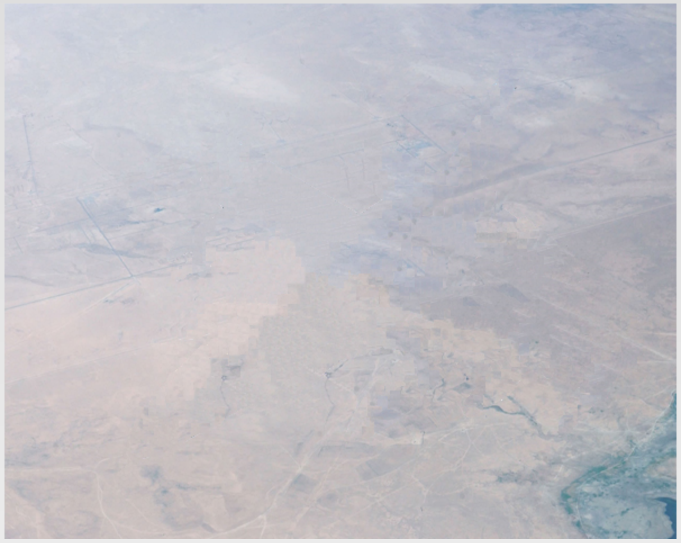


GPU Accelerated Image Inpainter

Image Inpainting

- Filling in a defined region of the image in a visually pleasing way.
- Done by artists for centuries.
- Recent research into doing it from a computational perspective.
- Many algorithms developed.
- Adobe Photoshop CS5 “Content-Aware Filling”

Uses

- Fill in and restore damaged parts of an image.
- Noise removal.
- Removal of unwanted objects from a scene, e.g. rubbish, lens flares, political opponents.
- Fill in black borders caused by “panograph stitching”

Goal

- To research into different algorithms available for inpainting, and understand the algorithm.
- From understanding the algorithm, understand the fundamental ideas of image inpainting
- Develop a versatile relatively user-friendly image inpainting tool, rather than just a demo.

Technical

- C++
- Win32 GUI
- OpenCV
- CUDA
- Windows x86/x64, 512MB RAM (1024 for larger images)
- Requires CUDA-enabled GPU for GPU acceleration features – Nvidia Geforce 8600 or newer

Scope

- 7 supported image inpainting algorithms
- 4 colourspaces
- BMP, JPEG, PNG

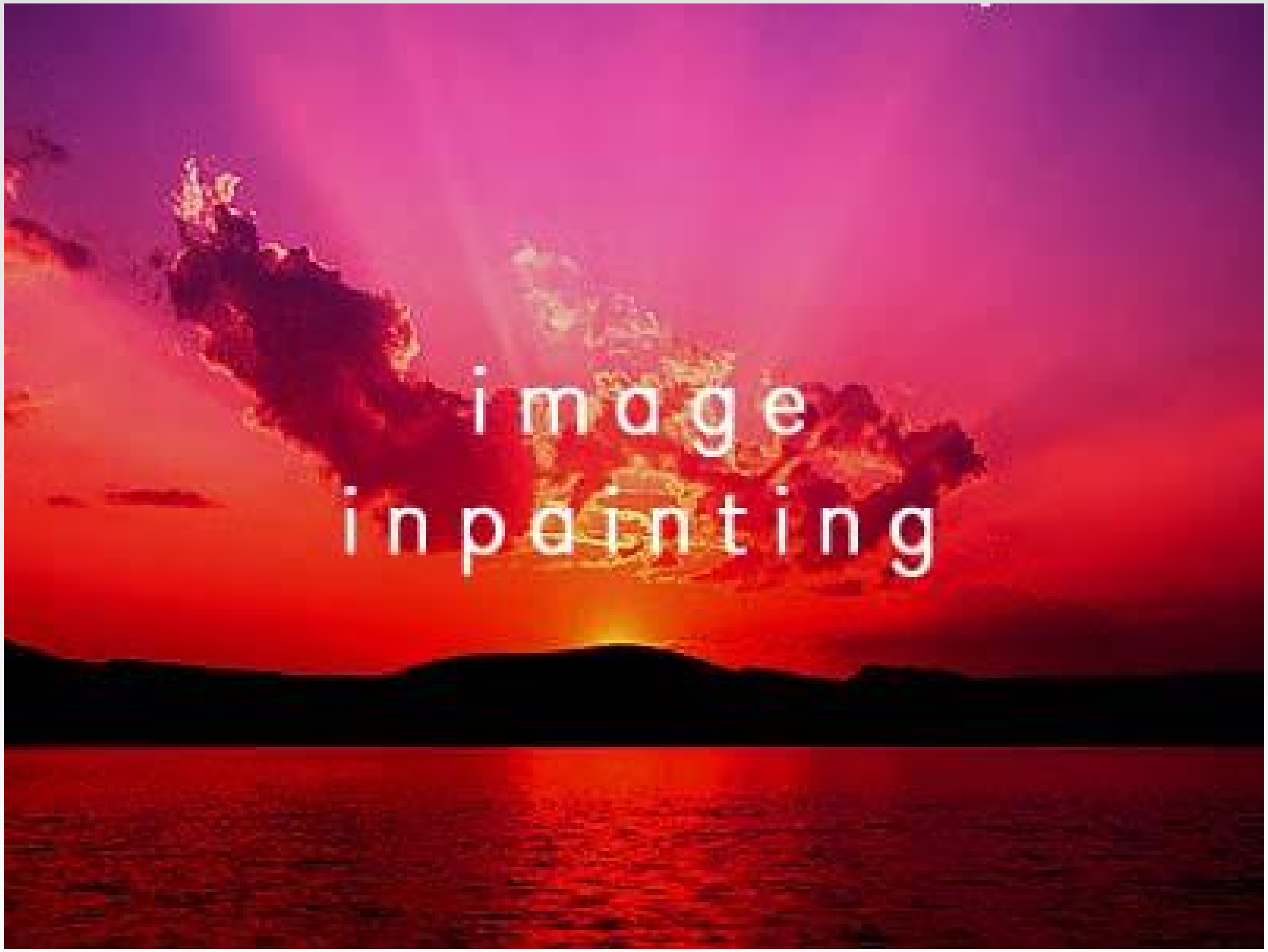
PDE Methods

- Try to solve a partial differential equation, such that the isophotes of the image continue seamlessly into the region.
- Simplest form: Gaussian blur
- Modern methods are very fast at doing this
- Produces inevitable “blurring” of results
- Texture is not extended into the filling region.
- Good for images with thin fill region radius, lots of structure info, but not a lot of texture

