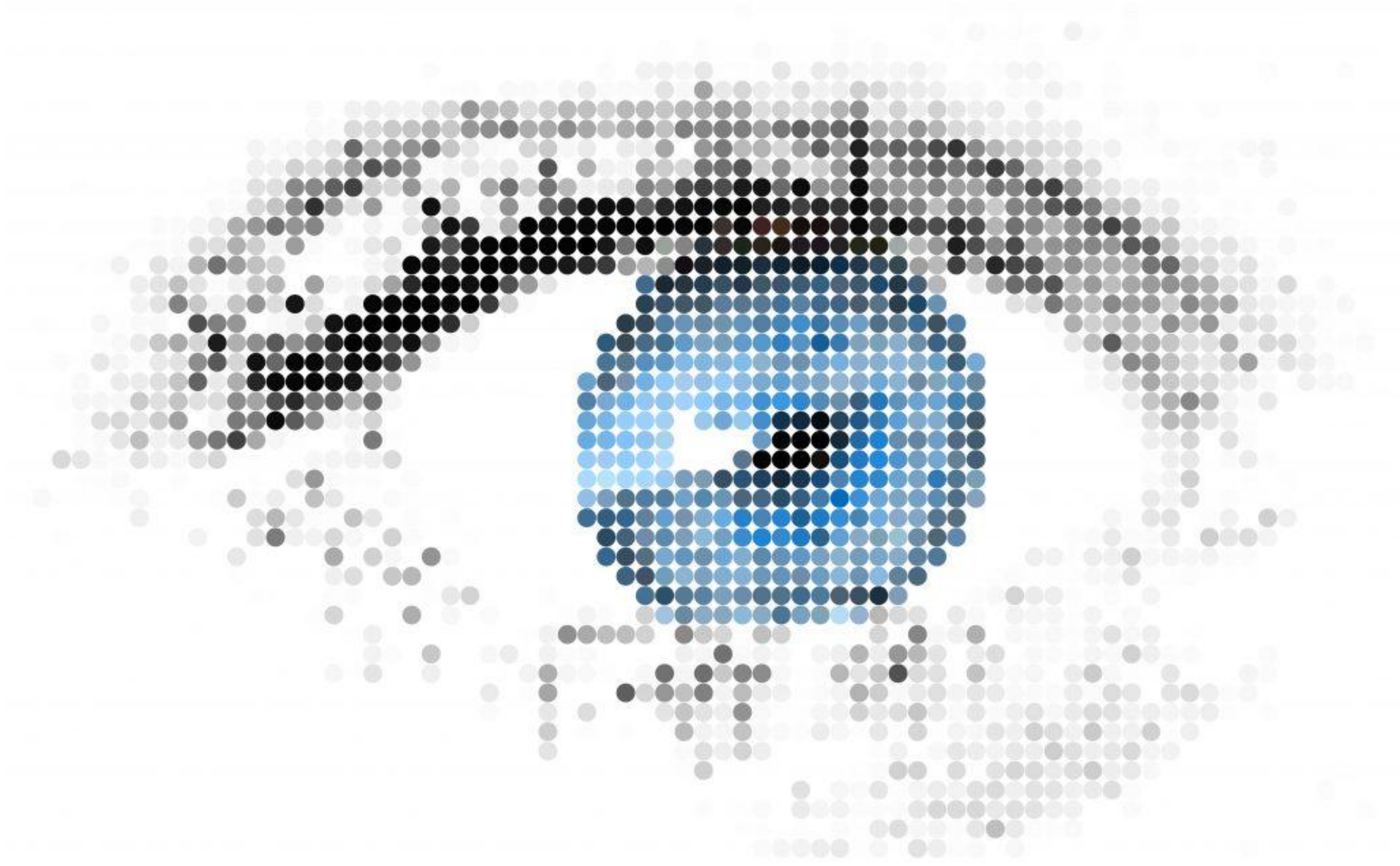


MCV M2 Project

Optimization and inference for Computer Vision

Pérez, Ferran
Poveda, Jonatan
Group X

OVERVIEW

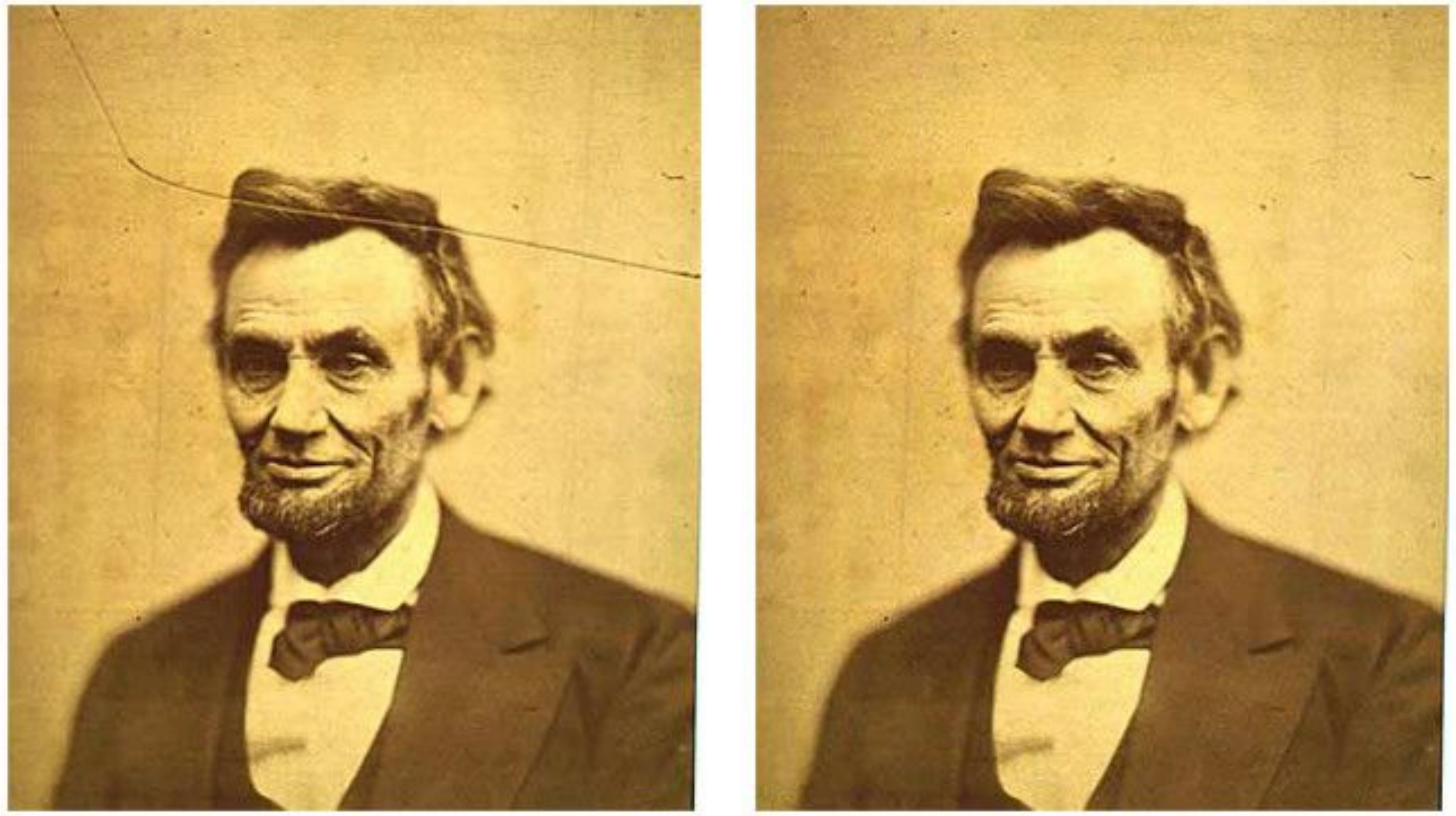
- Week 1: Inpainting
- Week 2: Poisson editing
- Week 3: Segmentation
- Week 4: Graphical models

Week 1

Image Inpainting

What is image inpainting?

