Video Inpainting & Dynamic Texture Modeling

Alasdair Newson[3] **Andrés Almansa**[12] Matthieu Fradet[3]
Yann Gousseau[2] Patrick Perez[3]

1 2 3

Challenges in Dynamic Imaging Data - 9-11 June 2015
TGM - Isaac Newton Institute - Cambridge

Introduction

What is inpainting?

Introduction

What is inpainting?

- Removal and filling of a region in an image or video
- The inpainted region should be visually convincing/pleasing

Image to inpaint

Inpainted image

Introduction

What is inpainting?

- Removal and filling of a region in an image or video
- The inpainted region should be visually convincing/pleasing

Image to inpaint

Inpainted image

What is inpainting useful for?

- Restoring/improving/modifying images/videos
- Post-production of films

Original

Restored

Introduction

Video inpainting

Inpainted video

Introduction

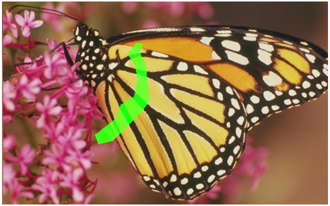
Video inpainting

Original video

Introduction

Challenges in *image* inpainting

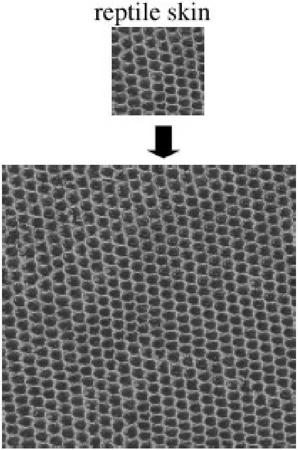
- Filling-in geometric structures (amodal completion)
  [Masnou & Morel 1998], [Bertalmio *et al.* 2000]



Introduction

Challenges in *image* inpainting

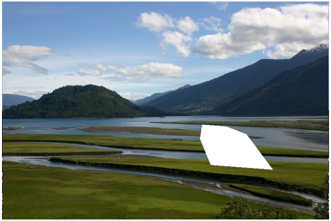
- Filling-in geometric structures (amodal completion)
  [Masnou & Morel 1998], [Bertalmio *et al.* 2000]
- Texture synthesis
  [Efros & Leung 1999]



Introduction

Challenges in *image* inpainting

- Filling-in geometric structures (amodal completion)
  [Masnou & Morel 1998], [Bertalmio *et al.* 2000]
- Texture synthesis
  [Efros & Leung 1999]
- Geometry + conditional texture synthesis
  [Cao *et al.* 2011]



Introduction

Additional challenges of *video* inpainting

- *Temporal coherency*
- *Dynamic geometry* (reconstruction of moving objects)
- *Dynamic texture* (water pouring, flowing, flames)
- *Simultaneous foreground/background reconstruction*
- *Extremely long computational times*

Inpainting example (from Wexler *et al.* 2007)

Proposed inpainting algorithm

Exemplar-based inpainting

Similar to [Wexler 2004] (video inpainting) and [Arias et al. 2012] (image inpainting)

Key ingredients

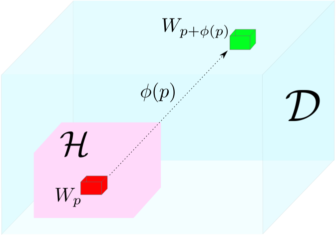
- spatio-temporal patches
- alternated minimization of bi-convex energy
- multi-scale coarse-to-fine model
- accelerated ANN search
- **texture-aware distance between patches**
- how many scales?
- motion compensation

Proposed inpainting algorithm

Video inpainting notation

Target patch

Nearest neighbour patch



ϕ Shift map (nearest neighbour offsets)

