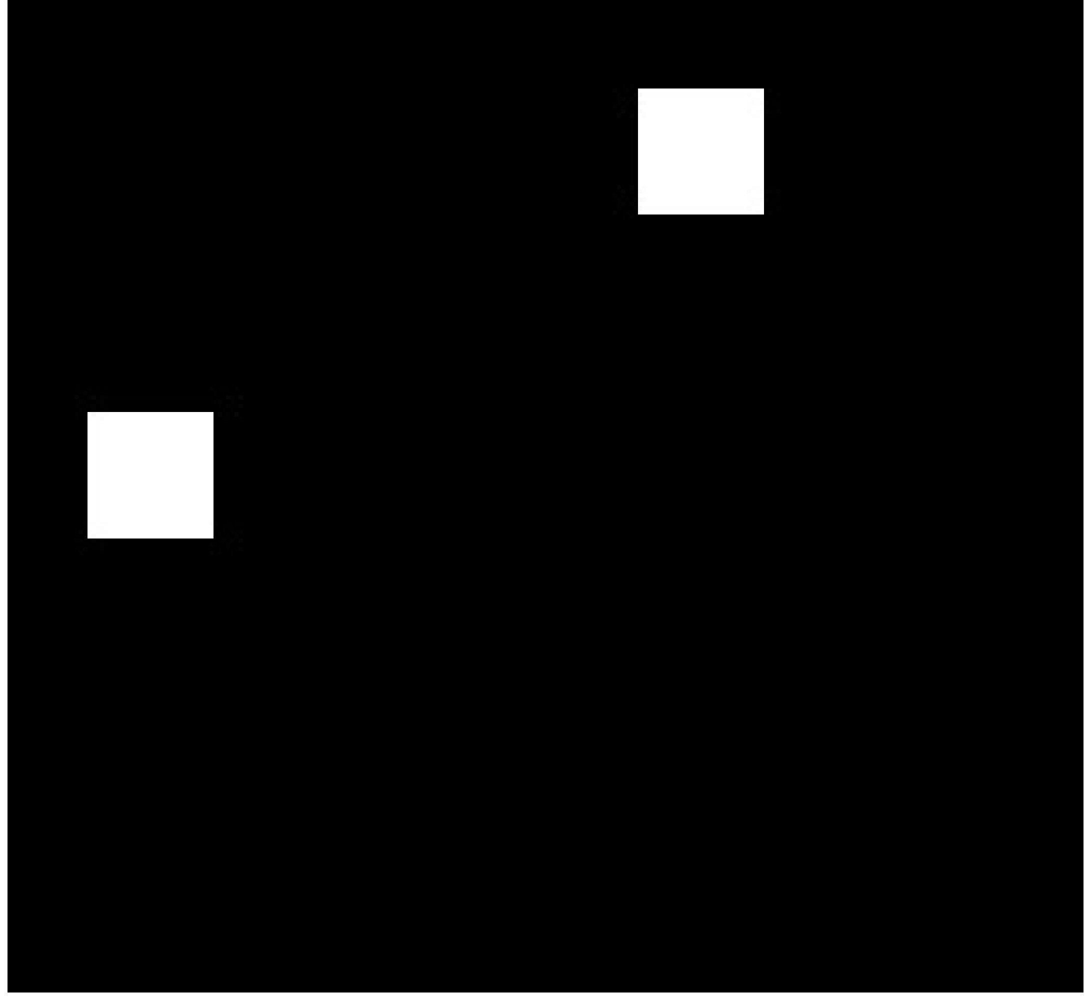


Telea

Implemented

Telea
PSNR:42.755 dB
SSI: 0.98009
UQI: 0.9975
Implemented
PSNR:43.656 dB
SSI: 0.98062
UQI: 0.99672