- We have a deteriorated image
- We want to reconstruct it realistically
- We want to remove some object(s)
- We select the region with a mask
- We use optimization to generate the reconstructed image



Criteria

- The output image,
  - should look natural (smoothness)
  - should be similar to the original (difference)

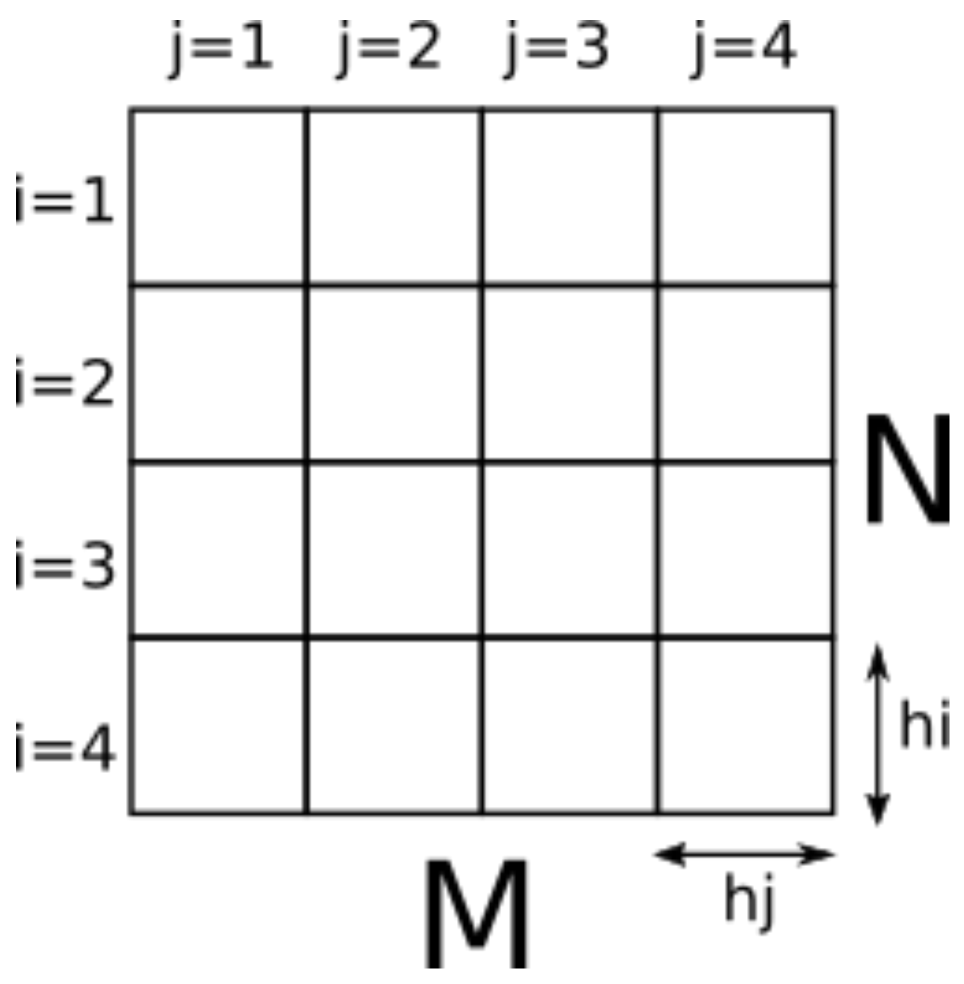
Energy functional to optimize

$$J(u) = \int_{\Omega} \frac{1}{2}|\nabla u|^2 + \frac{1}{2\lambda}|u - f|^2 dx$$

INPAINTING

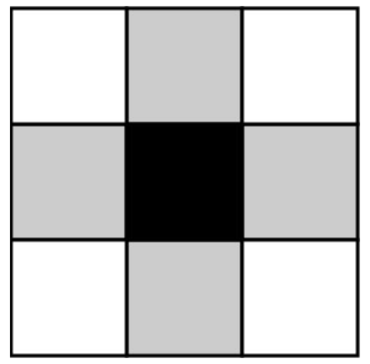
Computation

- Strategy: Find the necessary condition for the extremum $\longrightarrow$ $\nabla F(\mathbf{u}) = 0$
- We use the Euler-Laplace equation to solve it
- So, we have a system of equations **Au** = **b** to solve
  - **A** : sparse matrix of shape (*P*,*P*),
    where *P* = N · M, and (*N*,*M*) the image shape
  - **u**: "answer" image that minimizes the energy
  - **b**: output vector (0 inside region, *f(i,j)* otherwise)



Setup and parameters

- The output image (**u**) should be equal to the original (**f**),
  - outside the region to be inpainted
  - on the external boundaries of the region to be inpainted
- In the region to be inpainted,
  - we use the Laplacian to approximate **J(u)** with 5 values (4-connectivity)

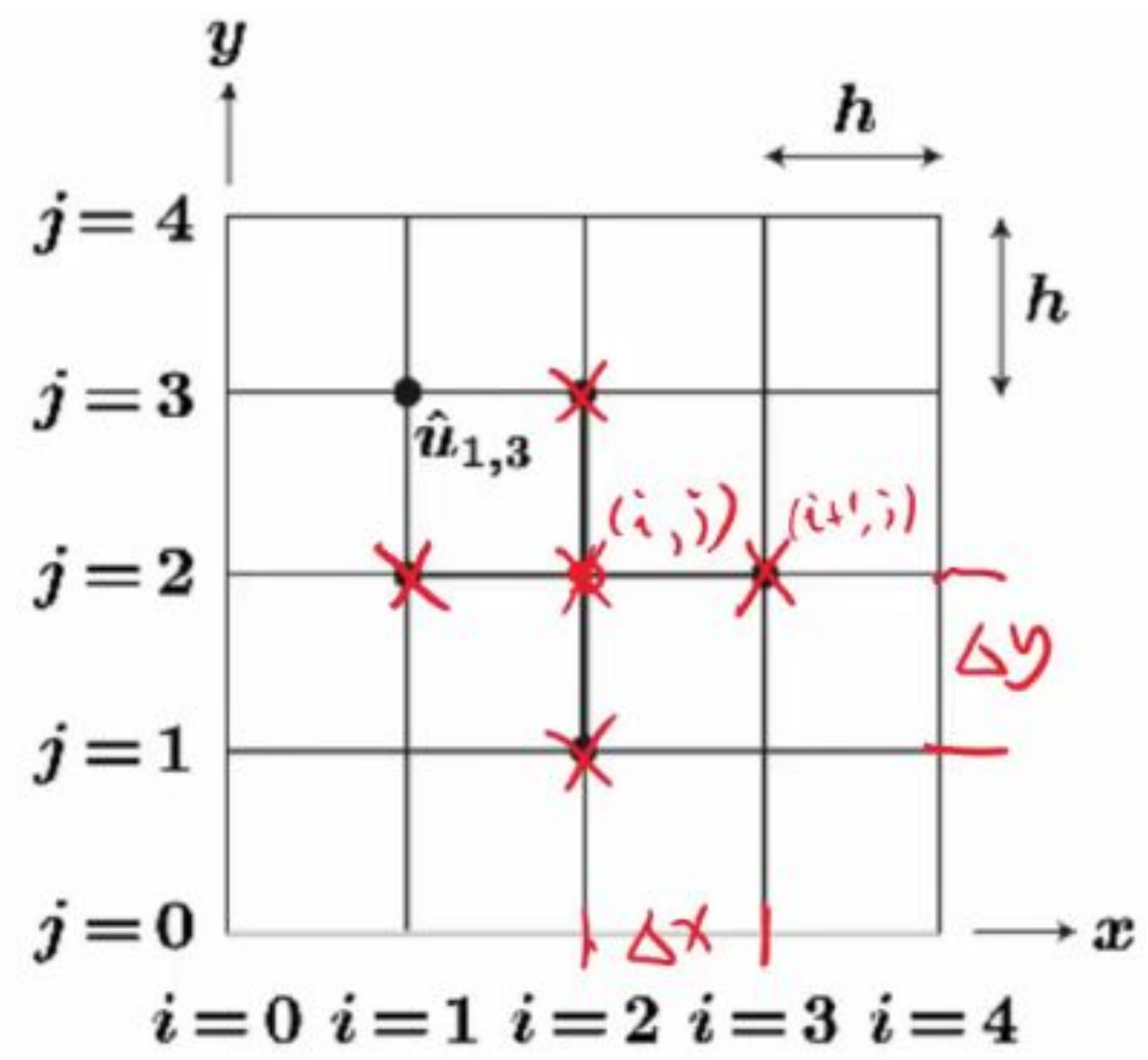


- Distance between pixels in *i*, j direction:
  - **hi** = 1/(N-1)
  - **hj** = 1/(M-1)

Computation

- We use the **implicit** formulation of the Gradient Descent (GD).
- We use the Euler-Laplace equation to solve it.
- We have a system of equations **Au** = **b**
  - **A** : sparse matrix of shape (*P*,*P*),

    where *P* = N · M, and (*N*,*M*) the image shape
  - **u**: "answer" image that minimizes the energy
  - **b**: output vector (contains 0's & values of original image **I**(j,i))