$\mathcal{D}$ Unoccluded region

$\mathcal{H}$ Occlusion

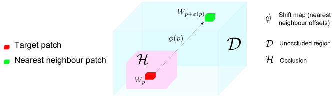
W_p: a patch centered at p

Proposed inpainting algorithm

Inpainting Principle

Input: $u|_{\mathcal{D}}$
Output: $u|_{\mathcal{H}}$
Find $u|_{\mathcal{H}}$ by minimizing

$$E(u,\phi) = \sum_{p \in \mathcal{H}} ||W_p^u - W_{p+\phi(p)}^u||_2^2$$



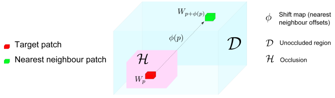
W_p: a patch centered at p

Proposed inpainting algorithm

Inpainting Principle

Input: $u|_{\mathcal{D}}$
Output: $u|_{\mathcal{H}}$
Find $u|_{\mathcal{H}}$ by minimizing

$$E(u,\phi) = \sum_{p \in \mathcal{H}} ||W_p^u - W_{p+\phi(p)}^u||_2^2$$



W_p: a patch centered at p

Challenges

- non-convex energy
- high dimensionality (dimension $= 5 \times 5 \times 5 \times 3 \approx 500$)

Solutions

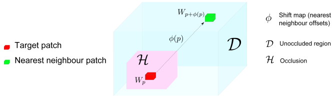
- alternate (convex) minimization w.r.t. u and ϕ
- coarse-to-fine processing
- approximate nearest neighbours
- fine-level texture features in coarsest level

Proposed inpainting algorithm

Inpainting Principle

Input: $u|_{\mathcal{D}}$
Output: $u|_{\mathcal{H}}$
Find $u|_{\mathcal{H}}$ by minimizing

$$E(u,\phi) = \sum_{p\in\mathcal{H}} ||W_p^u - W_{p+\phi(p)}^u||_2^2$$



W_p: a patch centered at p

Denoising Principle

Input: noisy $\tilde{u}$
Output: denoised $\hat{u}$
Find u by minimizing

$$E(u,w) = \sum_{p,q} w(p,q)||W_p^{\tilde{u}} - W_q^u||_2^2 + h\sum_p H(w(p,\cdot))$$

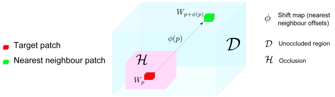
‡ *P. Arias, G. Facciolo, V. Caselles, G. Sapiro,* **A Variational Framework for Exemplar-Based Image Inpainting**, *IJCV 2011*

Proposed inpainting algorithm

Inpainting Principle

Input: $u|_{\mathcal{D}}$
Output: $u|_{\mathcal{H}}$
Find $u|_{\mathcal{H}}$ by minimizing

$$E(u,\phi) = \sum_{p \in \mathcal{H}} ||W_p^u - W_{p+\phi(p)}^u||_2^2$$



W_p: a patch centered at p

Algorithm (inspired by † and ‡):

Alternate Minimization on u and ϕ:

$u^0 \leftarrow$ Initialisation($u|_{\mathcal{D}}$,$\mathcal{H}$)
1/ $\phi^{k+1} \leftarrow$ NearestNeighbourSearch(u^k)
2/ $u^{k+1} \leftarrow$ VideoReconstruction(ϕ^{k+1})
(aggregation of patches)

(Carried out in a Multiresolution scheme)

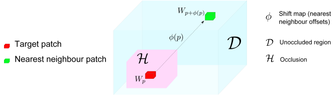
† *Y. Wexler, E. Schechtman, M. Irani,* **Space-Time Completion of Video**, *PAMI 2007*
‡ *P. Arias, G. Facciolo, V. Caselles, G. Sapiro,* **A Variational Framework for Exemplar-Based Image Inpainting**, *IJCV 2011*

Proposed inpainting algorithm

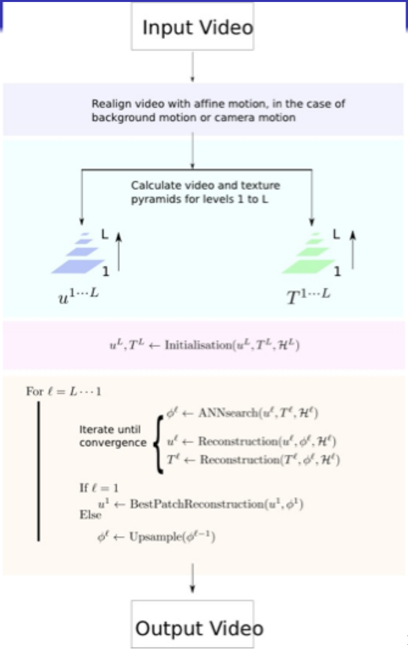
Inpainting Principle

Input: $u|_{\mathcal{D}}$
Output: $u|_{\mathcal{H}}$
Find $u|_{\mathcal{H}}$ by minimizing