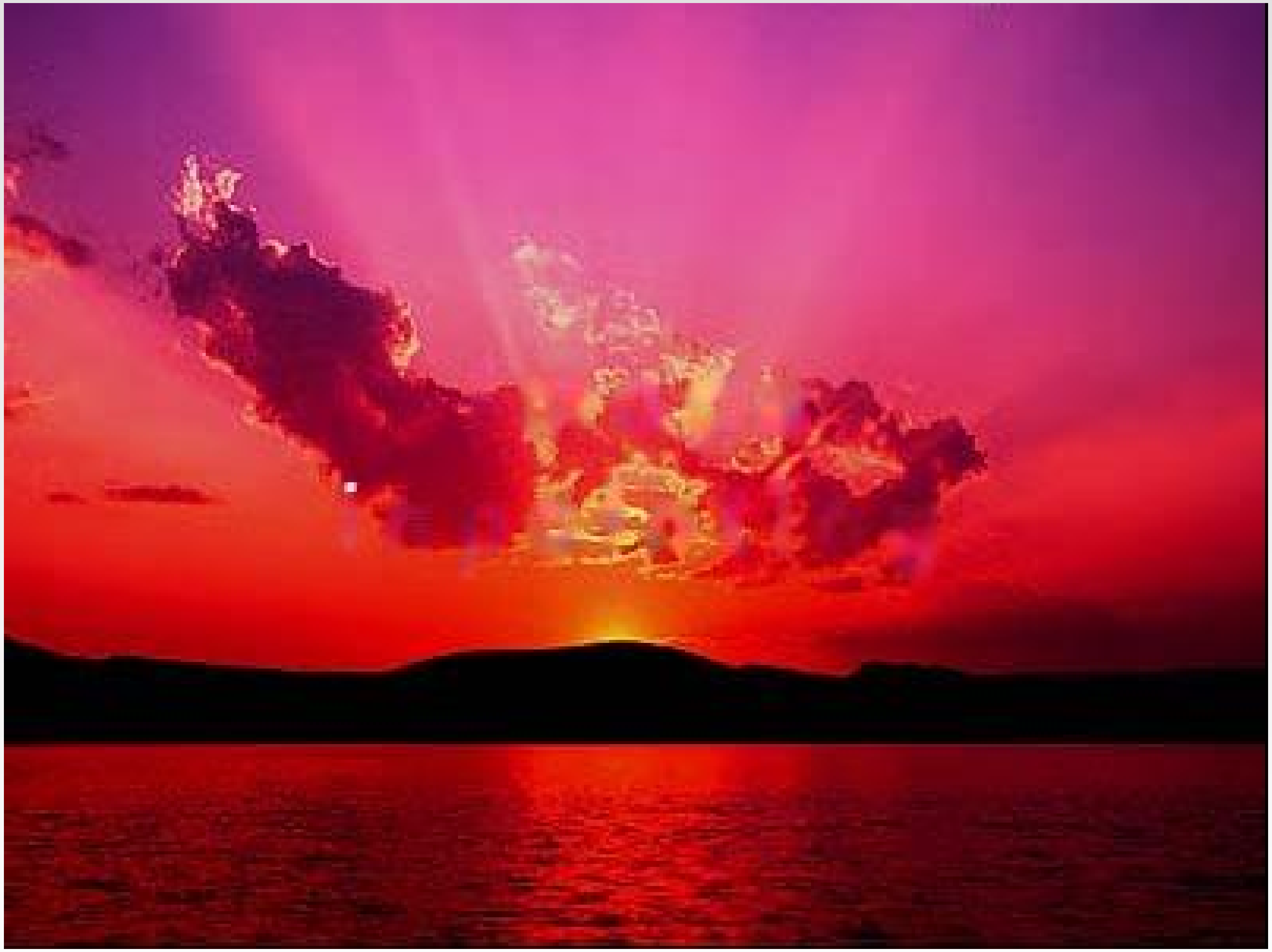
PDE Methods We’ve Implemented

- Bertalmio SIGGRAPH 2000 (extremely slow, numerically unstable, best results)
- Fast Digital Image Inpainting (Repeated Gaussian blur kernel applied to image)
- Progressive Octave Gaussian (custom algorithm)
- Image Inpainting using Navier Stokes Fluid Dynamics (OpenCV)
- Fast Marching Method (OpenCV)