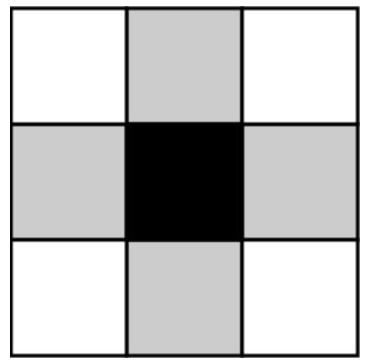
$$2D \quad \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f$$

$$\approx \boxed{\frac{u_{i+1,j} + u_{i-1,j} - 2u_{i,j}}{\Delta x^2} + \frac{u_{i,j+1} + u_{i,j-1} - 2u_{i,j}}{\Delta y^2}} = f_{i,j}$$

Setup and parameters

- The output image (**u**) should be equal to the original (**f**),
  - outside the region to be inpainted
  - on the external boundaries of the region to be inpainted
- In the region to be inpainted,
  - we use the Laplacian to approximate **J(u)**

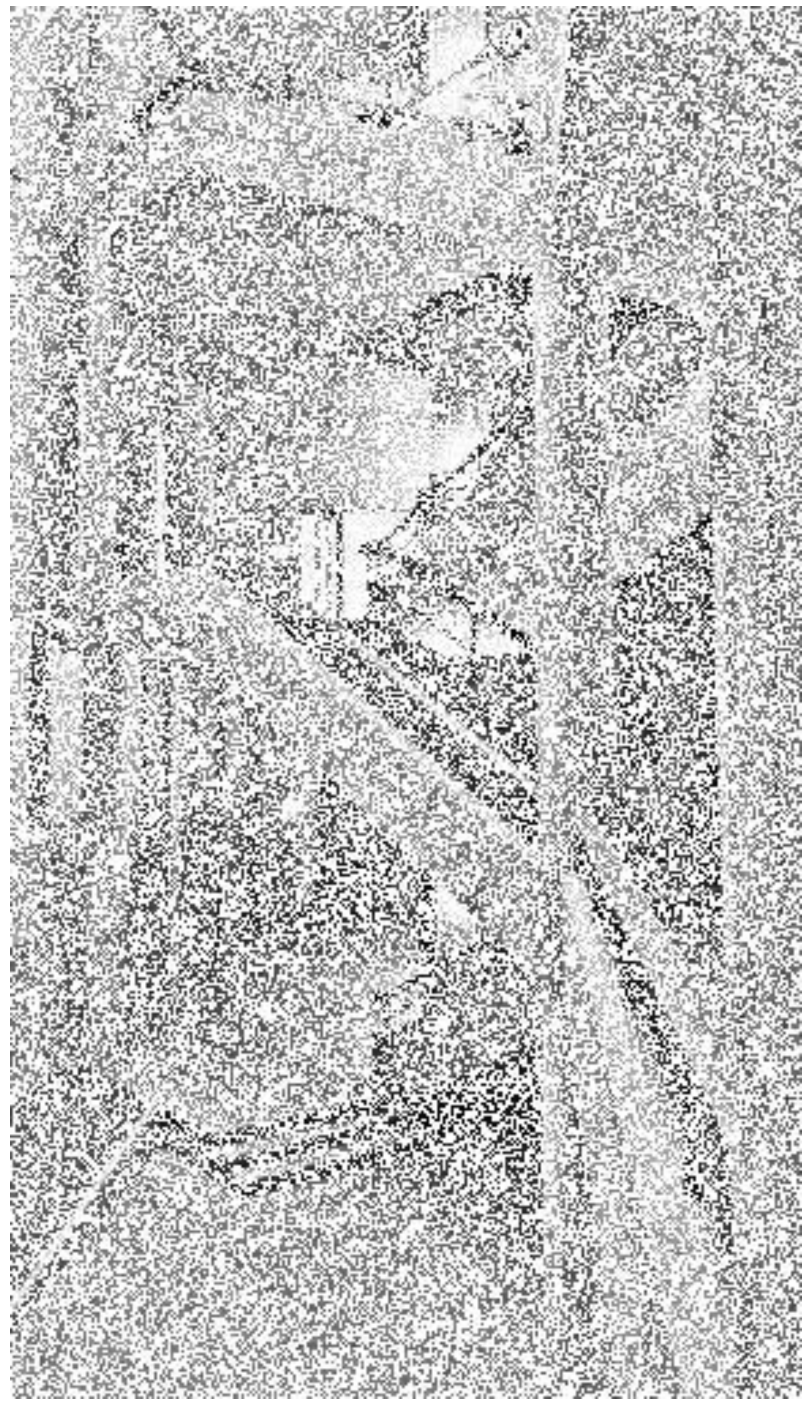
    with 5 values (4-connectivity)



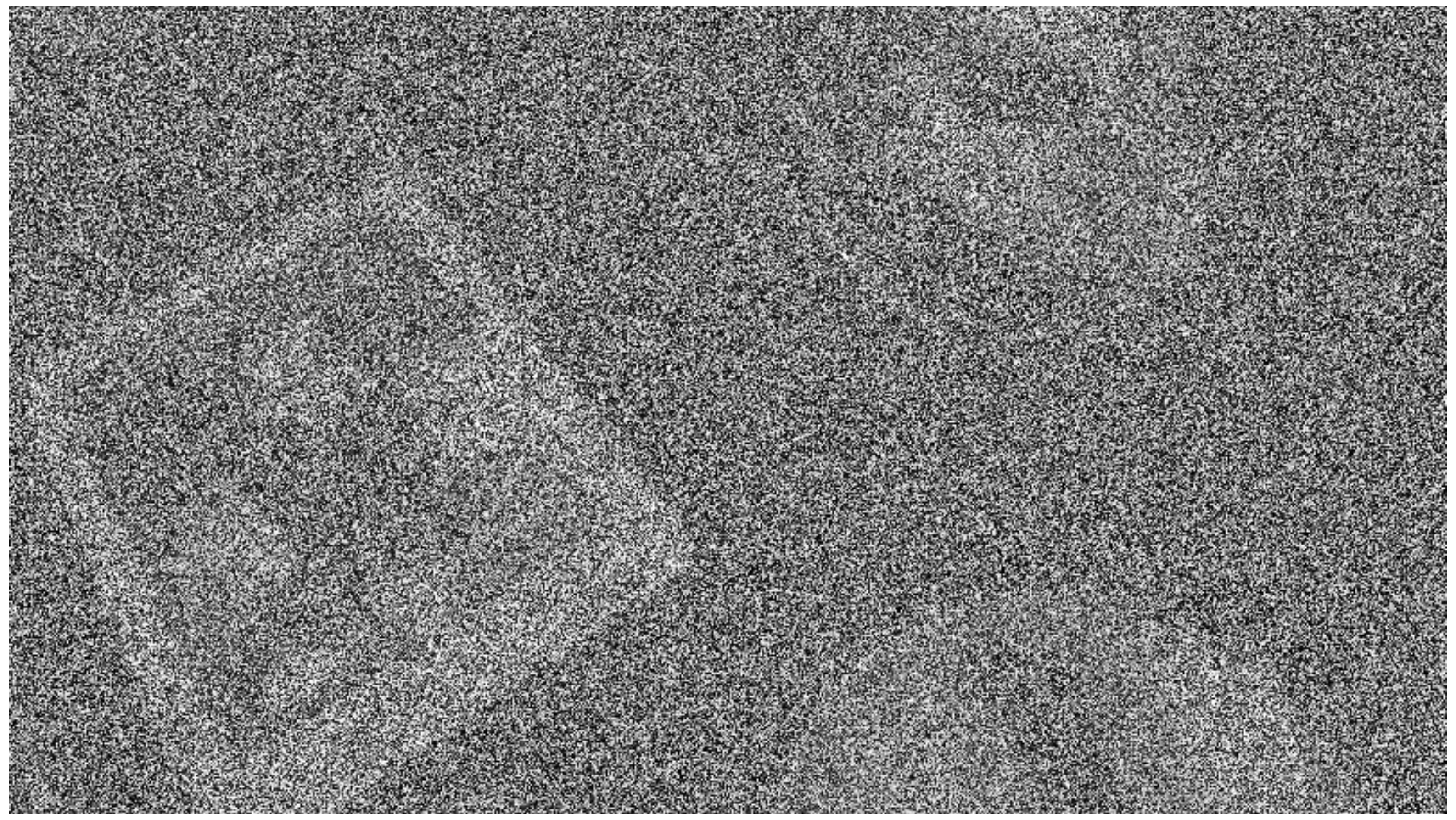
- Distance between pixels in *i*, j direction:
  - **hi** = 1/(N-1)
  - **hj** = 1/(M-1)