$$E(u,\phi) = \sum_{p \in \mathcal{H}} ||W_p^u - W_{p+\phi(p)}^u||_2^2$$



W_p: a patch centered at p

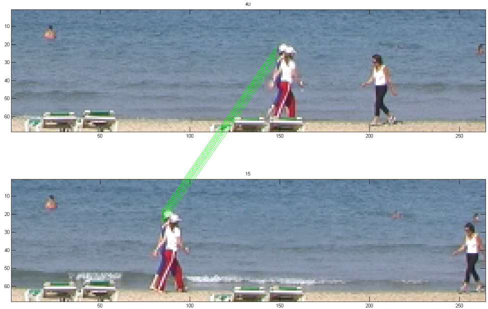


Approximate Nearest Neighbour (ANN) search

Approximate Nearest Neighbour (ANN) search

High dimensionality of problem means NN search is *very* slow

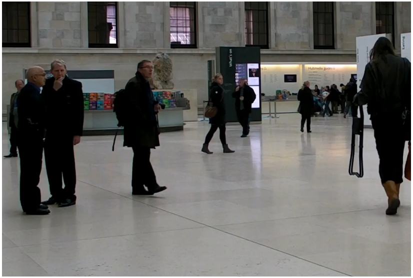
- Previously used ANN search algorithm (kdTrees) very slow
- We extend the PatchMatch [Barnes *et al.* 2009]† algorithm to spatio-temporal case.
- PatchMatch based on piecewise constancy of the shift map ϕ



† *C. Barnes, E. Schechtman, A. Finkelstein, D. B. Goldman,* **PatchMatch: a randomized correspondence algorithm for structural image editing**, *ACM Transactions on Graphics (2009)*

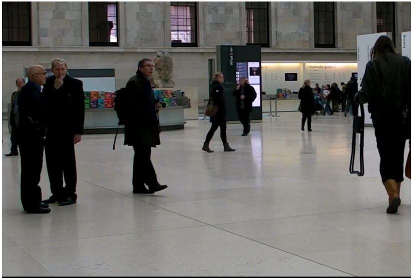
Visual comparisons [Granados *et al.* 2012]

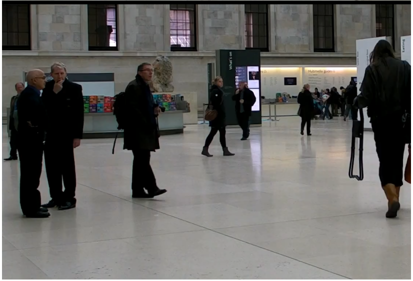
High definition example (1120 × 754)



- 10-50 times speedup with 3D PatchMatch
- 10 times speedup compared to Granados *et al.*

M. Granados, J. Tompkin, K.I. Kim, O. Grau, J. Kautz, C. Theobalt,
How Not to Be Seen - Object Removal from Videos of Crowded Scenes,
Computer Graphics Forum, 2012





Visual comparisons [Granados *et al.* 2012]

	One matching pass at full res.		
	Beach umbrella (265x68x200)	Crossing ladies (170x80x87)	Jumping girl (1120x754x200)
Wexler (kd-tree)	∼1000s	∼1000s	∼8000s
Ours (PatchMatch3D)	∼50s	∼30s	∼150s
	Total timing		
	Beach umbrella (265x68x200)	Duo (960x704x154)	Museum (1120x754x200)
Granados (graph-cut)	11h		90h
Ours w/o texture	14mn	4h	4h
Ours	24mn	6h	6h

Textures in image/video inpainting

Dealing with textures in images and videos

Why do textures pose a problem ?

Original image

Dealing with textures in images and videos

Why do textures pose a problem ?

Inpainted image

Dealing with textures in images and videos

Why do textures pose a problem ?

Incorrect approximate nearest neighbours

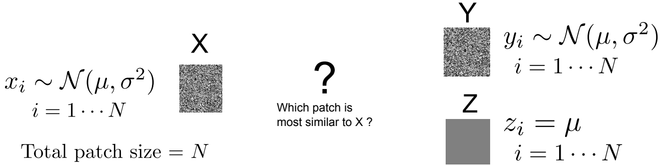
Dealing with textures in images and videos

Why do we identify incorrect patches ???