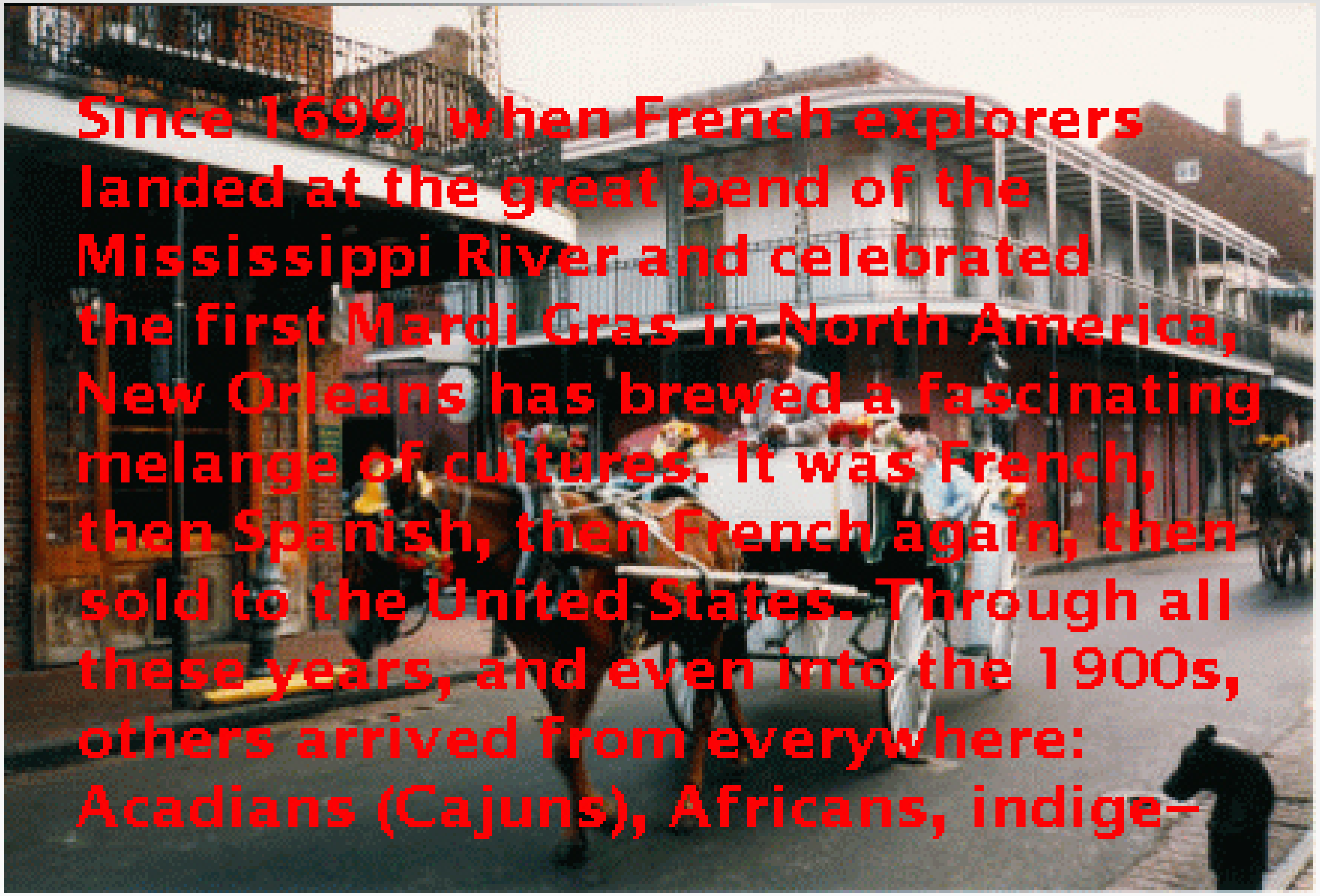
PDE Method Results



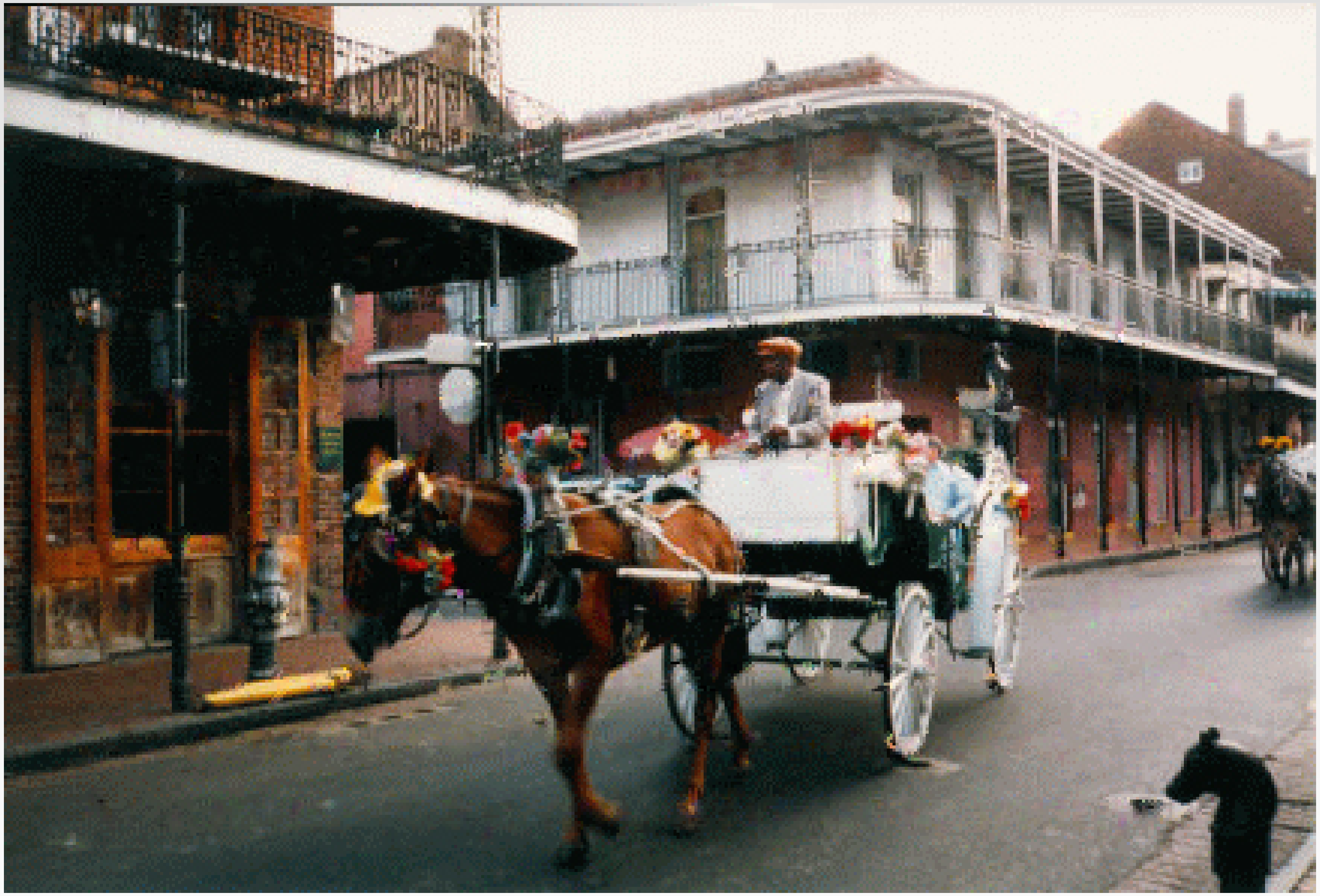
PDE Method Results: Bertalmio



PDE Method Results

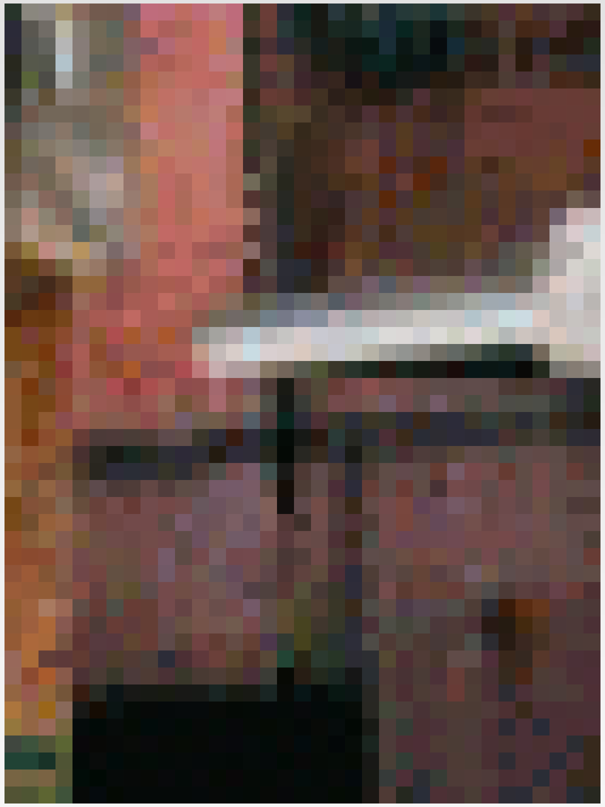

Since 1699, when French explorers
landed at the great bend of the
Mississippi River and celebrated
the first Mardi Gras in North America,
New Orleans has brewed a fascinating
melange of cultures. It was French,
then Spanish, then French again, then
sold to the United States. Through all
these years, and even into the 1900s,
others arrived from everywhere:
Acadians (Cajuns), Africans, indige-