Results - Image to restore #1-3

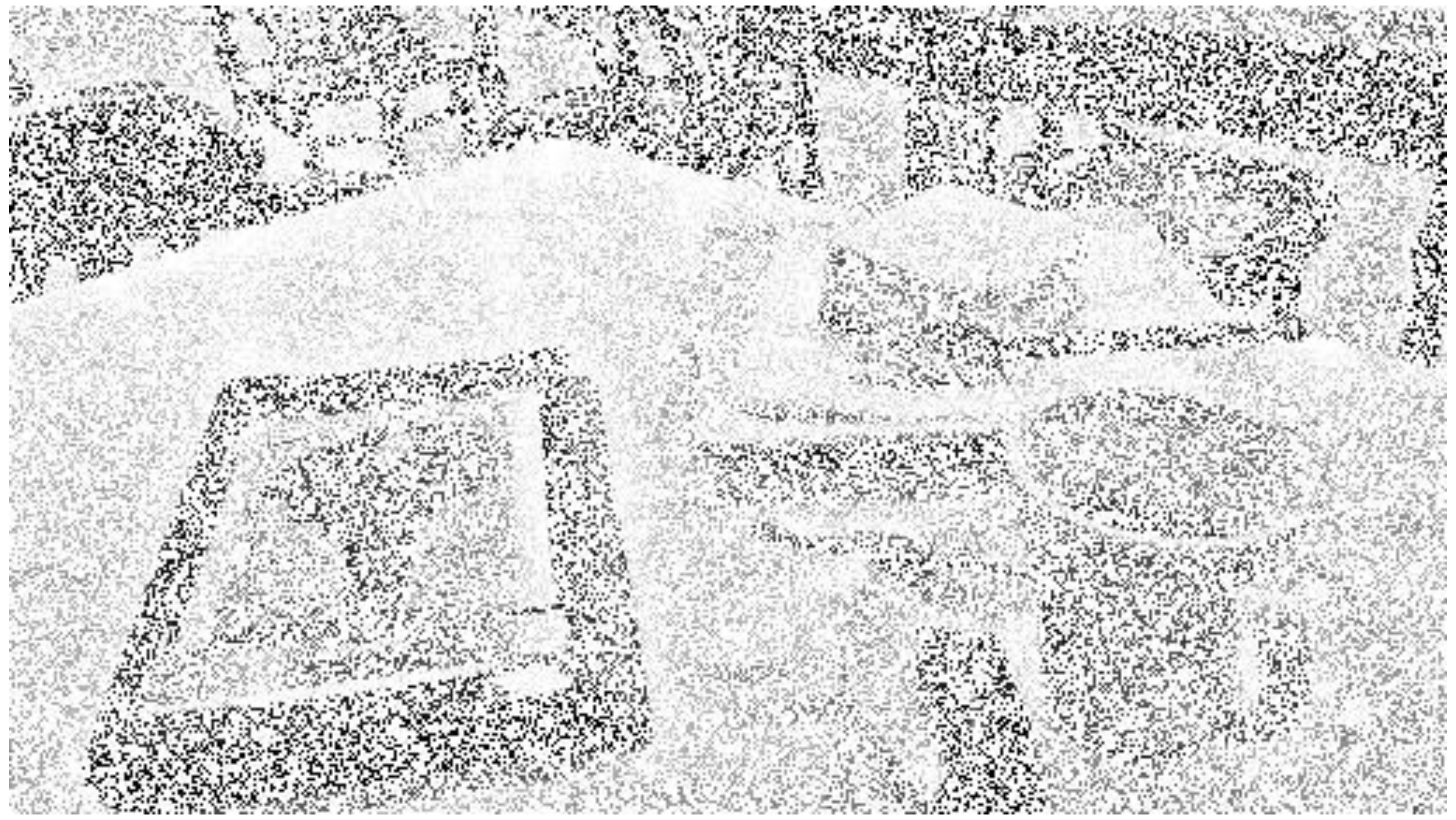
IMG1



IMG2

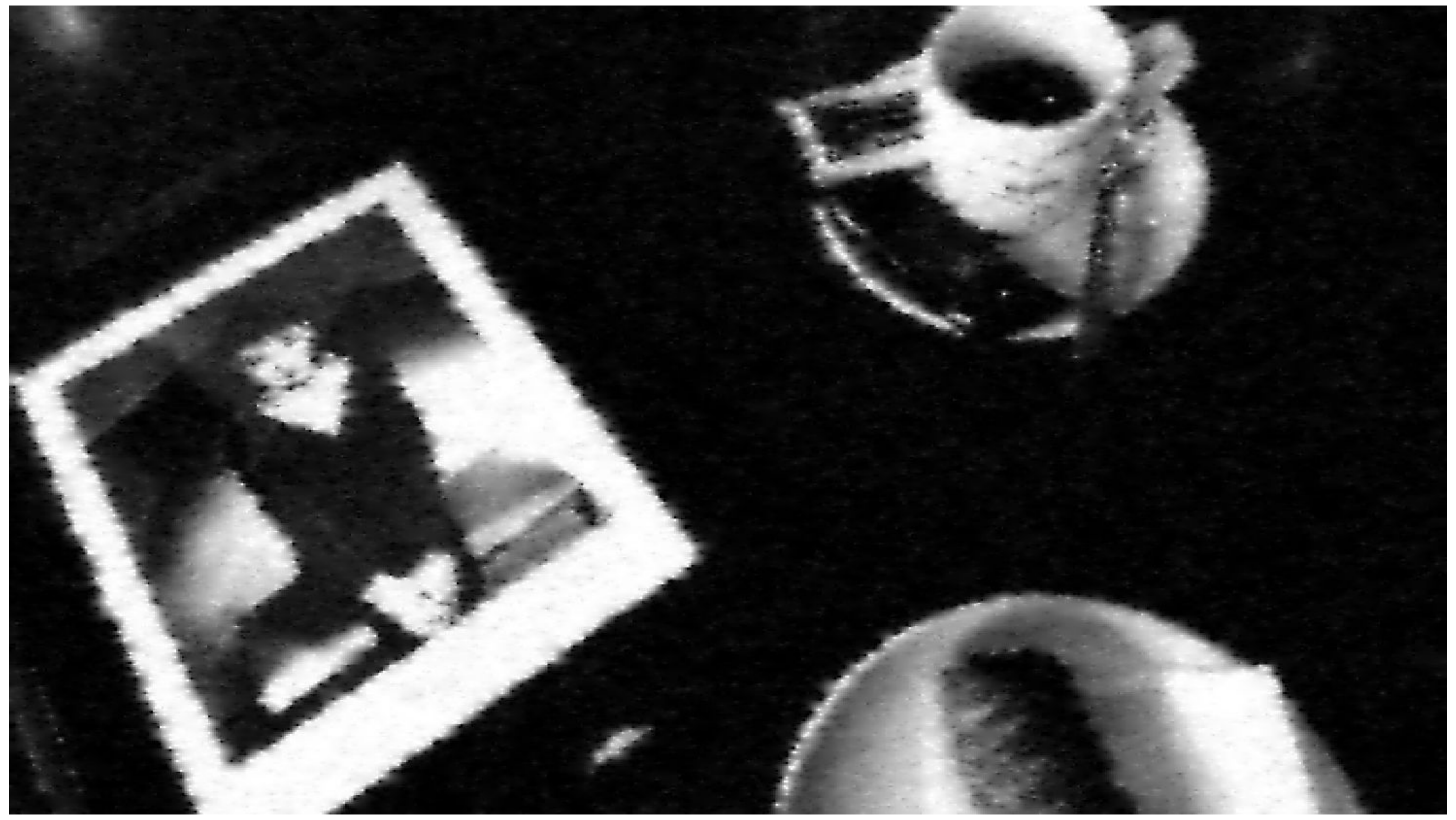


IMG3



IMG1-RESTORED

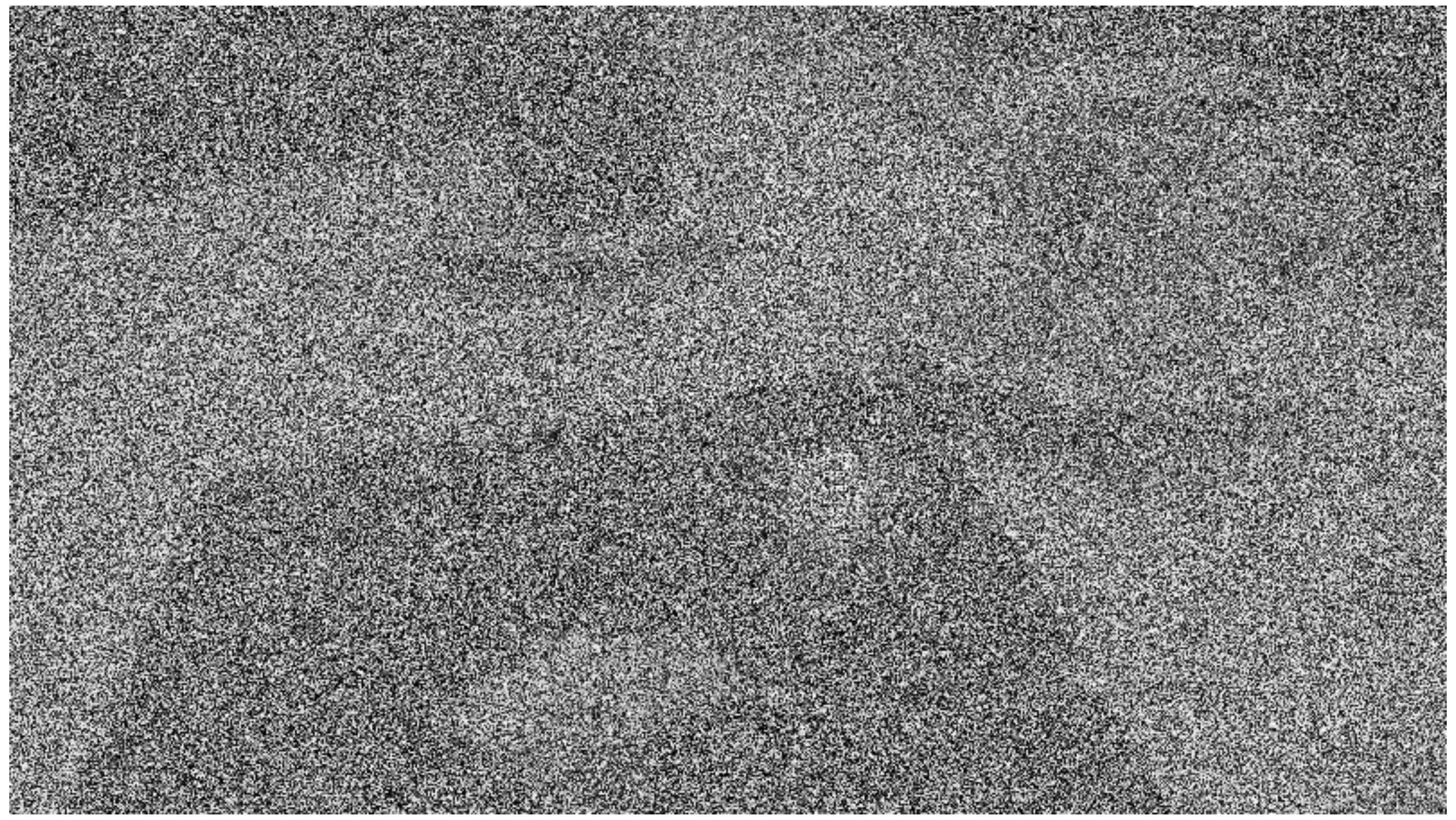
IMG2-RESTORED



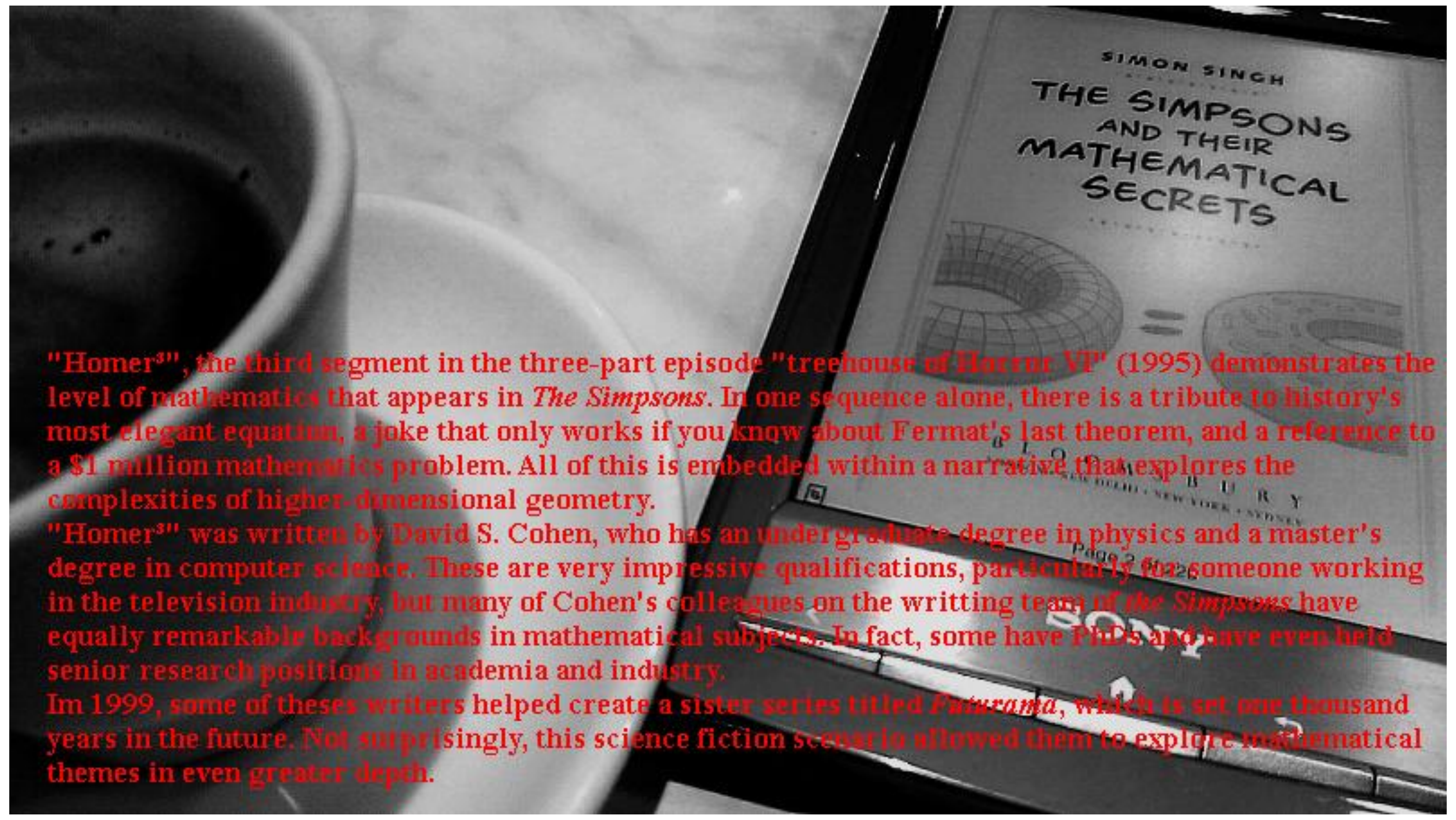
IMG3-RESTORED

Results - Image to restore #4-6