Comparison

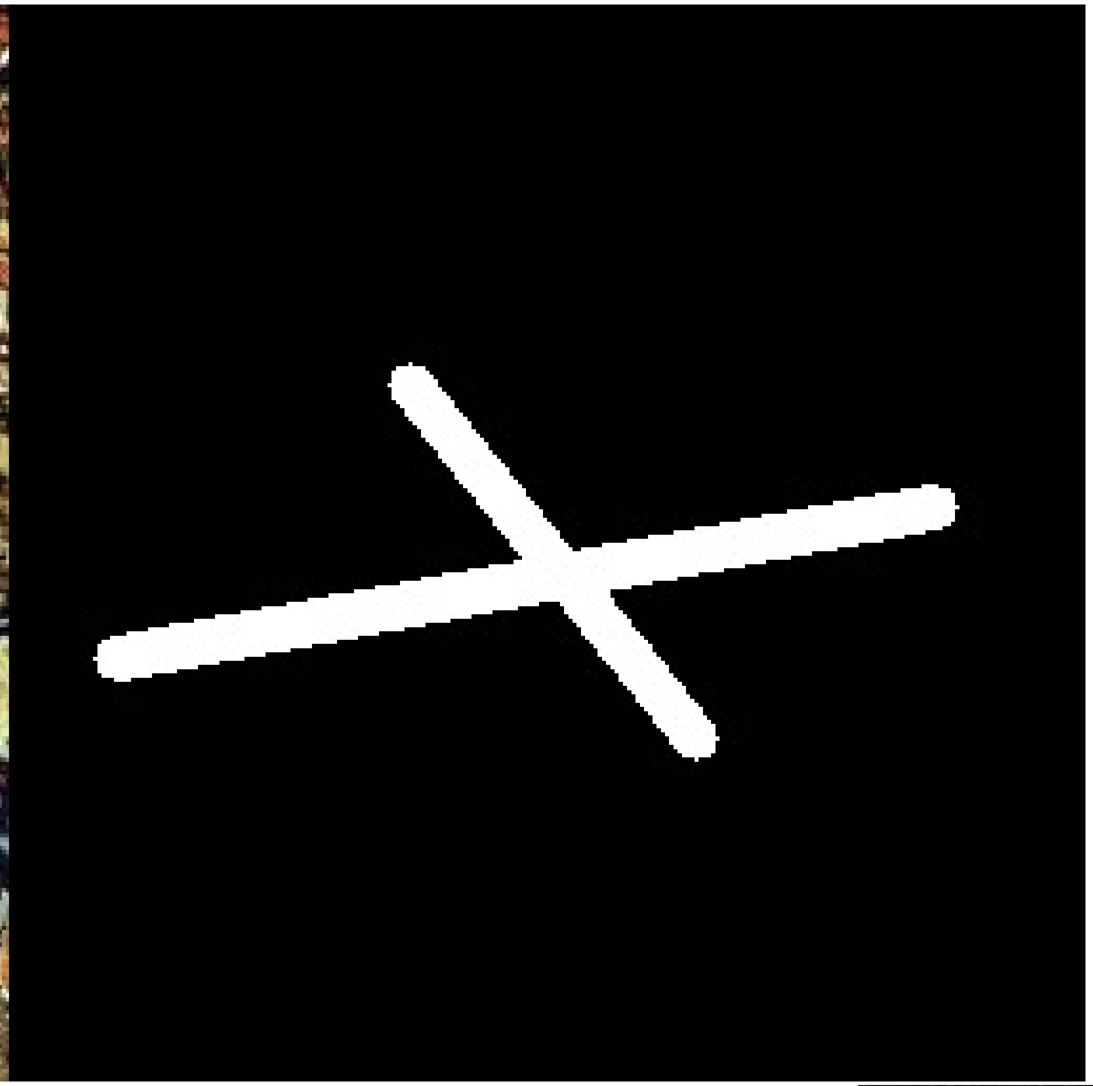


Telea

Implemented

Telea
PSNR:42.755 dB
SSI: 0.98009
UQI: 0.99753
Implemented
PSNR:43.65 dB
SSI: 0.98062
UQI: 0.99672