PDE Method Results: Bertalmio



Adobe Photoshop CS5

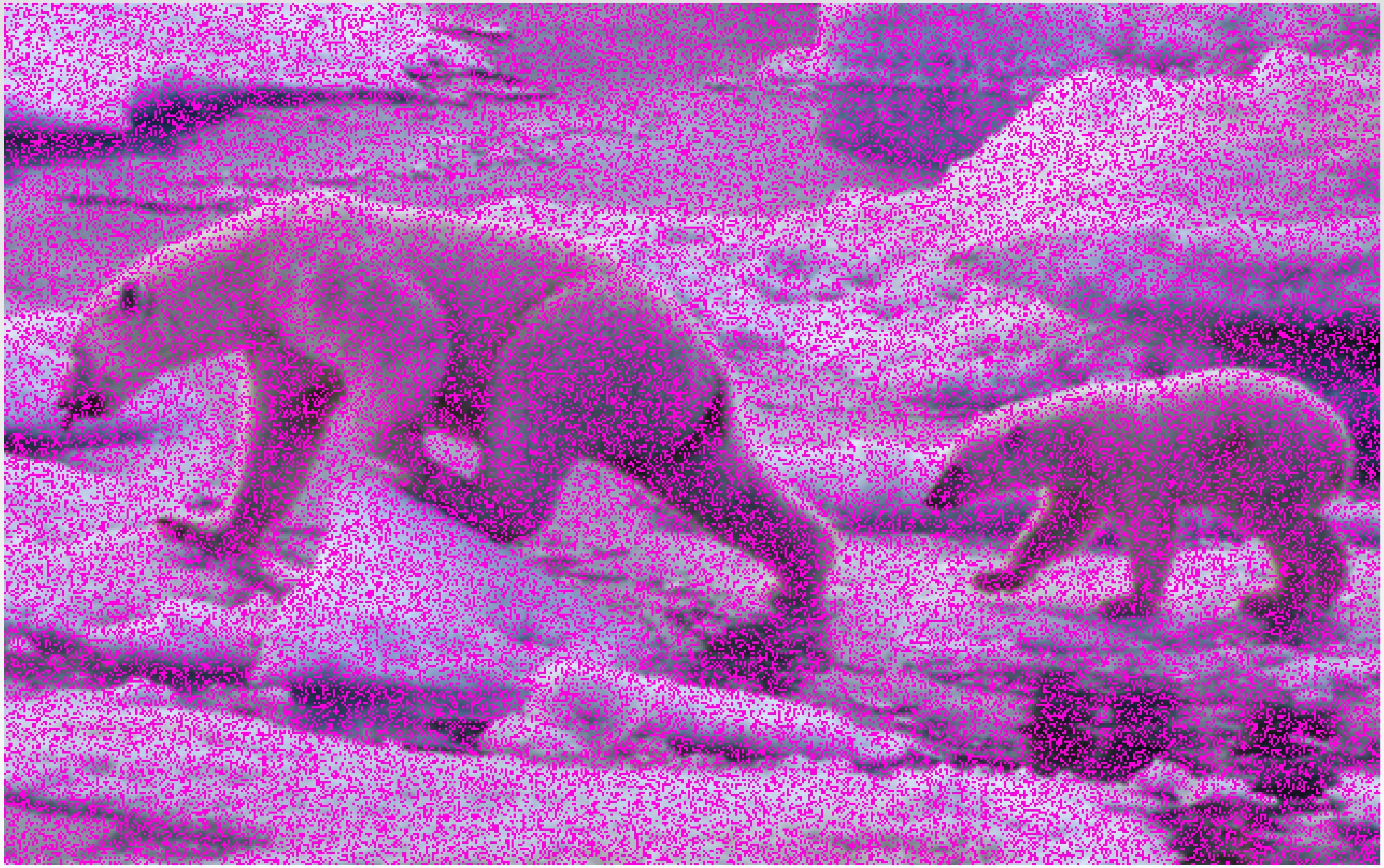
Closer Look



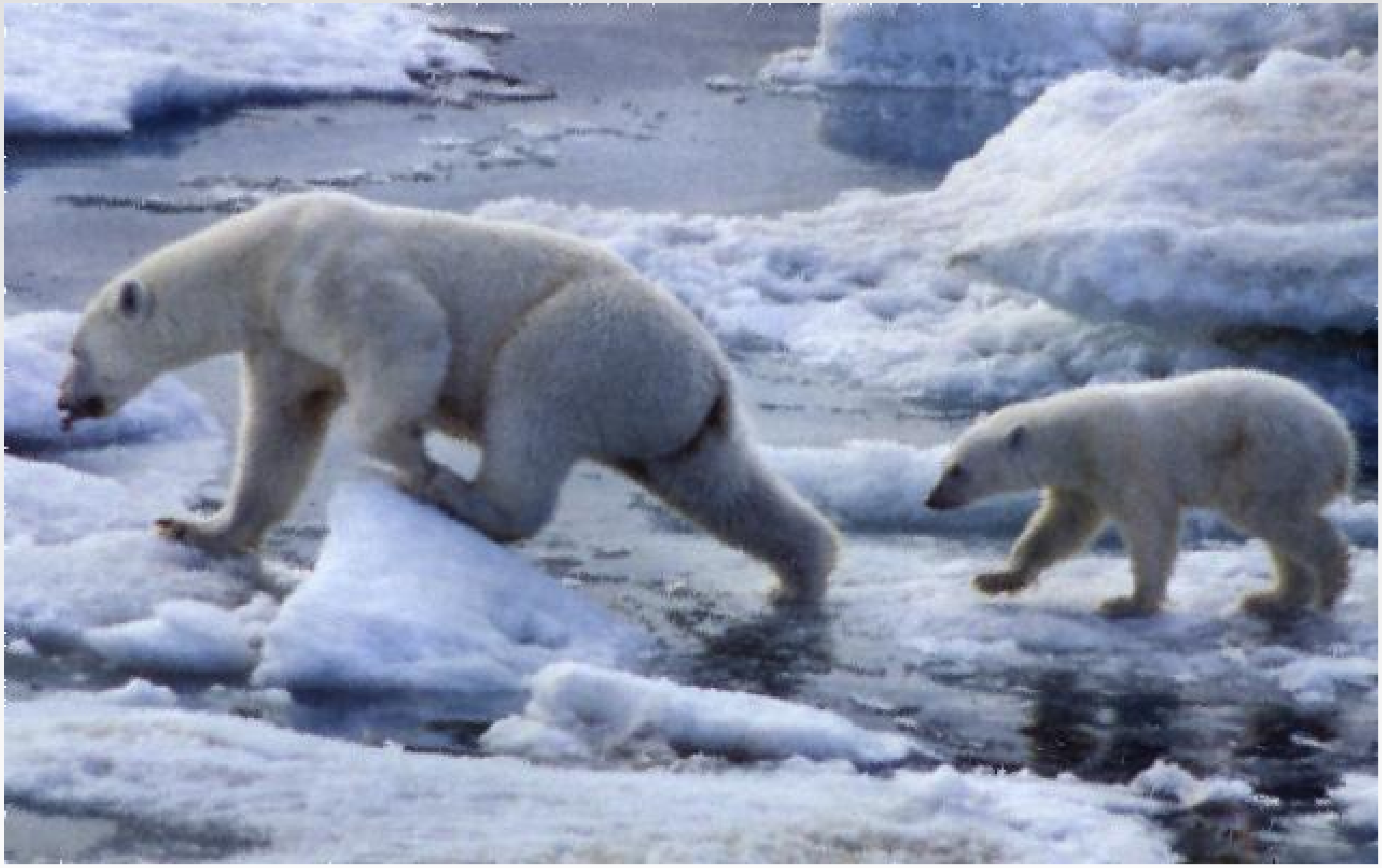
Closer Look

Closer Look

PDE Method Results: Bertalmio

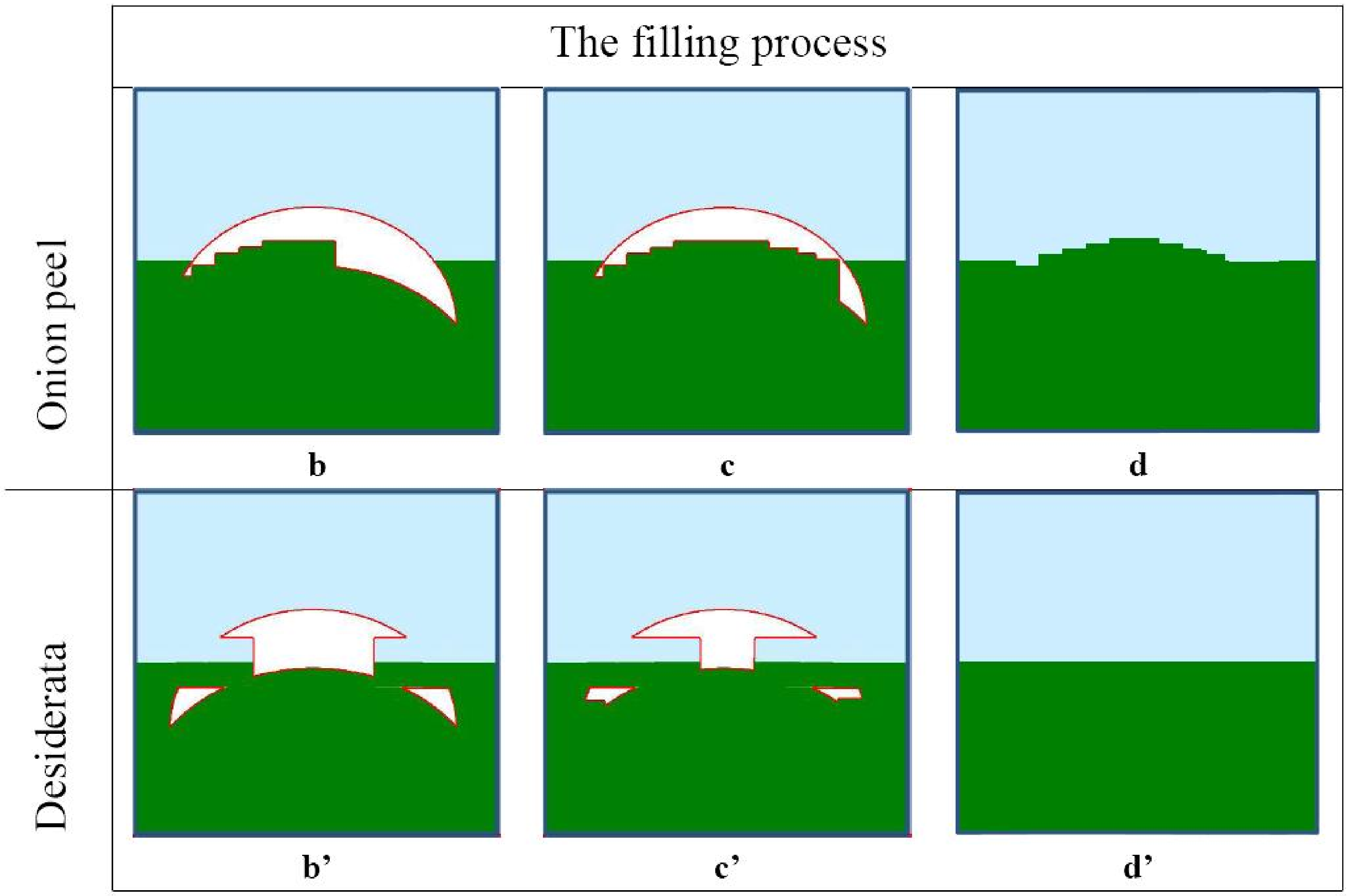


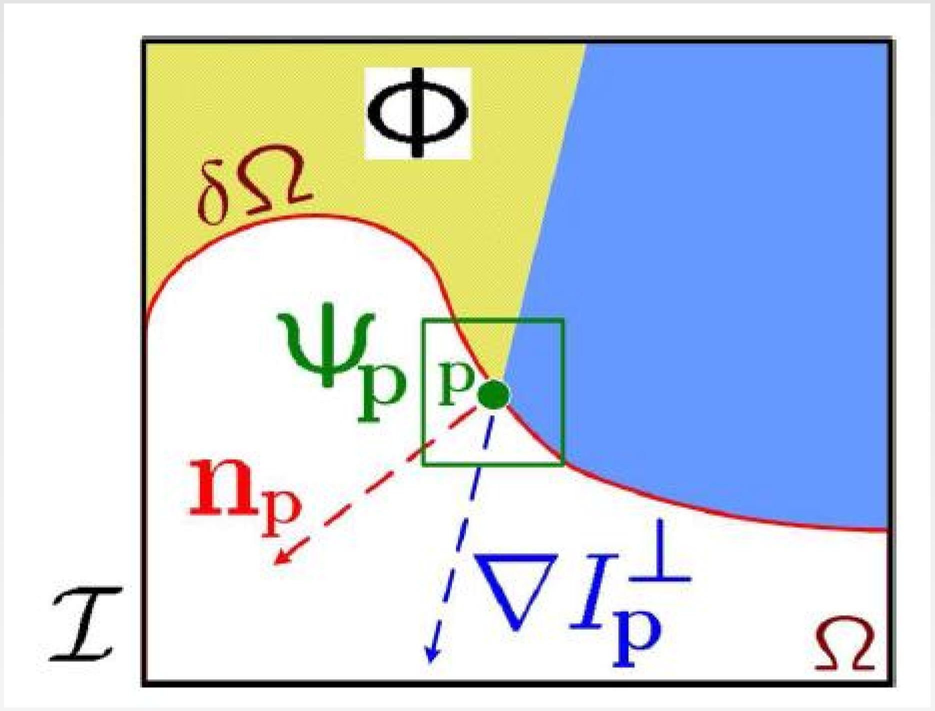
PDE Method Results: Bertalmio

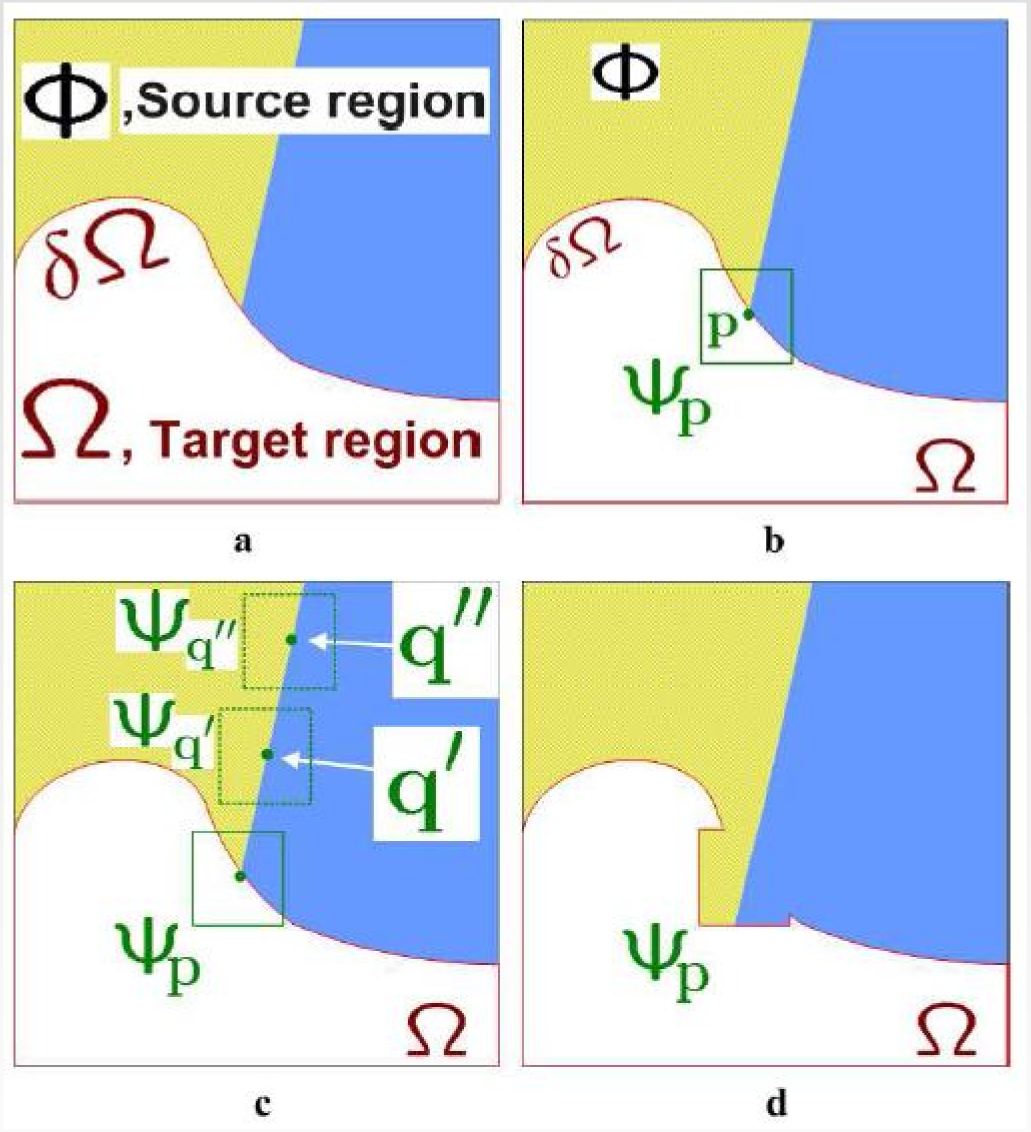


Texture Synthesis:

- Originally bad because it requires a lot of user interaction
- PDE Methods preserve structure; texture synthesis does not.