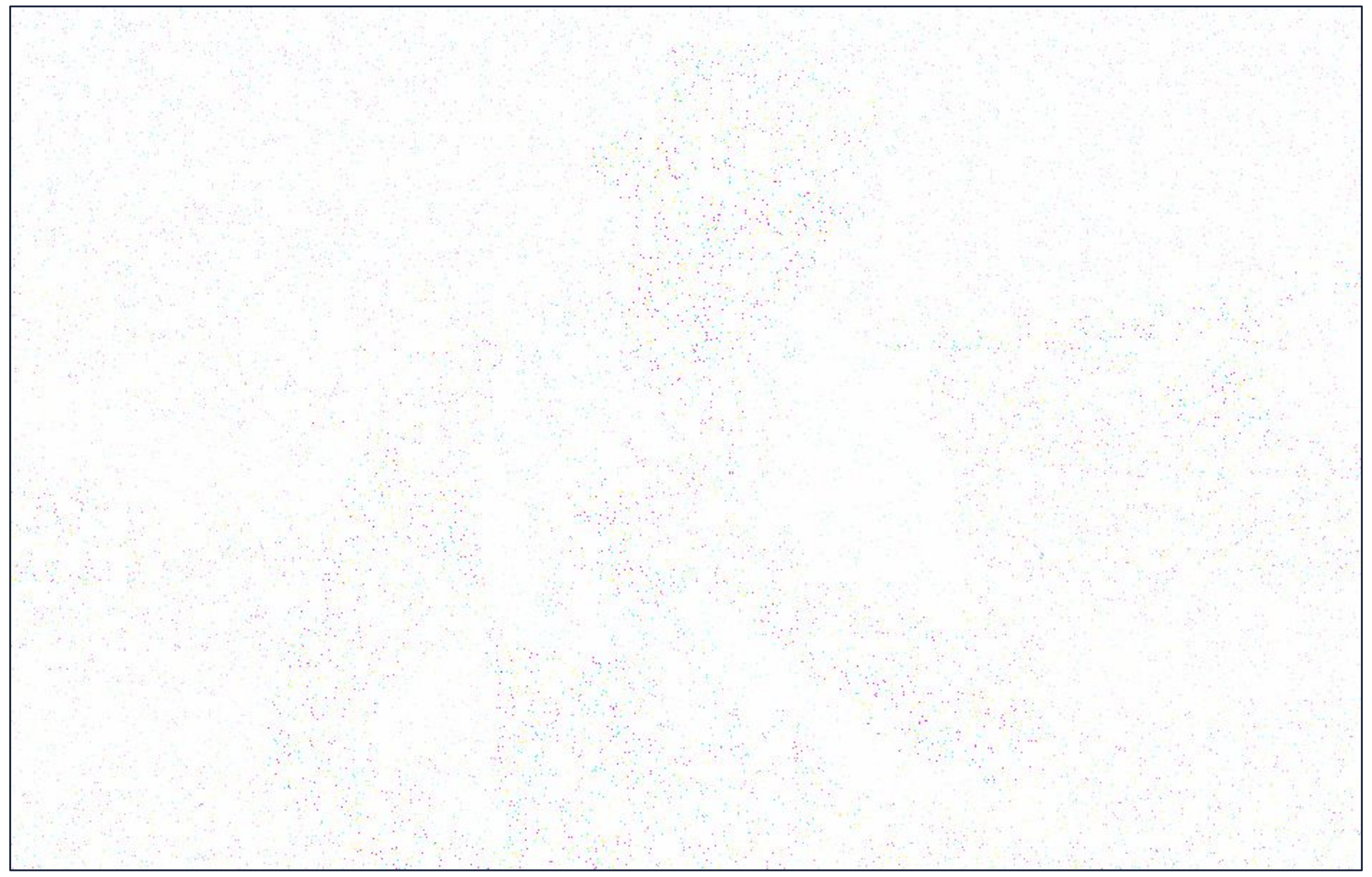
IMG4



IMG5

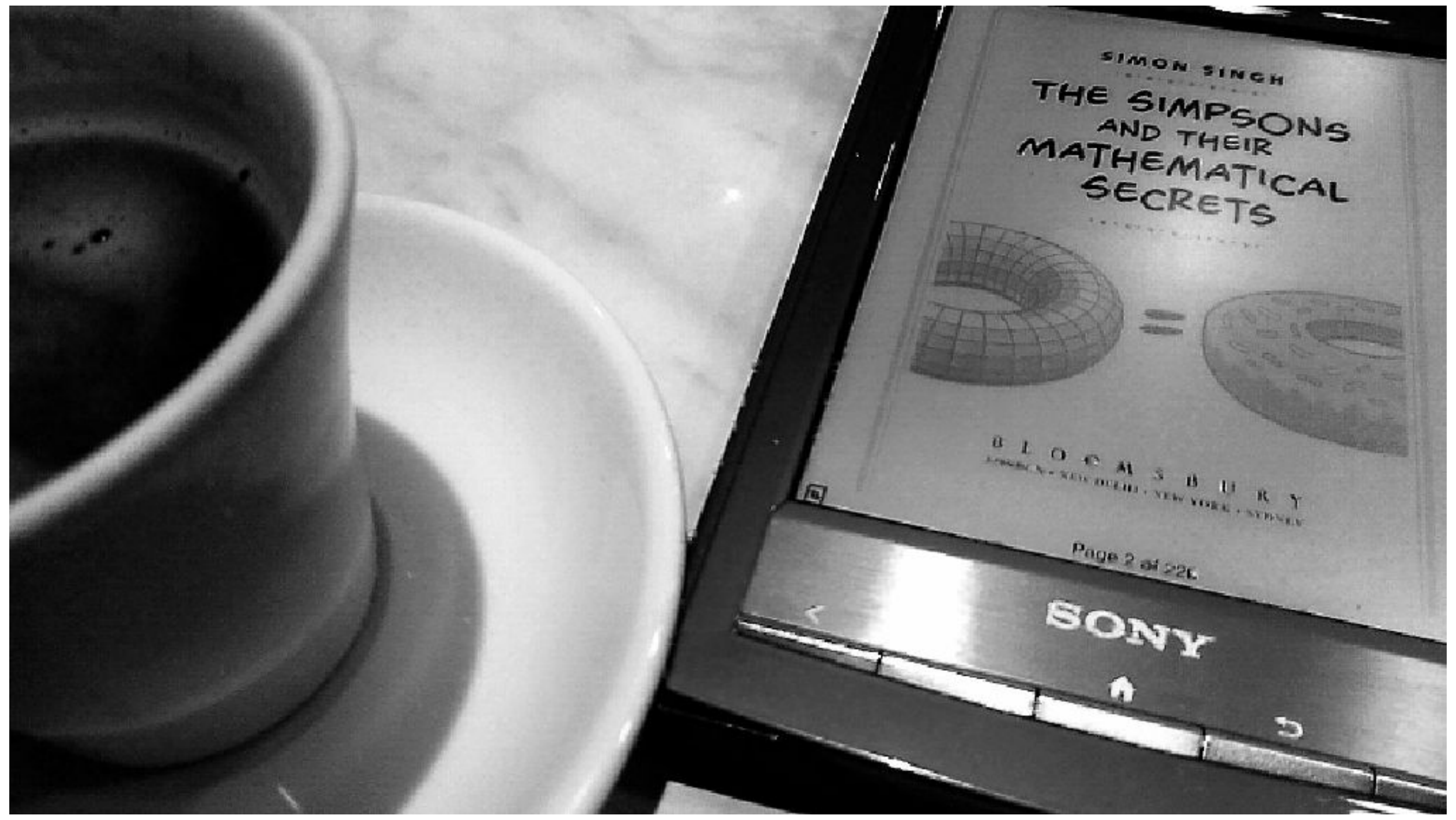


IMG6



IMG4-RESTORED

IMG5-RESTORED



IMG6-RESTORED

Results - Final image to restore

Image

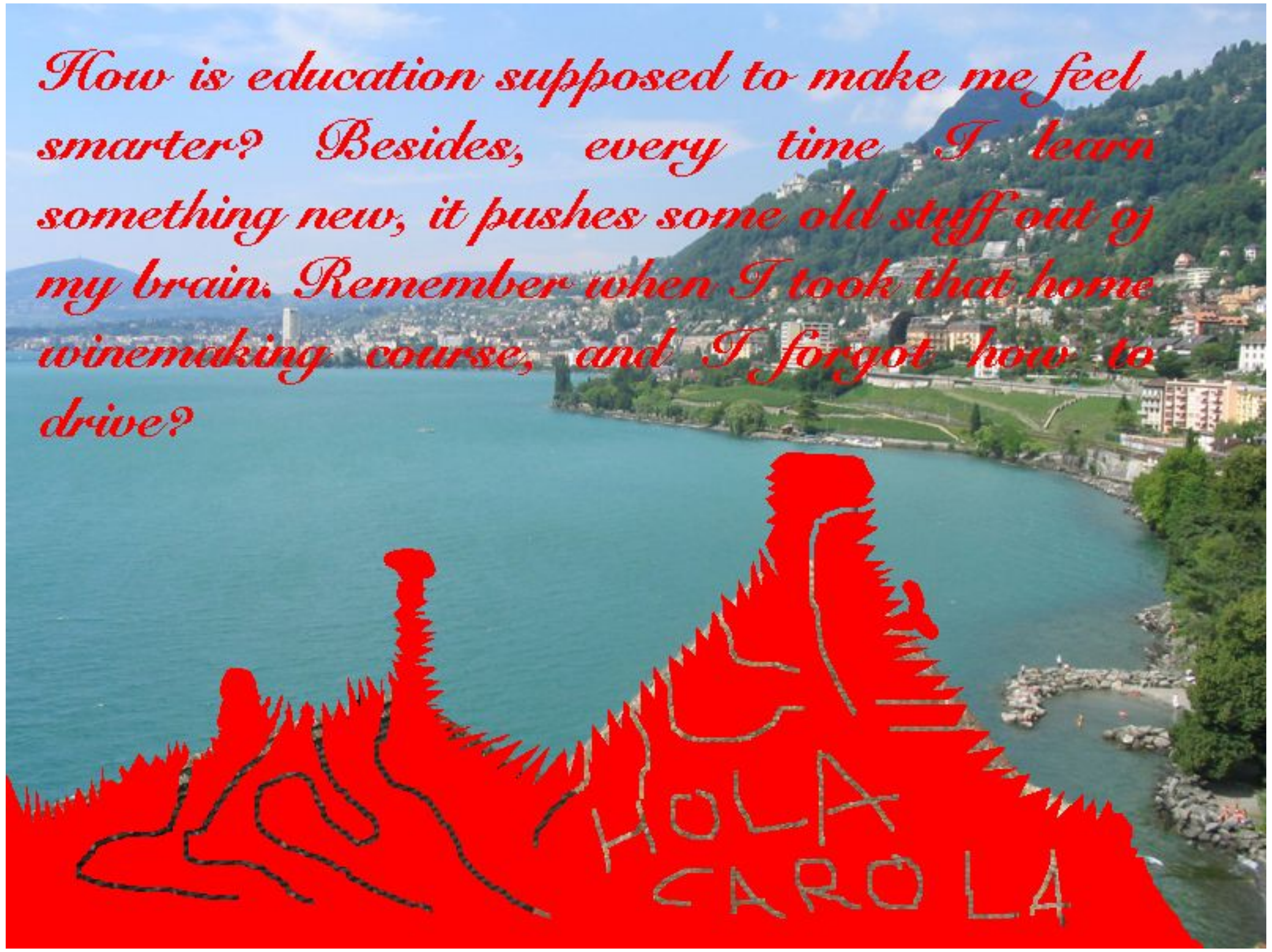


Image restored

Conclusions

- Text-over-background can be removed without noticeable quality loss
- Blur effect due to the effect of the smoothness parameter λ
  - Spatial high frequency cannot be recovered
    - Contours, corners, texture, ... are not recovered
  - More unpleasant with colour images
- Some results are impressive, even with only 1% of original pixels we can recognise objects in the image