Comparison

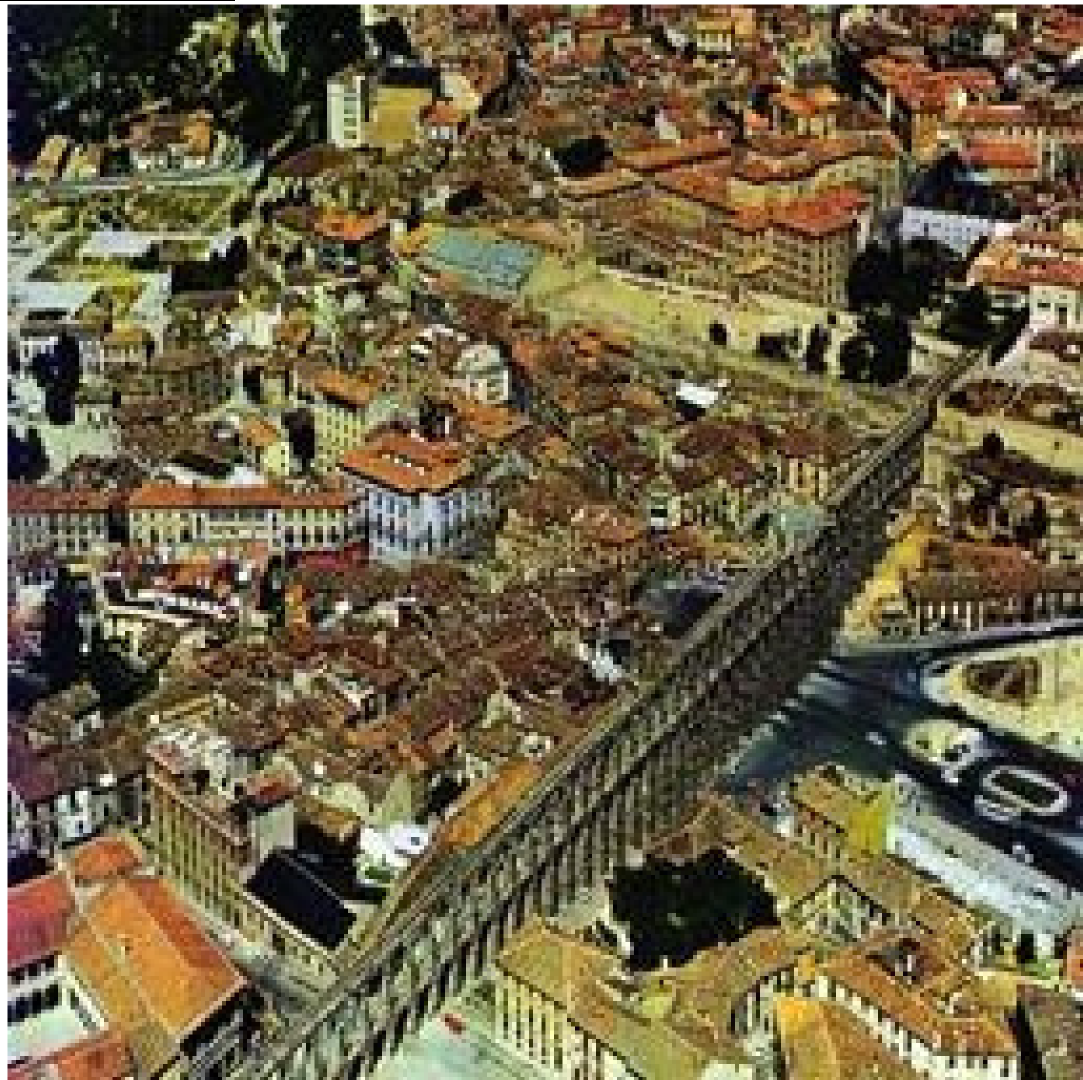


Telea

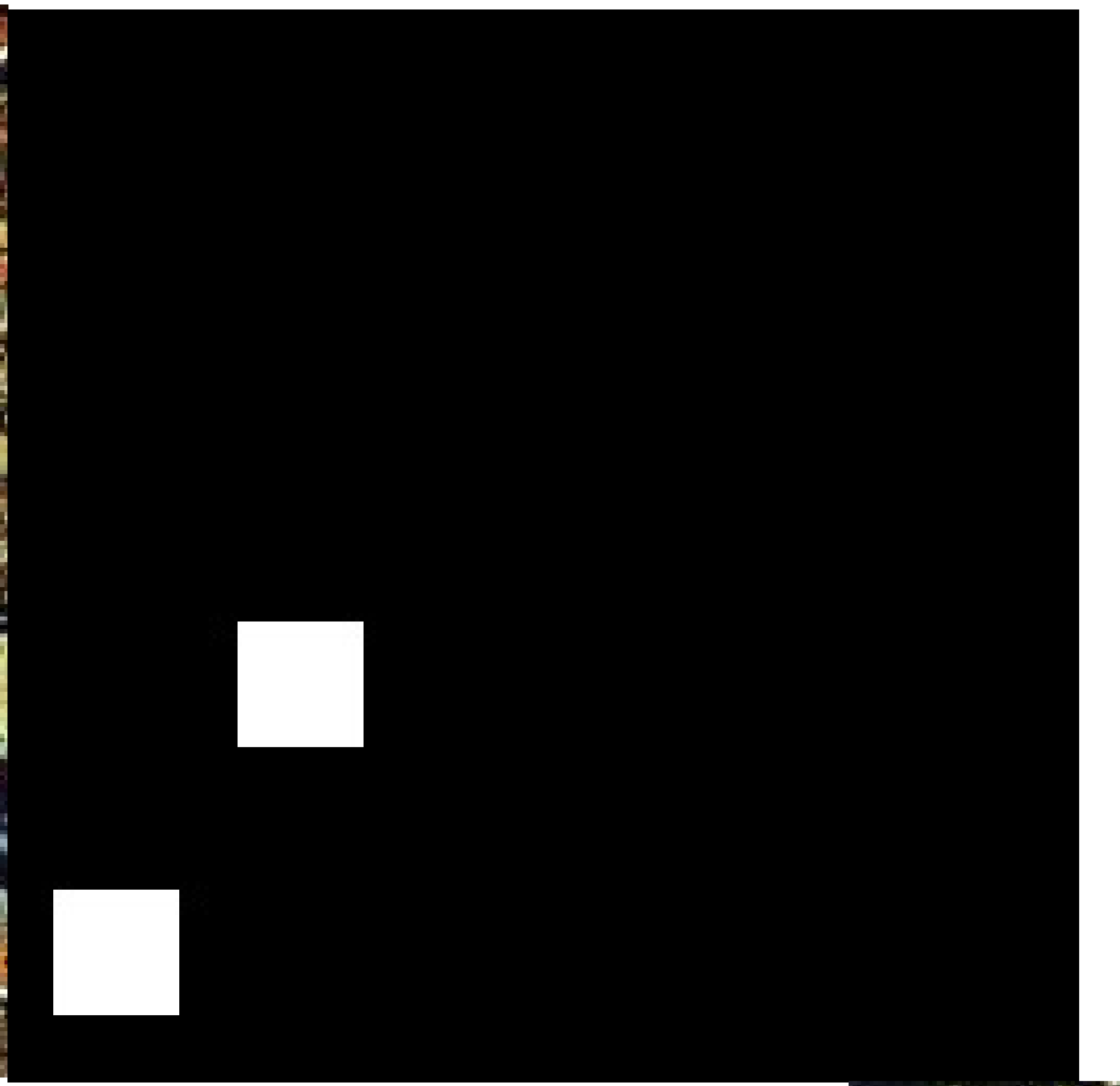
Implemented

Telea
PSNR:38.91 dB
SSI: 0.95923
UQI: 0.99062

Implemented
PSNR:39.01 dB
SSI: 0.94995
UQI: 0.9858



Comparison



Telea

Implemented

full image: Telea

PSNR:40.338 dB