- Recent breakthroughs (2004) allow texture synthesis to be used for inpainting, while taking into account image structure.
- Exemplar-Based Image Inpainting
- Local Optimisation Method (2010)

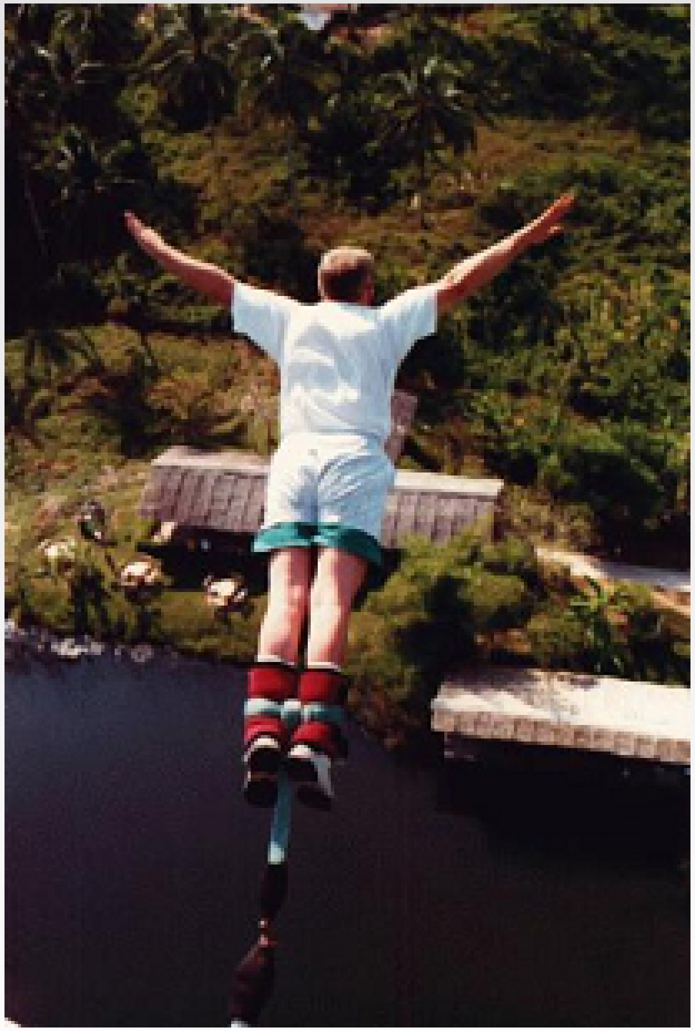

The filling process
Onion peel
Desiderata
b
c
d
b'
c'
d'

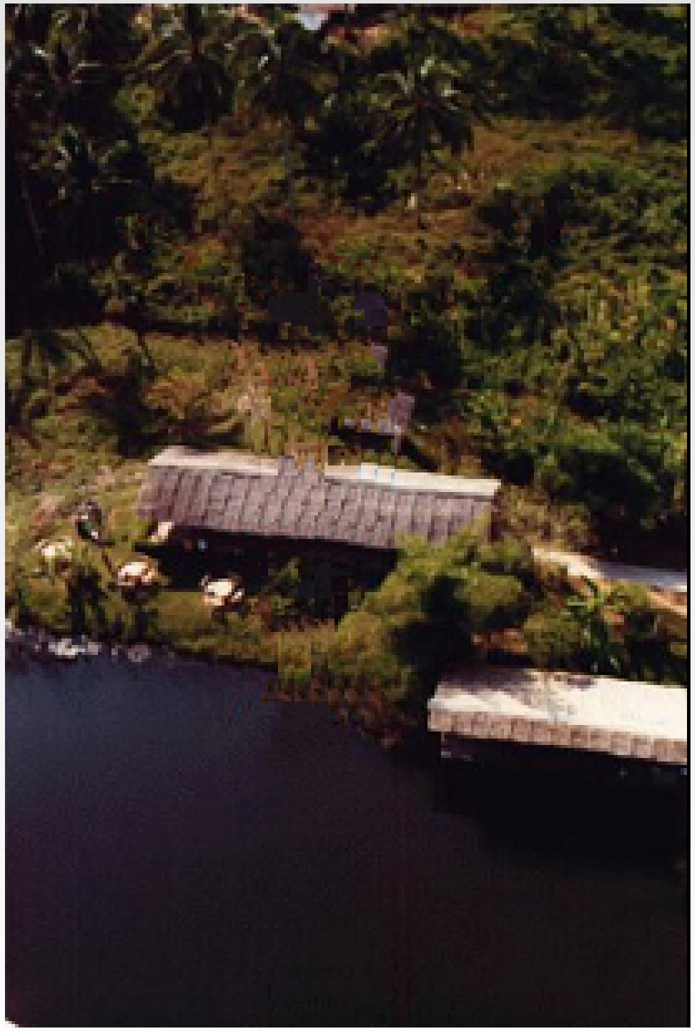

Φ
$\delta\Omega$
Ψ_{p}
p
$\mathbf{n}_{\mathrm{p}}$
$\nabla I_{\mathrm{p}}^{\perp}$
$\mathcal{I}$
Ω



Φ, Source region
$\delta\Omega$
Ω, Target region
a
Φ
$\delta\Omega$
p
Ψ_p
Ω
b
$\Psi_{q''}$
q''
$\Psi_{q'}$
q'
Ψ_p
Ω
c
Ψ_p
Ω
d