Improvements

- Tune λ for each case
- Use an 8-connectivity to get more context (diagonal gradient is not considered on 4-conn)
- Apply some techniques to guess some lost contours beforehand. For example *Active Contour Models*
- Is inpainting each RGB channel separately and then combining them, the best option?
  - Channel-specific inpainting
  - Combine them in a different manner

- Proof that the solution of the Laplace's equation is a solution of the problem

$$\begin{cases} \underset{u \in W^{1,2}(\Omega)}{\arg\min} \int_D |\nabla u(x)|^2 dx, \\ u|_{\partial D} = f \end{cases} \longrightarrow \nabla F(\mathbf{u}) = 0$$

$$u\Big|_{\Omega \setminus D} = f$$

Let $\frac{1}{2}\int_D |\nabla u(x)|^2 dx = F(\vec{u})$

then $argmin_u F(\vec{u})$

then to minimize $\frac{\delta}{\delta u} F(\vec{u}) = \vec{0}$

that by definition is $\nabla F(\vec{u}) = \vec{0}$ ∎

(1) Applying *finite backward difference* to approximate Laplacian

$$\Delta_u \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_i{}^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_j{}^2}$$

$$u(x,y) \approx u_{i,j}$$

DEFINITIONS

(**) we apply Neumann boundary conditions (i.e.: mirroring)

- Necessary condition for the extremum (minimum in our case)

$$\frac{\partial J}{\partial u} = -\sum_{i=1}^{D} \frac{\partial}{\partial X_i} \frac{\partial F}{\partial u_{x_i}} + \frac{\partial F}{\partial u} = 0$$

where D : dimension of our space. D=2, in our case.

- Our functional has no *f* terms so, the second term is already 0

The first term can be computed in parts as:

$$\frac{\partial F}{\partial u_x} = \frac{1}{2} 2u_x = u_x \; ; \; \frac{\partial F}{\partial u_y} = \frac{1}{2} 2u_y = u_y$$

$$\frac{\partial J}{\partial u} = -\left(\frac{\partial}{\partial y} u_y + \frac{\partial}{\partial x} u_x\right) = -\Delta_u =^{(1)}$$

$$\frac{-u_{i+1,j} + 2u_{i,j} - u_{i-1,j}}{h_i{}^2} + \frac{-u_{i,j+1} + 2u_{i,j} - u_{i,j-1}}{h_j{}^2}\Bigg|_{h_i = h_j = 1} =$$

$$\boxed{4u_{i,j} - u_{i+1,j} - u_{i-1,j} + -u_{i,j+1} - u_{i,j-1} = 0} \quad \forall i,j$$

We can build a sparse matrix of coefficients **A** to define, with the other criteria, a system of equations like: $\boxed{\mathbf{A}\mathrm{u} = \mathbf{b}}$

METRICS (I)

Dataset	Training		Test		Time
Split	Split1	*Split2*	Split1	*Split2*	(s)
Method1					
Method2					
Method3					
Method4					
Method5					

METRICS (II)

Dataset	Training		Test		Time
Split	Split1	*Split2*	Split1	*Split2*	(s)
Method1					
Method2					
Method3					
Method4					
Method5					

Graph 1:

*

Thanks!

Any questions?

You can contact us at:

jonatan.poveda@e-campus.uab.cat
ferran.perezg@e-campus.uab.cat

REFERENCES

[1] Ref1

[2] Ref2

[3] Ref3

[4] Ref4

[5] Ref5

[6] Ref6

[7] Ref7

[8] Ref8

CREDITS

Special thanks to all people who made and share these awesome resources for free:

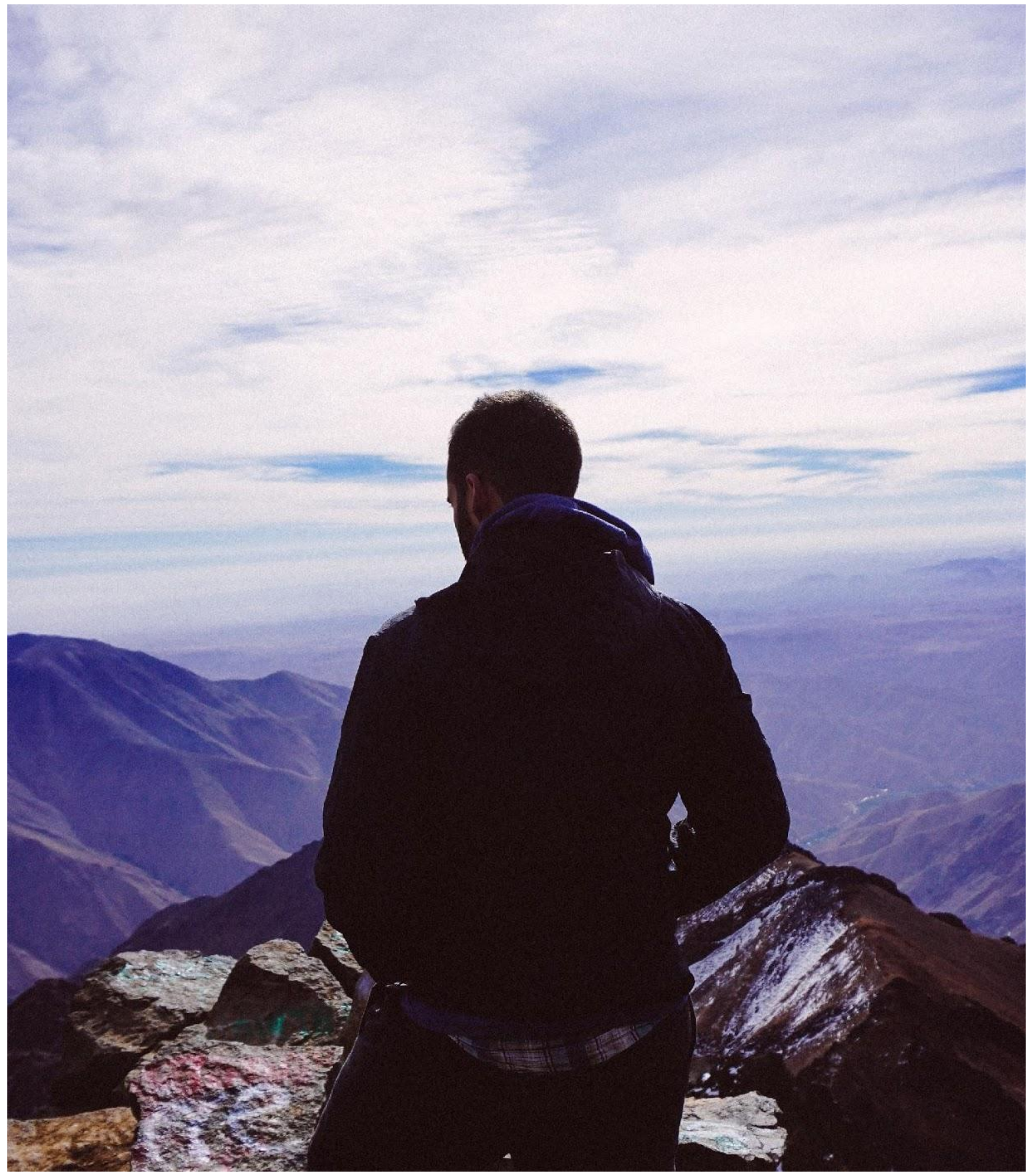
- Presentation template designed by Slidesmash
- Photographs by unsplash.com and pexels.com
- Vector Icons by Matthew Skiles

Hello!

I Am John Miller

I am here because I love to design presentations.

You can contact me at @username



A Picture Is Worth A Thousand Words

Itself is what the end-user derives value from also can refer is what the end-user derives value from also can refer to the information provided through the medium.