SSI: 0.9767

UQI: 0.9954

Crimnisi

PSNR:41.608 dB

SSI: 0.9732

UQI: 0.9928

patch: Telea

PSNR:31.464 dB

SSI: 0.7050

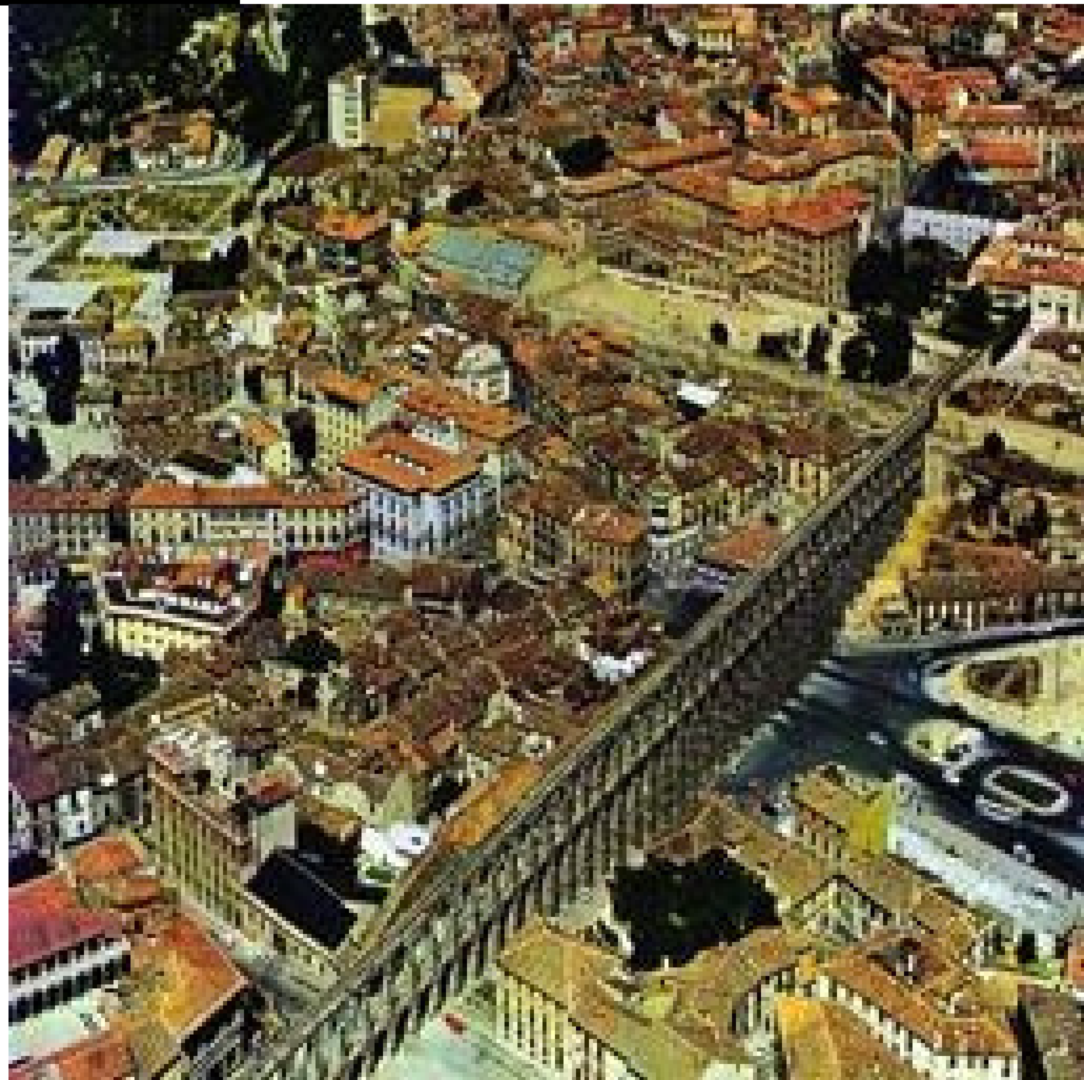
UQI: 0.9491

Crimnisi

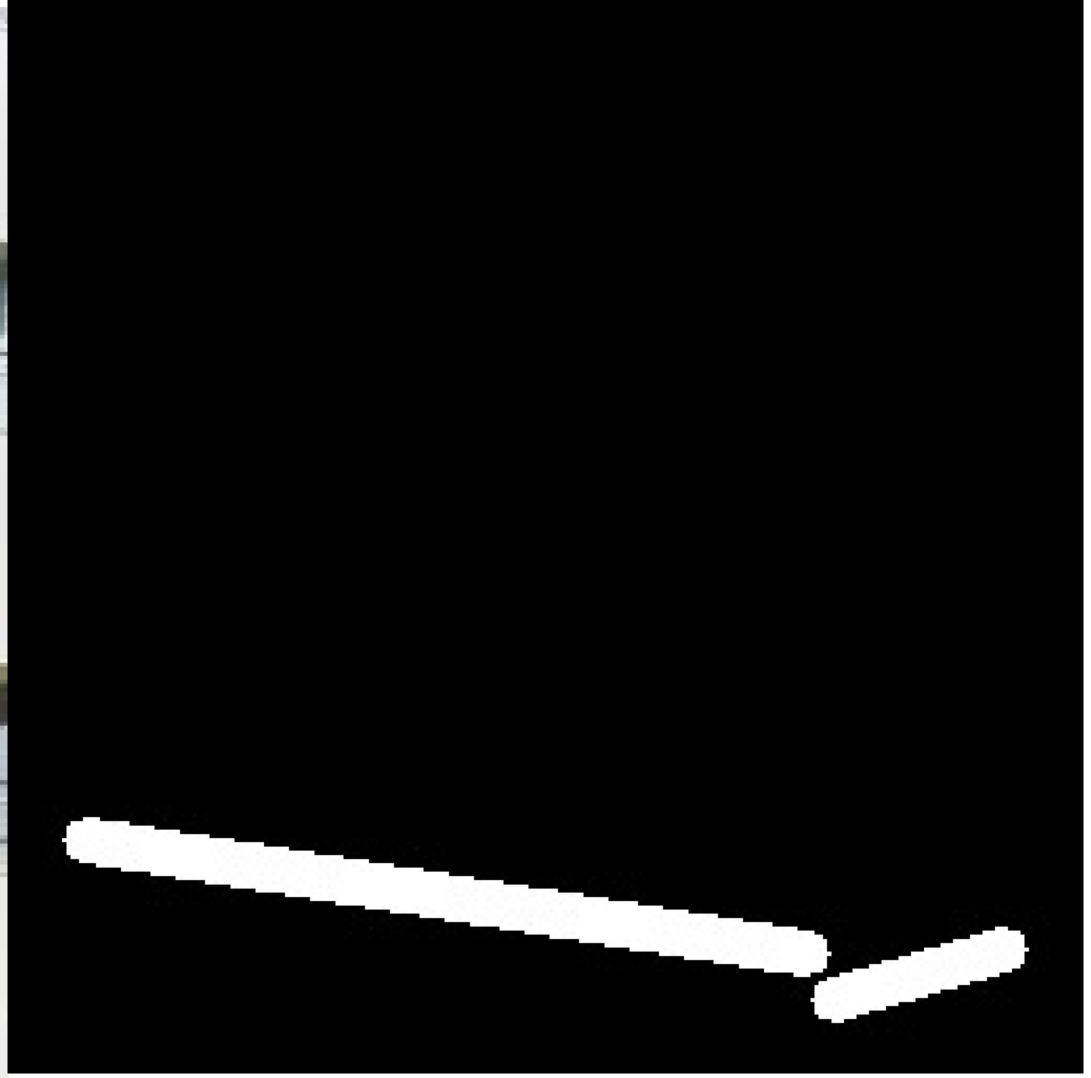
PSNR:31.845 dB

SSI: 0.6254

UQI: 0.9050



Comparison