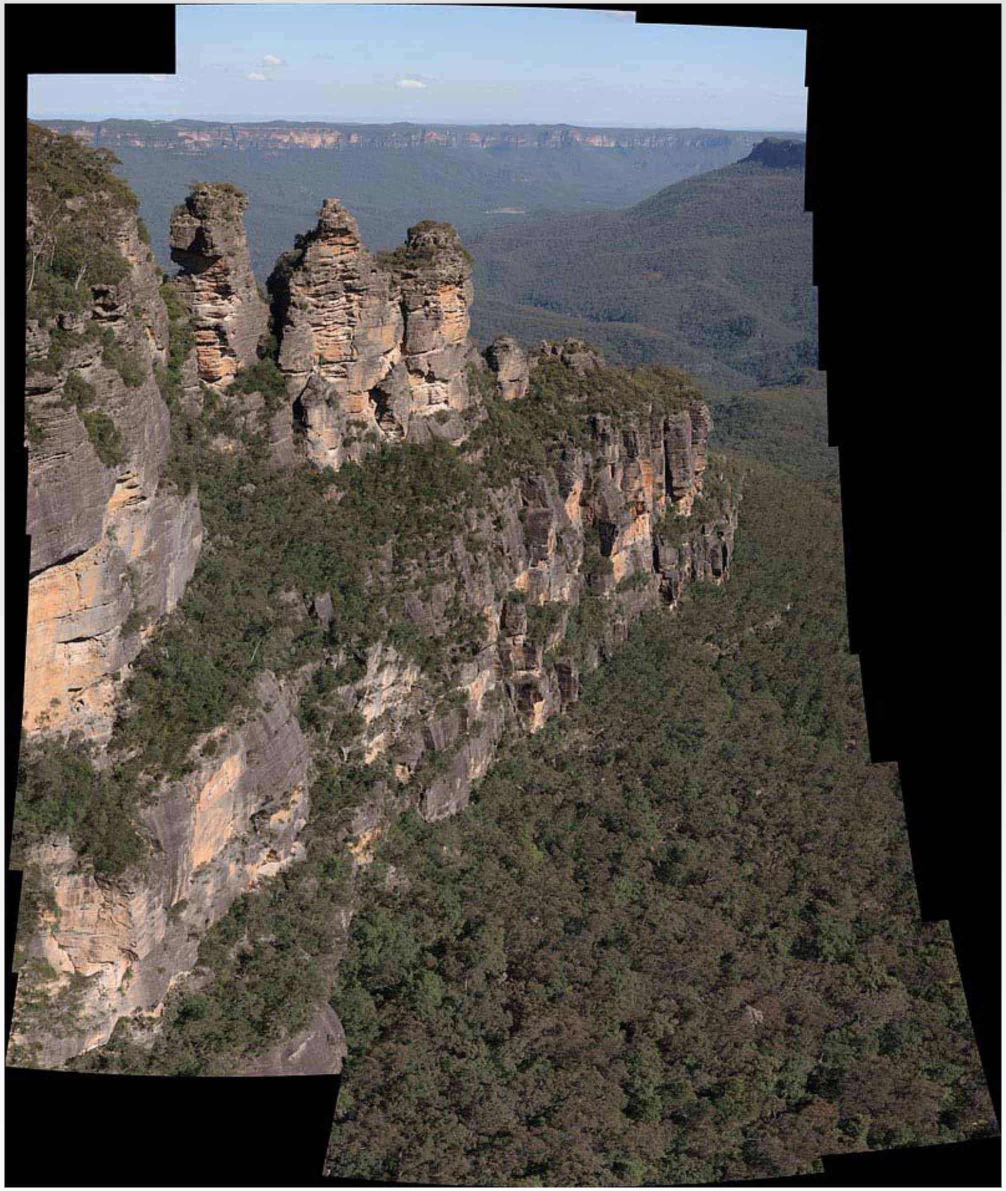
Results: Exemplar Based Image Inpainting