Dealing with textures in images and videos

Why do we identify incorrect patches ???

Imagine we want to find the ANN of a *random* patch:



Dealing with textures in images and videos

Why do we identify incorrect patches ???

Imagine we want to find the ANN of a *random* patch:

$x_i \sim \mathcal{N}(\mu, \sigma^2)$
$i = 1 \cdots N$

X

Total patch size $= N$

?

Which patch is most similar to X ?

Y

$y_i \sim \mathcal{N}(\mu, \sigma^2)$
$i = 1 \cdots N$

Z

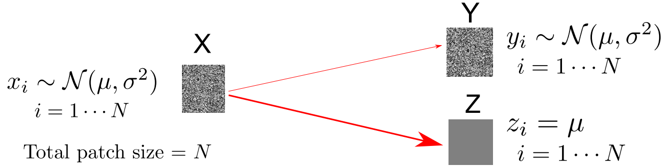
$z_i = \mu$
$i = 1 \cdots N$

$$E[d(X, Y)] = 2N\sigma^2$$
$$E[d(X, Z)] = N\sigma^2$$

Dealing with textures in images and videos

Why do we identify incorrect patches ???

Imagine we want to find the ANN of a *random* patch:



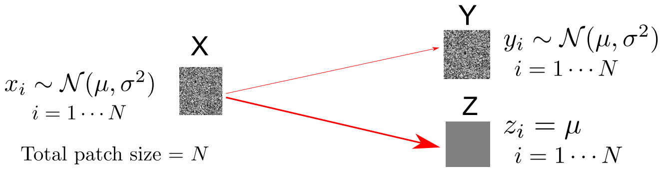
$$E[d(X, Y)] = 2N\sigma^2$$
$$E[d(X, Z)] = N\sigma^2$$

On average, $d(X, Y)$ is *twice* as large as $d(X, Z)$.
On average, constant patch Z is preferred !

Dealing with textures in images and videos

Why do we identify incorrect patches ???

Imagine we want to find the ANN of a *random* patch:



On average, $d(X, Y)$ is *twice* as large as $d(X, Z)$.

On average, constant patch Z is preferred !

Solution ? Change patch distance !

Modified patch distance

We wish to include some information pertaining to the texture.

Idea : include an estimation of the local variance

Modified patch distance

We wish to include some information pertaining to the texture.

Idea : include an estimation of the local variance

Different possibilities were tested. Finally, we chose (inspired by Liu and Caselles 2013[†]) :

$$\text{SSD: } [\mathrm{R}, \mathrm{G}, \mathrm{B}, \alpha g_\nu * |\nabla_x I|, \alpha\, g_\nu * |\nabla_y I|]$$

α: a weighting scalar

g_ν in a gaussian kernel of size ν .

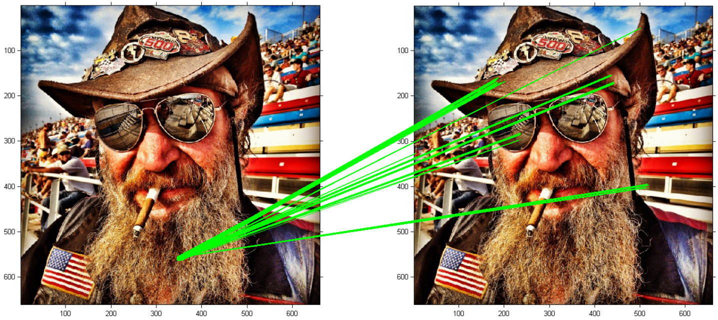
[†] *Y. Liu, V. Caselles,* **Exemplar-Based Image Inpainting Using Multiscale Graph Cuts**, *IEEE TIP (2013)*
[‡] *J. Bruna & S. Mallat (2013).* **Invariant scattering convolution networks.** *IEEE TPAMI, 35(8), 187286*