Use Big Images
To Show Ideas

USE SHAPE TO EXPLAIN IDEAS

TITLE
01

TITLE 02

TITLE 03

“A person who never made a mistake never tried anything new”

56,790,500

Write here your big numbers

FUNNY FACTS

56,790,500

Revenue from sales

130%

Project Achievements

56,790,500

Users around the world

OUR PROCESS

01

The marketing mix is a business tool used in marketing

02

The marketing mix is a business tool used in marketing

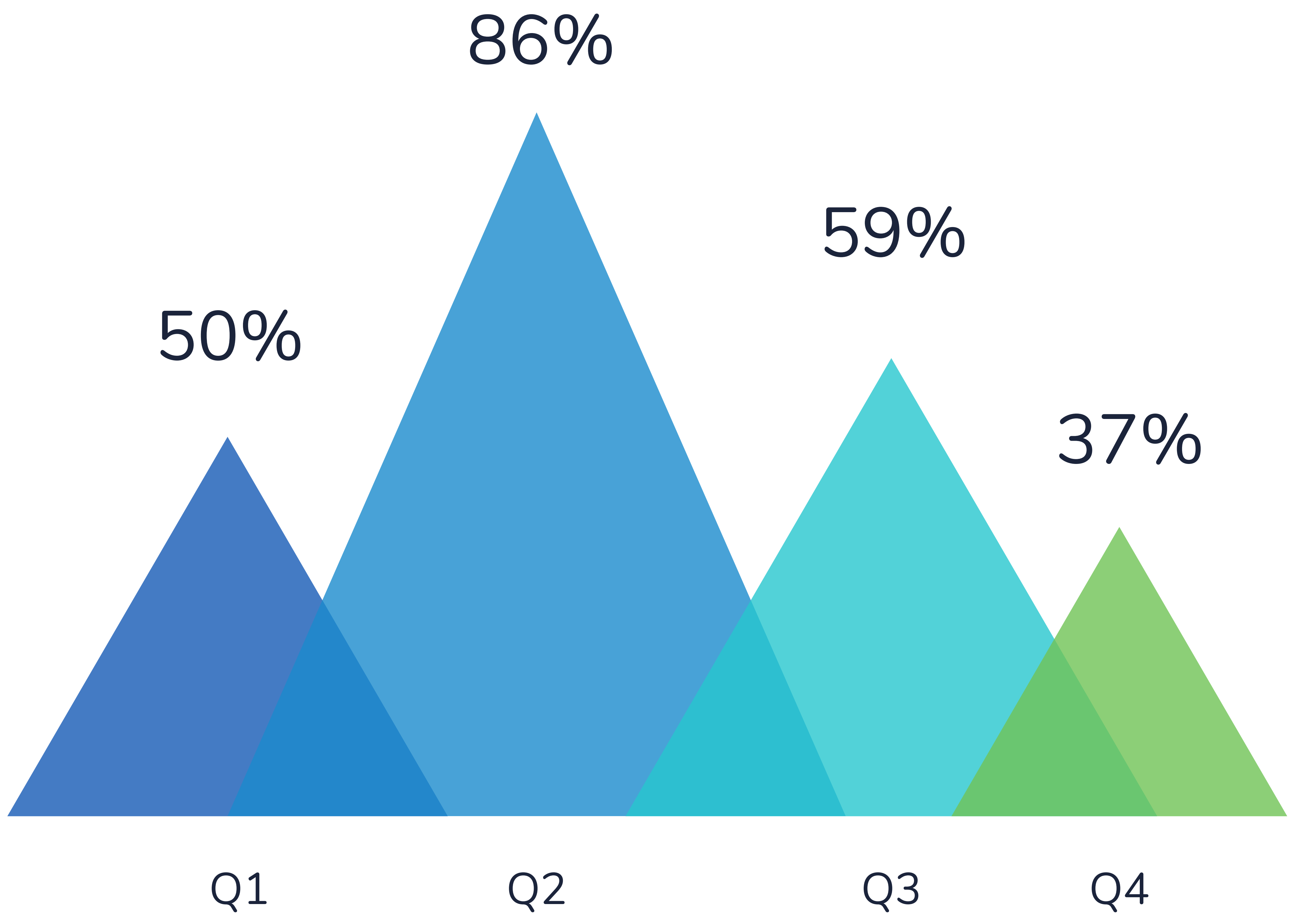
03

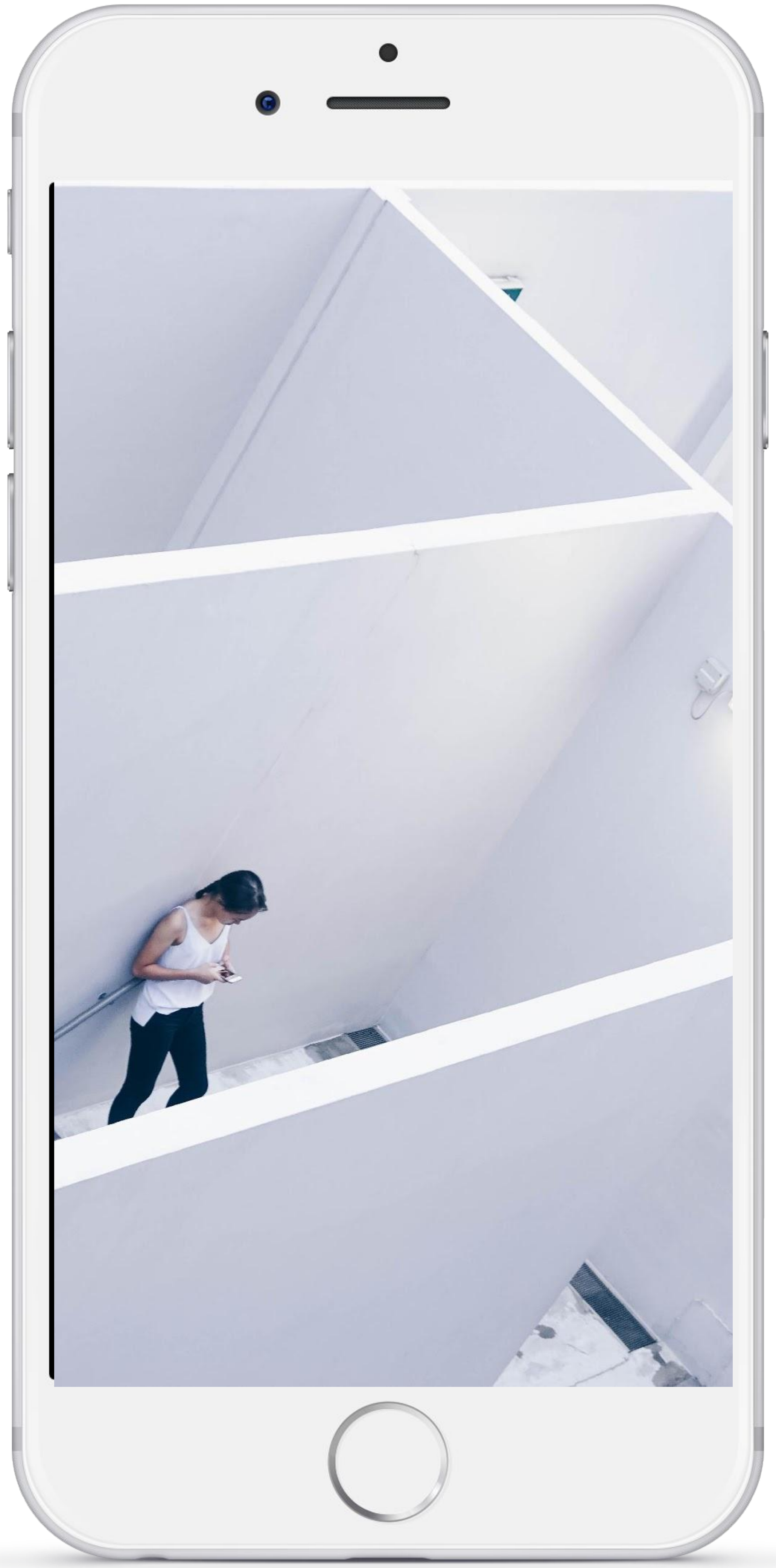
The marketing mix is a business tool used in marketing

04

The marketing mix is a business tool used in marketing

CHARTS TO PRESENT DATA





iPhone App Project

Itself is what the end-user derives value from

also can refer to the information

Laptop Project

Itself is what the end-user derives value from also can refer to the information

Content B

Itself is what the end-user derives value from also can refer to the information

Content D

Itself is what the end-user derives value from also can refer to the information

Presentation Design

This presentation uses the following typographies and colors:

Free Fonts used:

https://www.fontsquirrel.com/fonts/nunito

Colors used