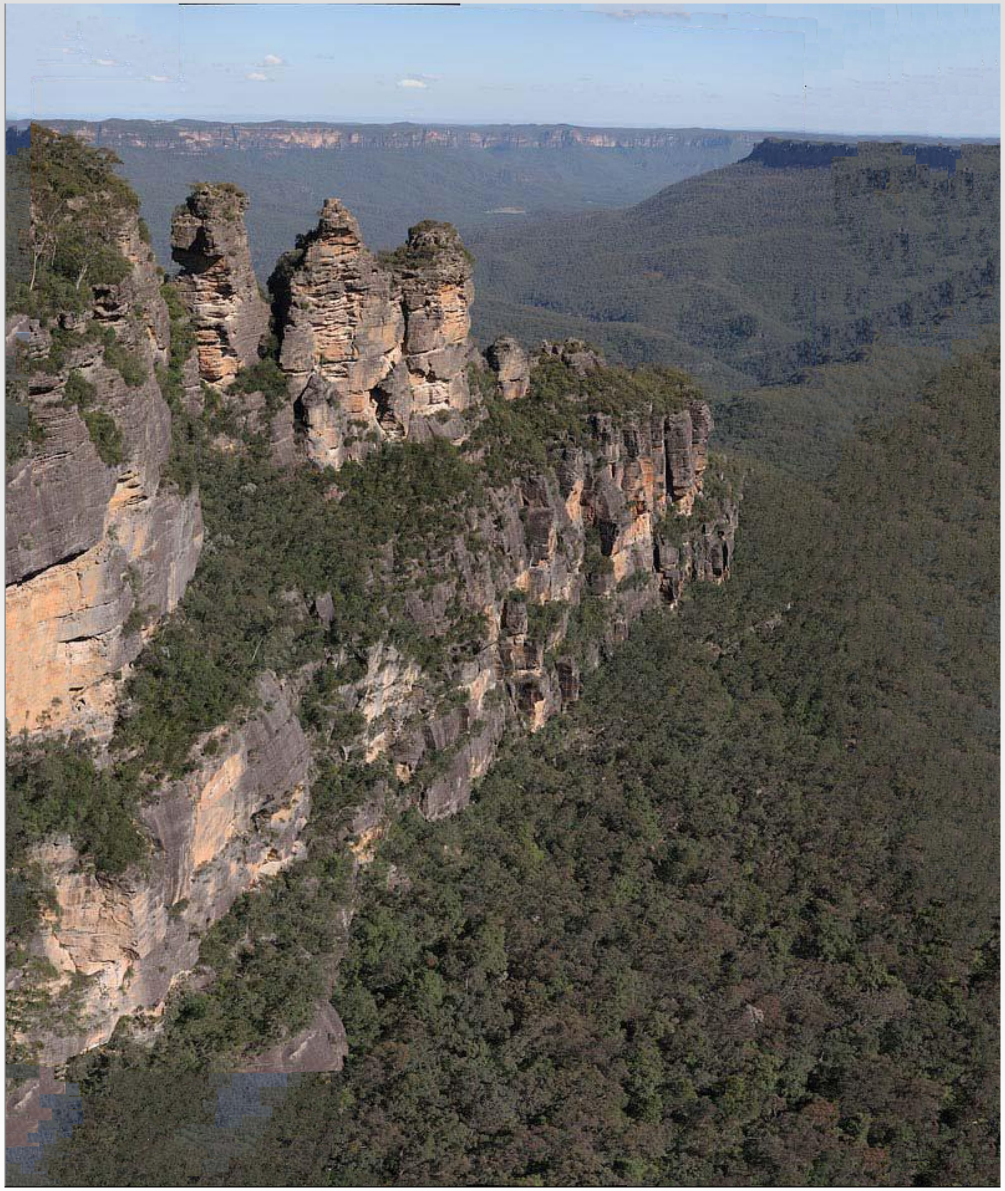


HOLLYWOOD

Results: EBII



Results: EBII

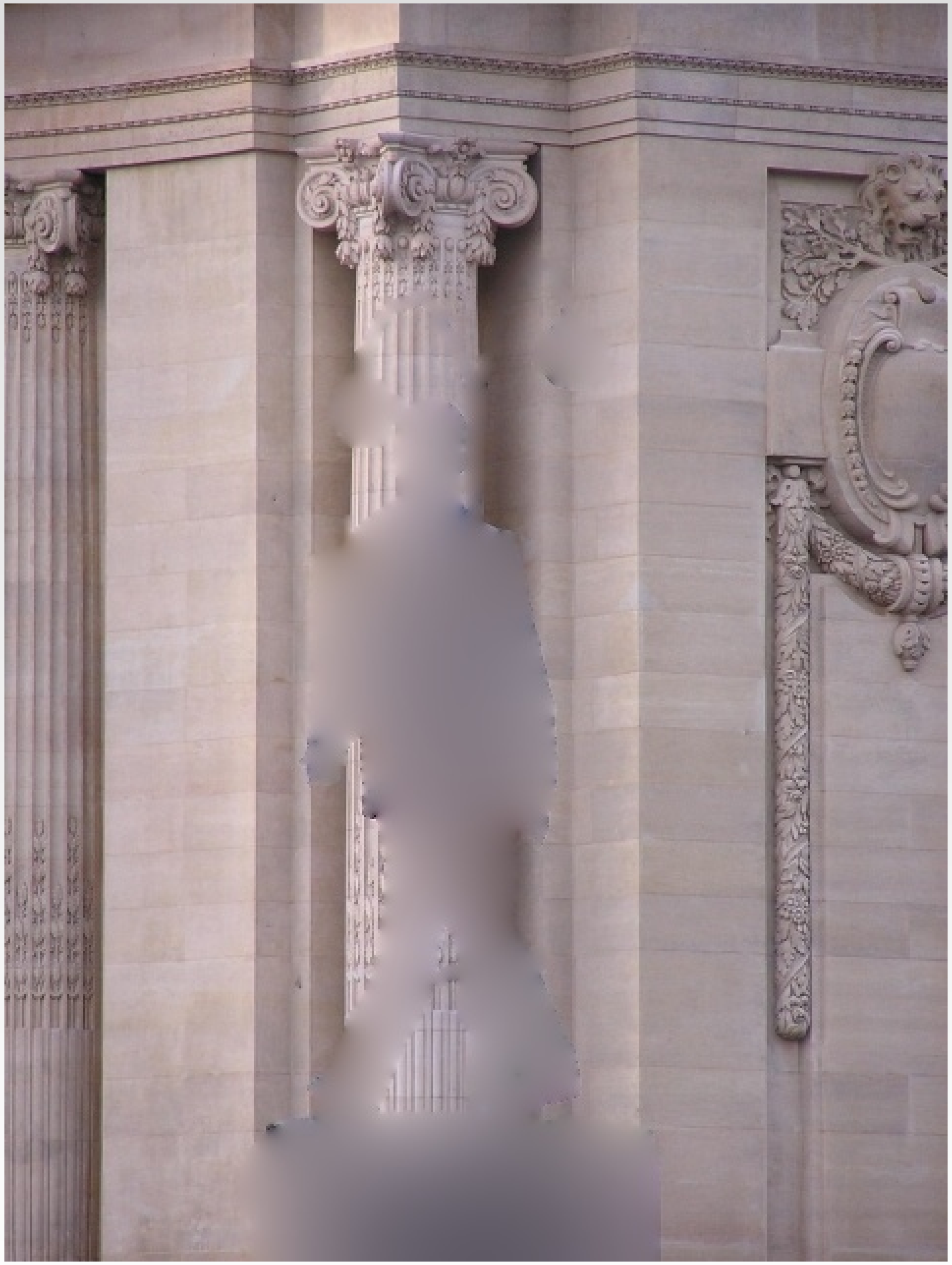


GPU Accelerated Implementation

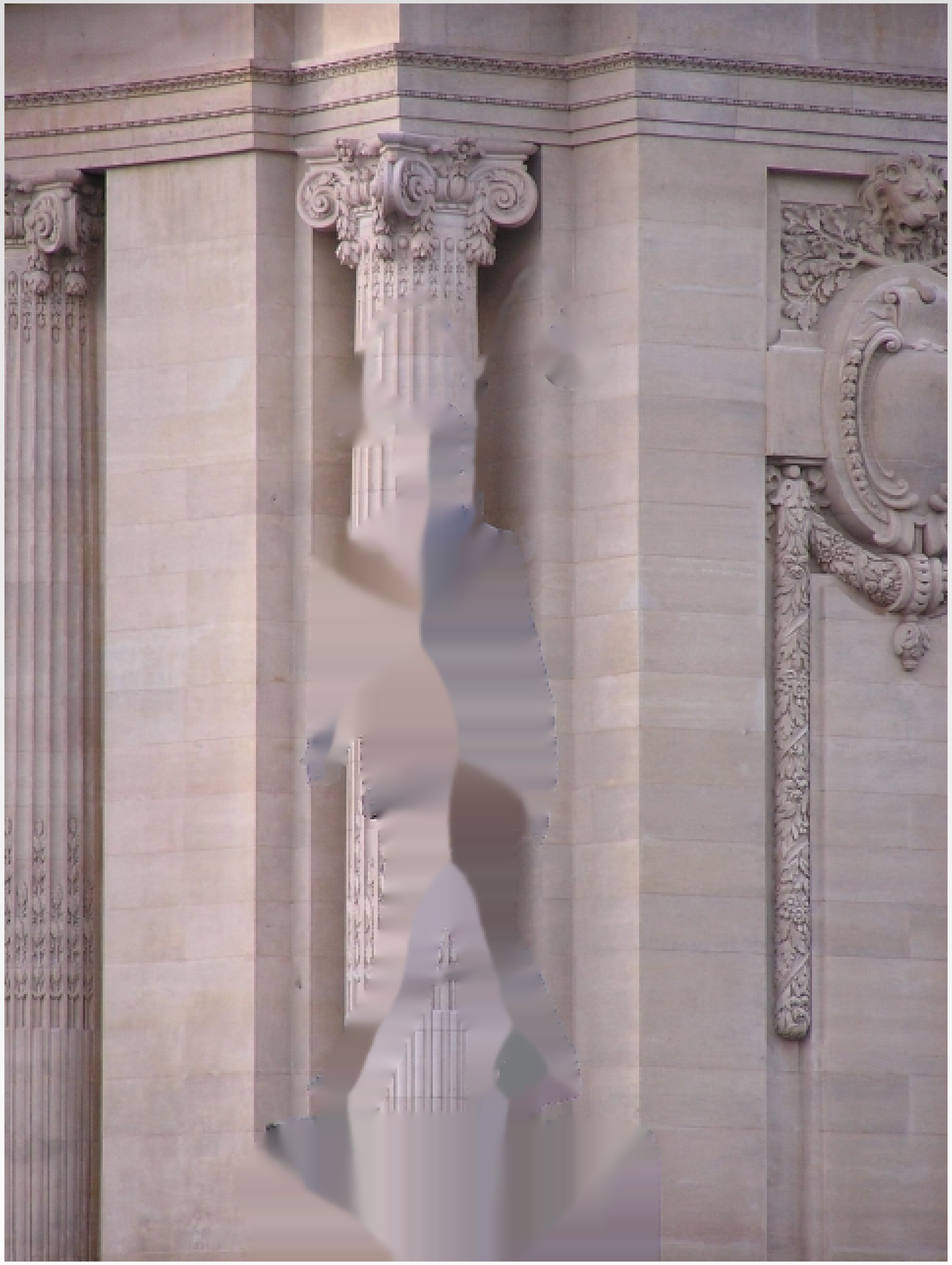
- Both Exemplar-Based Image Inpainting and Local Optimisation methods require the computer to search through the whole image for image patches matching a template patch.
- We implemented this in the GPU – much much faster (about 5 – 10x)
- Bottleneck is calculating border priorities, still done on CPU.