Modified patch distance

Example of image created by $|\nabla_x I|_\nu$

Modified patch distance

Example of the impact of the modified distance



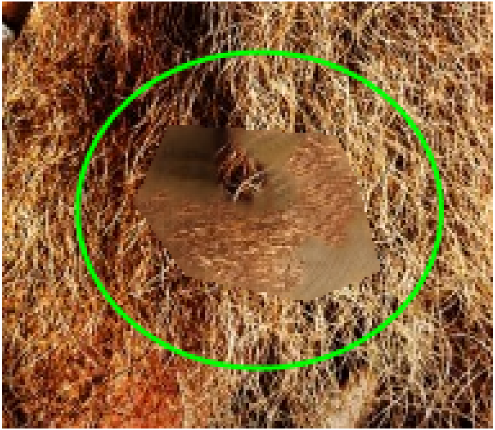
PatchMatch with regular SSD

Modified patch distance

Example of the impact of the modified distance

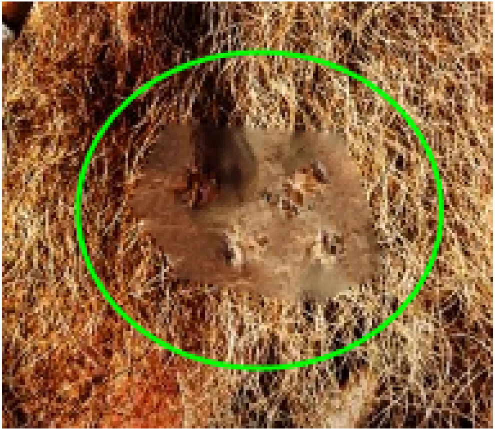
PatchMatch with modified SSD

Image example



Inpainting with unmodified patch distance

Image example



Inpainting with "Image Melding" (Darabi *et al.* 2012)

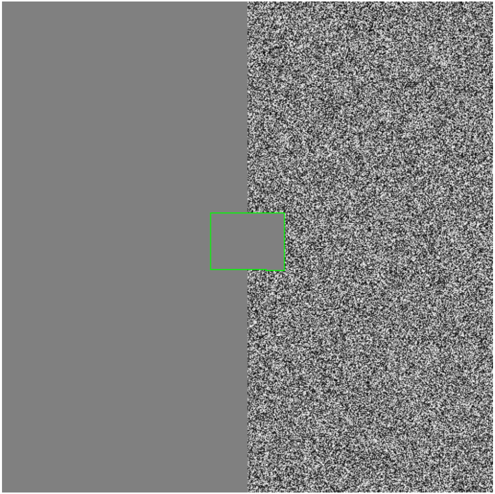
Image example

Inpainting with modified patch distance

Image example

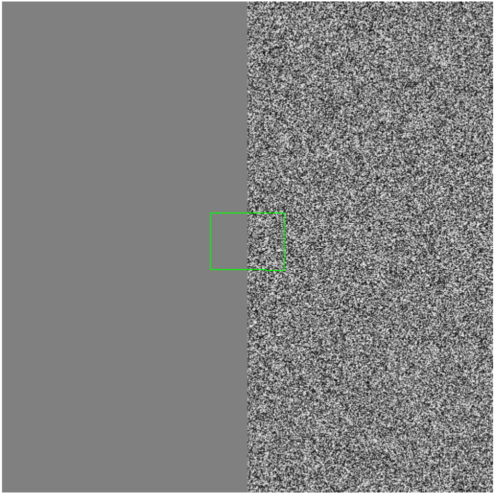
Original image

Noise example



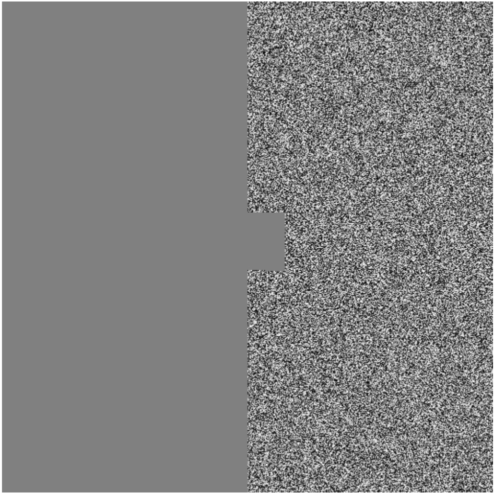
Inpainting with unmodified patch distance