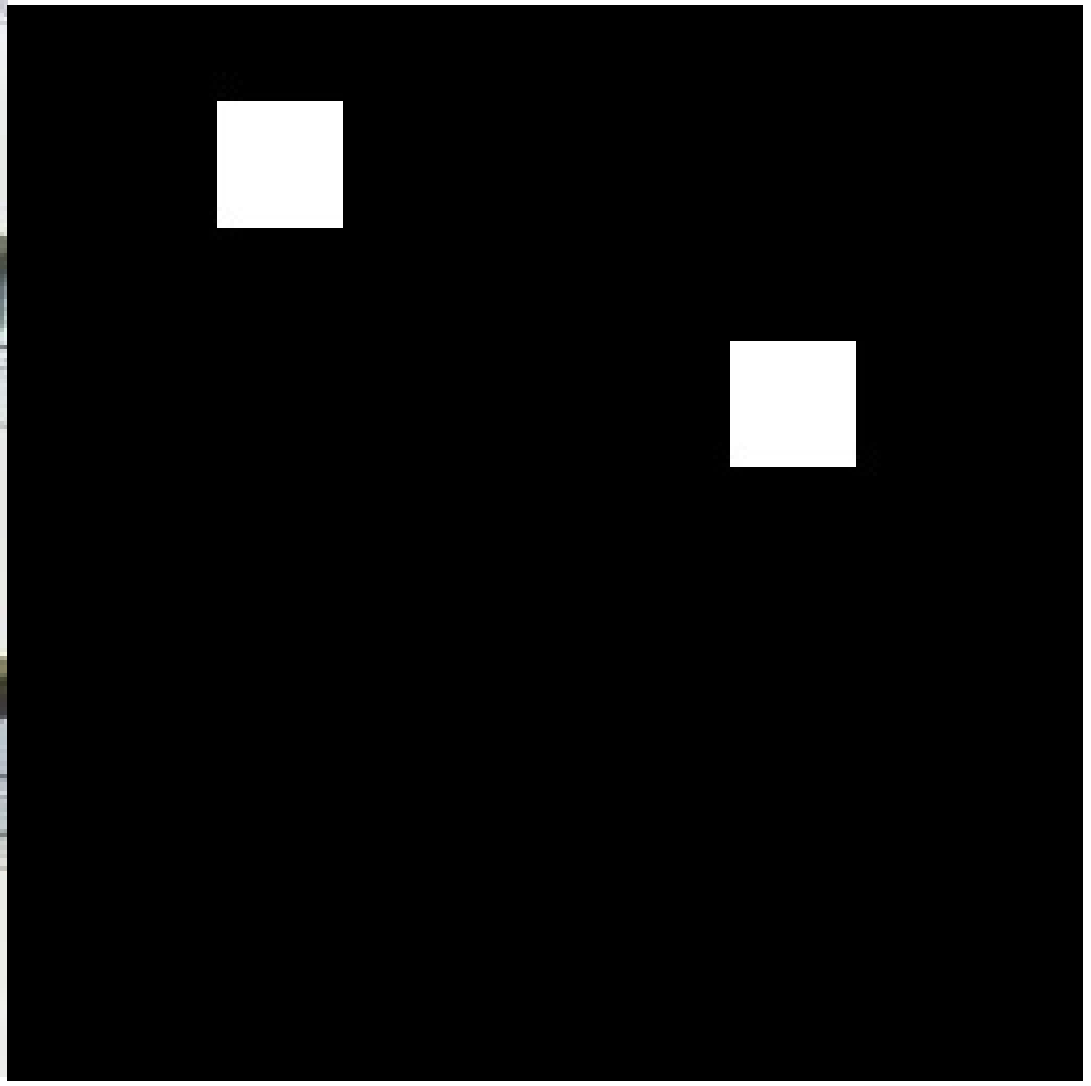


Telea

Telea
PSNR:44.417 dB
SSI: 0.98586
UQI: 0.99656
Implemented
PSNR:44.659 dB
SSI: 0.98647
UQI: 0.99667

Implemented

Comparison



Telea

Implemented

full image:

Telea
PSNR:43.163 dB
SSI: 0.97879
UQI: 0.99814
Implemented
PSNR:43.739 dB
SSI: 0.98780
UQI: 0.9990

patch:

Telea
PSNR:32.619 dB
SSI: 0.7278
UQI: 0.9800
Implemented
PSNR:33.266 dB
SSI: 0.8047
UQI: 0.9902

Conclusion

- Implemented method works better if there are linear structures in the image
- OpenCV method works better with complex geometric structures

Literature

- "Region Filling and Object Removal by Exemplar-Based Image Inpainting" A. Criminisi 2004.
- Study on Image Inpainting Algorithms, Fan, Qian 2018
- Finite difference methods in image processing
- Computational Foundation of Cognitive Science - Frank Keller

Thank You!