Hard Example

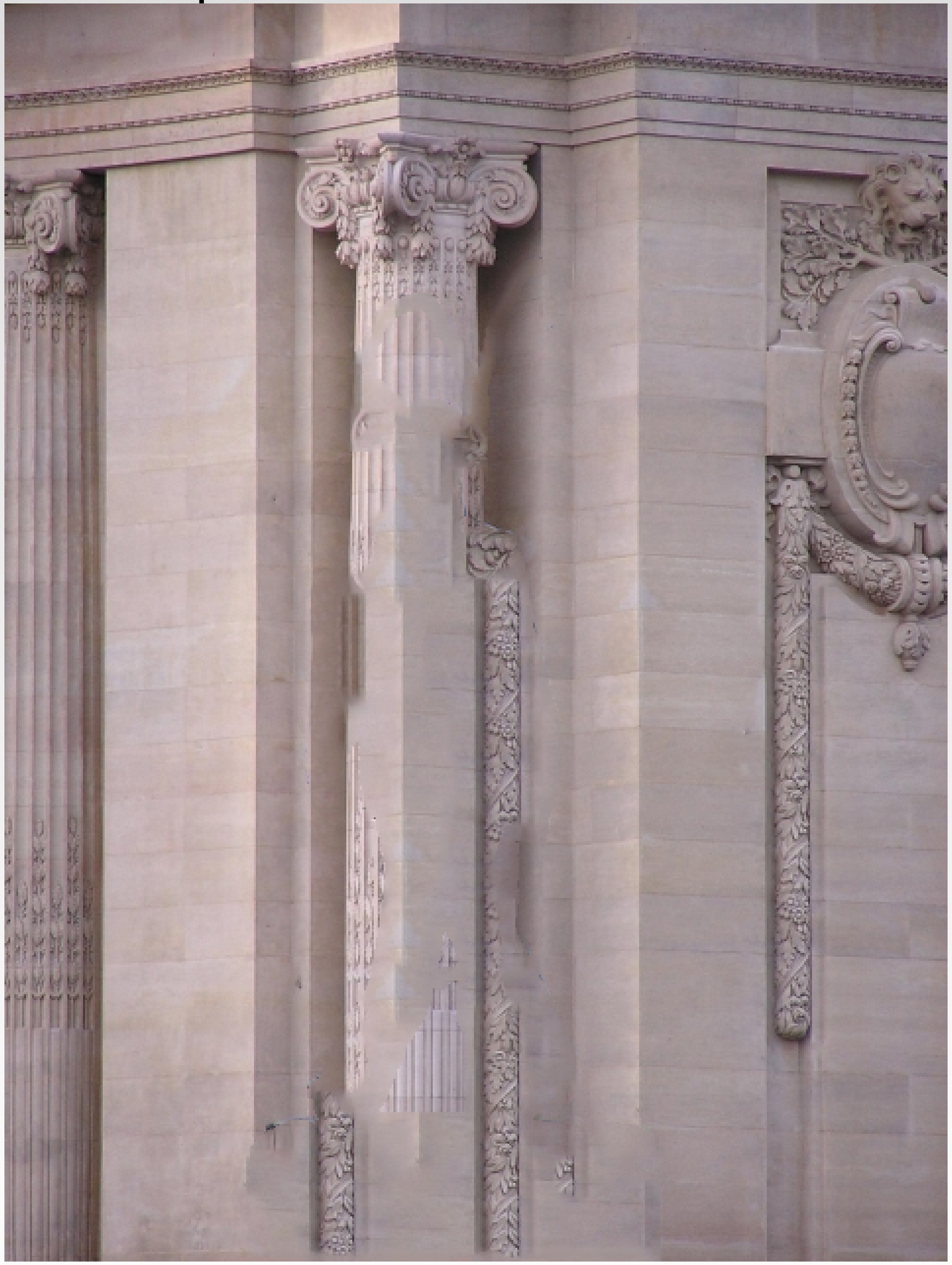
FDII



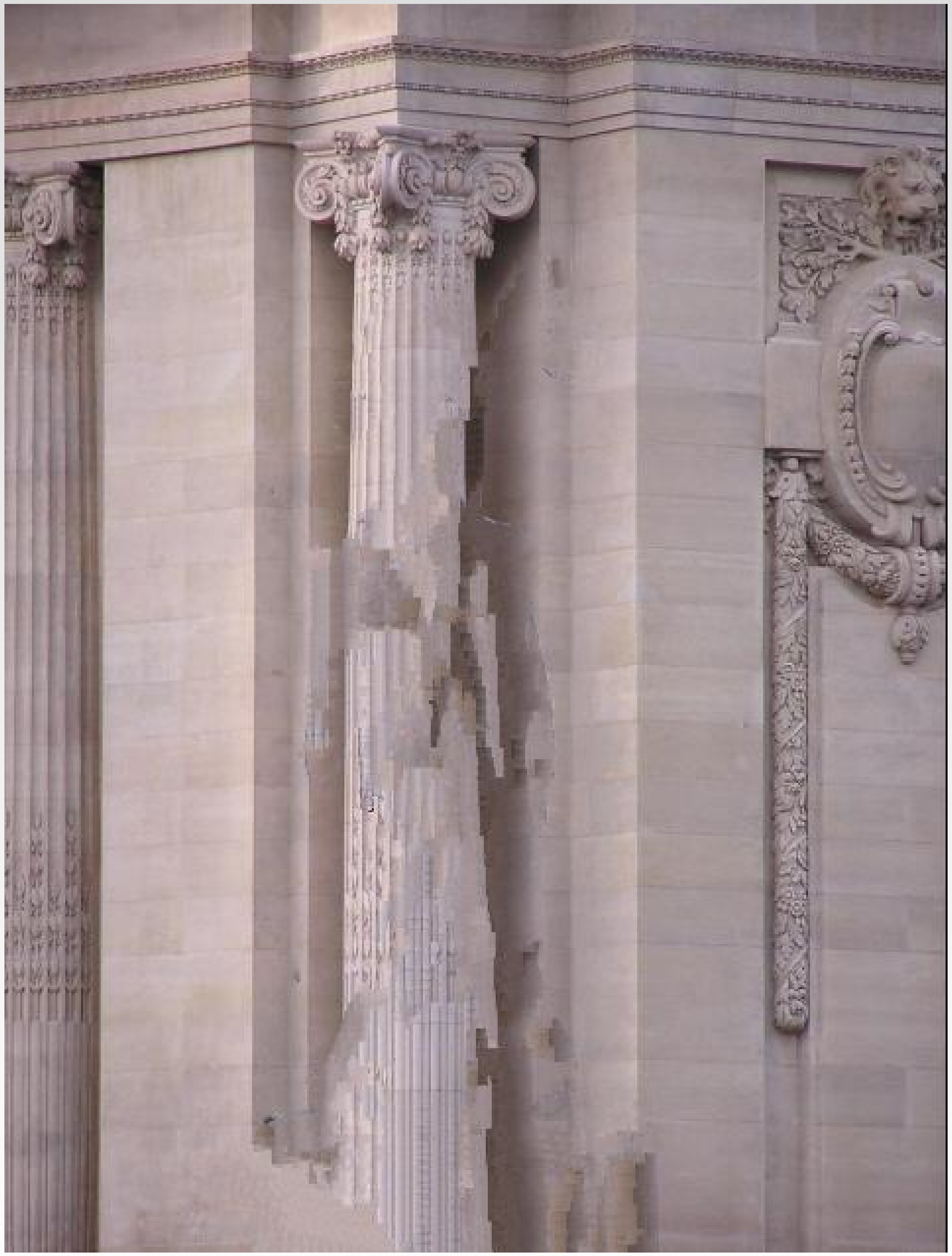
Navier-Stokes



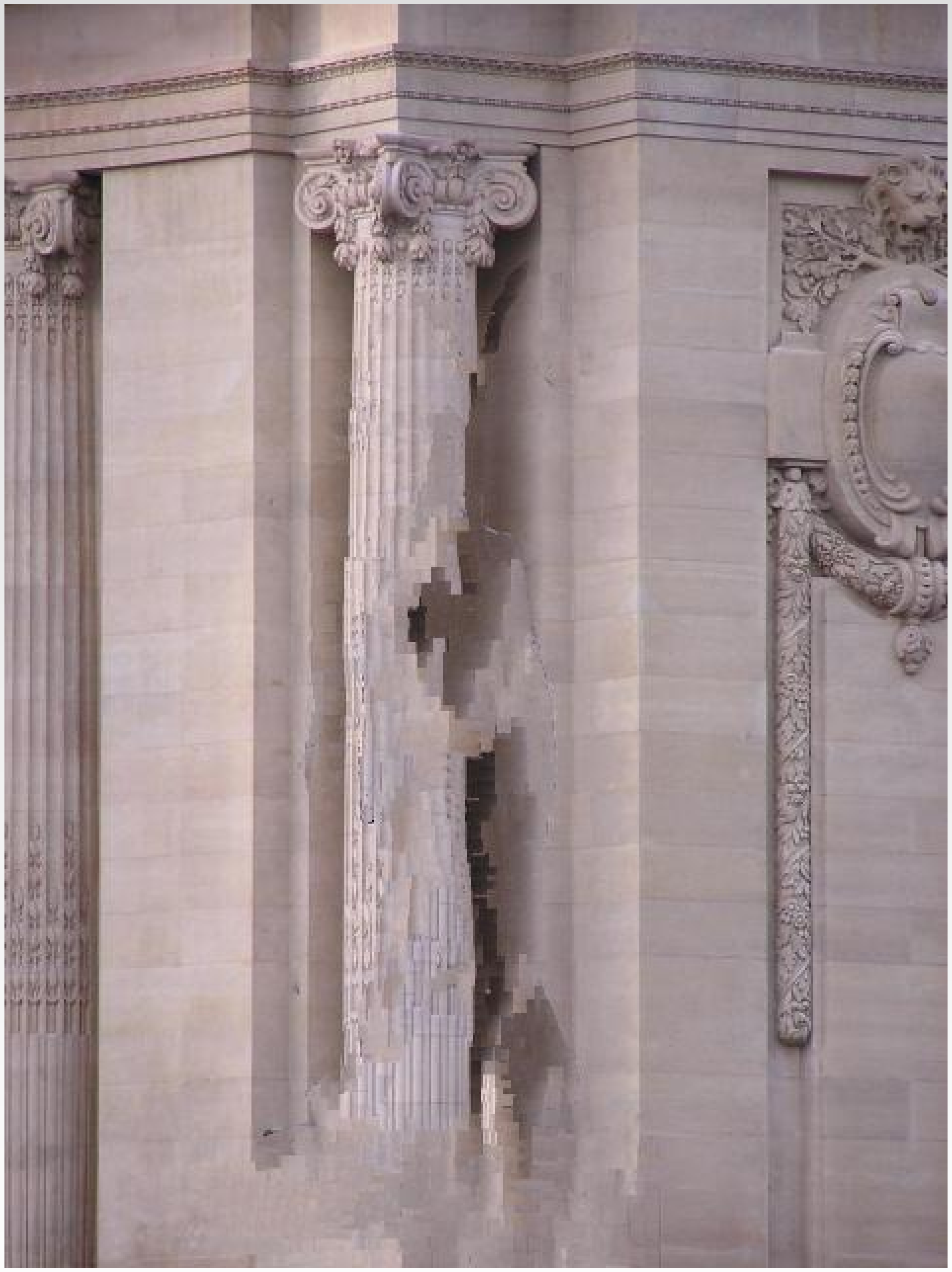
Adobe Photoshop CS5



Exemplar Based Image Inpainting



Local Optimisation



Future

- Current research – Adobe Photoshop CS5 – using random to speed up texture-based edge-aware algorithms
- Sacrifice result for speed
- Our opinion of future – GPU acceleration. Algorithms with highly separated stages designed for GPU
- Handheld devices – image inpainting apps?