Noise example



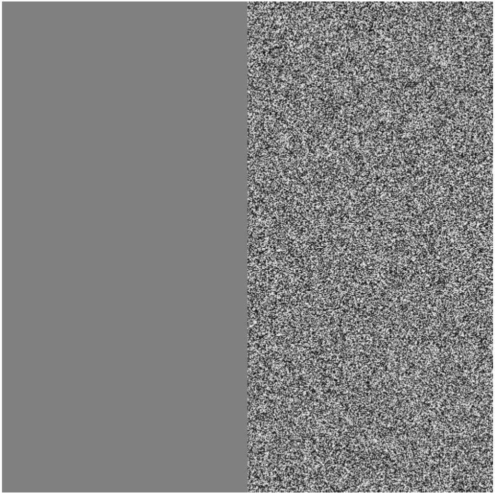
Inpainting with modified patch distance

Noise example



Inpainting with unmodified patch distance

Noise example



Inpainting with modified patch distance

Video example

Original video

Video example

Unmodified patch distance

Video example

Modified patch distance

Local minima, convergence and binary inpainting

Convergence questions, local minima

The *multi-resolution* scheme is necessary to correctly inpaint structures.

Occluded image

Result with one pyramid level

Result with three pyramid levels

Convergence questions, local minima

The *multi-resolution* scheme is necessary to correctly inpaint structures.

Occluded image

Result with one pyramid level

Result with three pyramid levels

Some interesting questions:

- Can we quantify the amount of subsampling needed ?
- Can we guarantee convergence to a desirable solution ?

Such questions are difficult to answer in general, so we use a simple situation !

Convergence questions, local minima

The *multi-resolution* scheme is necessary to correctly inpaint structures.

Occluded image

Result with one pyramid level

Result with three pyramid levels

- Study (very) simple situation
- Behaviour of the algorithm is easier to study

Binary inpainting

Main theoretical results in simple 1D case :

- Algorithm converges if the occlusion size is less than $2N - 2\sqrt{N} + 1$
  - N is the patch size
- Otherwise, algorithm may be stuck in a local minimum

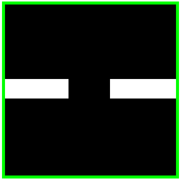
Binary inpainting

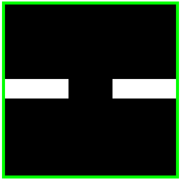
Main theoretical results in simple 1D case :

- Algorithm converges if the occlusion size is less than $2N - 2\sqrt{N} + 1$
  - N is the patch size
- Otherwise, algorithm may be stuck in a local minimum

Verified for simple 2D situations.

Simple structure example. Patch size $= 11 \times 11$





Occlusion size $= 16$ Occlusion size $= 17$

Inpainting results

Original video

Inpainting results

Our inpainting result

Inpainting results

Original video

Inpainting results

Our inpainting result

Open Issues

- complex motions
- long temporal occlusions
- select the patch size

Input image (occlusion border in red)

Patch size $= 3 \times 3$

Patch size $= 5 \times 5$

Patch size $= 7 \times 7$

Patch size $= 9 \times 9$

Patch size $= 11 \times 11$

Perspectives & Challenges

- More general features to discriminate dynamic textures and shapes
  - Motion features
  - True multi-scale criterion
  - Scattering transform [Bruna & Mallat 2013]†
- Generative vs. deterministic inpainting:
  - Sample from a local conditional gaussian model instead of NN [Raad *et al*. 2015]‡
  - Accelerated learning/querying of local Gaussian models [Guillemot *et al*. 2014]*

† *Bruna, J., & Mallat, S. (2013).* **Invariant scattering convolution networks.** *IEEE TPAMI, 35(8), 187286*

‡ *L. Raad, A. Desolneux, J-M. Morel (2014),* **Locally Gaussian Exemplar-Based Texture Synthesis**

* *T. Guillemot, A. Almansa, T. Boubekeur,* **Covariance Trees for 2D and 3D processing**, *CVPR 2014.* *http://perso.telecom-paristech.fr/~boubek/papers/CovTree/*

Thank you for listening !

- More **videos / paper / source-code**: http://perso.enst.fr/~almansa/video_inpainting/
- Come see our **poster** on *Single-Shot High Dynamic Range Imaging* !!
- **Open PhD Position**

  Subject Video Inpainting

  Location Paris and/or Lyon

  Supervisors A. Almansa, Y. Gousseau, S. Masnou