Visual inpainting

- **Physical**: painting missing parts in art restoration
  
  ![Image of art restoration process](image1)

- **Computational**: completion of visual data, given surrounding
  - Visually plausible, at least pleasing
  
  ![Image of computational inpainting](image2)

[Crerinisi et al. 2003]
Applications

- Object removal, concealment of all sorts (including packet loss)

Welcome to King's College, a constituent college of Cambridge University, England. King's is one of the oldest Cambridge colleges, having been founded in 1441 by Henry VI. It is also Cambridge's premier tourist attraction, due above all to its spectacular Perpendicular chapel.

[Pérez et al. 2004]
Applications

- Object **removal**, concealment of all sorts (including packet loss)

[Pérez et al. 2004]
Applications

- Image extension

[Pérez et al. 2004]
Applications

- Image *extension*

[Pérez et al. 2004]
Brief (partial) history

- **1998**: Masnou and Morel. “Disocclusion” with isophote completion
- **2000**: Bertalmio *et al.* Inpainting as PDE

- **1999**: Efros and Leung. Example-based texture synthesis
- **2001**: Harrison. Inpainting with example-based texture synthesis (GIMP)
- **2003**: Criminisi *et al.* Greedy patch-based inpainting (MS DIP)
- **2004**: Wexler *et al.* Iterative patch-based video inpainting
- **2009**: Barnes *et al.* “PatchMatch” (Photoshop)
- **2010**: Aujol *et al.* Variational patch-based inpainting
- **2012**: Granados *et al.* HD video inpainting
A greedy patch-based method

Dictionary of block’s neigh.

[Pérez et al. 2004]
A greedy patch-based method
A greedy patch-based method
A greedy patch-based method
A greedy patch-based method
A greedy patch-based method
A greedy patch-based method
Why square patches?

- Capture both geometry and texture
- Useful to speed things up

5x5, 1mn

54x54, 0.1s

[ Pérez et al. 2004 ]

- Easy to manipulate
- Can be linearly combined
- Have become variationally friendly
Why square patches?

- Capture both geometry and texture
- Useful to speed things up

5x5, 1mn

54x54, 0.1s

Easy to manipulate
- Can be linearly combined
- Have become variationally friendly

[Pérez et al. 2004]
Video inpainting

- **Not** a sequence of image inpainting

**Challenges**
- Plausible appearance and motion
- Shot-wise consistency
- Computational complexity

**First serious attempt:** Wexler *et al.* 2004
Video inpainting made easy

- Builds on Wexler *et al.*
  - Coarse-to-fine
  - Alternate patch search / reconstruction

- Improvements
  - Variational
  - Way faster
  - Suited to complex HD scenes
  - Reproducible

(960x704x154)
Notations

**Color image** \( u : \Omega \rightarrow \mathbb{R}^3 \)

**Spatio-temporal cuboids** \( \mathcal{N}_p \subset \Omega \) centered at \( p = (x, y, t) \)

**Spatio-temporal image patch** \( W_p = [u(q)]_{q \in \mathcal{N}_p} \in \mathbb{R}^{3 \times N} \)

**Hole and data** \( \Omega = \mathcal{H} \cup \mathcal{D} \), \( u^* = u|_\mathcal{D} \)

**Source of known patches** \( \mathcal{D} = \{q : \mathcal{N}_q \subset \mathcal{D}\} \)

**Correspondence map** \( T : \mathcal{H} \rightarrow \mathcal{D} \)

**Shift map** \( \phi = T - \text{Id} \)
Patch-based energy

**Compound energy** [Demanet 2003, Aujol 2009, Arias 2011]

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

\[
= \sum_{p \in \mathcal{H}} \sum_{q \in \mathcal{N}_p} \|u(q) - u^*(q + \phi(p))\|^2_2
\]
Patch-based energy

- **Compound energy**

\[
E(u|\mathcal{H}, \phi) = \sum_{p \in \mathcal{H}} \|W_p - W_{p+\phi(p)}\|^2_2 \\
= \sum_{p \in \mathcal{H}} \sum_{q \in \mathcal{N}_p} \|u(q) - u^*(q + \phi(p))\|^2_2
\]

- Special case of [Aujol et al. 2009] (no regularization)
- Special case of [Arias et al. 2011] (single correspondant)
- Relaxation of [Demanet et al. 2003]

\[
E(\phi) = \sum_{p \in \mathcal{H}} \sum_{q \in \mathcal{N}_p} \|u^*(q + \phi(q)) - u^*(q + \phi(p))\|^2_2
\]

- Boundary conditions

\[
\|W_p - W_{p+\phi(p)}\|^2_2 = \sum_{q \in \mathcal{N}_p \cap \mathcal{H}} \|u(q) - u^*(q + \phi(p))\|^2_2 \\
+ \sum_{q \in \mathcal{N}_p \cap \mathcal{D}} \|u^*(q) - u^*(q + \phi(p))\|^2_2
\]
Energy minimization

- Alternate minimization

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

\[ = \sum_{p \in \mathcal{H}} \sum_{q \in \mathcal{N}_p} \| u(q) - u^*(q + \phi(p)) \|_2^2 \]

- Patch matching, given image

\[ p + \phi(p) = T(p) = \arg \min_{r \in \tilde{D}} \| W_p - W_r \|_2^2, \quad \forall p \in \mathcal{H} \]

- Image reconstruction, given shift map

\[ u(q) = \frac{\sum_{p \in \mathcal{N}_q \cap \mathcal{H}} u^*(q + \phi(p))}{|\mathcal{N}_q \cap \mathcal{H}|}, \quad \forall q \in \mathcal{H} \]

except for final reconstruction: copy from best patch match
Final reconstruction

Weighted average
Final reconstruction

Best match
Coarse-to-fine

- To improve speed and result quality for large occlusions
- Image pyramid, same patch size through it
- Several iterations (e.g., 20 max.) at each level and up-sample to next
- First filling-in at coarsest resolution: greedy (concentric)
Coarse-to-fine

- To improve speed and result quality for large occlusions
- Image pyramid, same patch size through it
- Several iterations (e.g., 20 max.) at each level and up-sample to next
- First filling-in at coarsest resolution: greedy (concentric)
Fast approximate patch matching

- **PatchMatch** [Barnes 2009]
  - Fast, dense, approximate patch correspondences

- **Key ideas**
  - Shift map is piece-wise constant
  - Randomization

- **Principle**
  - Lexicographic passes
  - Past neighbors propose their shifts
  - Random proposal as well

- **Proposed**
  - Straightforward adaptation to 2D+t
SD inpainting of a static mask

(265x68x200) sequence: x20 speed-up
HD inpainting of a moving mask

(1120x754x200) sequence: 4h vs. 90h
HD inpainting of a moving mask

(1120x754x200) sequence: 4h vs. 90h
Dealing with texture

- Euclidean patch distance not sensitive enough to fine grain texture
Dealing with texture

- Euclidean patch distance not sensitive enough to fine grain texture
Extended patch

- Include simple texture information (inspired by Liu and Caselles 2013)

\[
W_p = [(u(q), \lambda v(q))]_{q \in N_p} \quad v(q) = \frac{1}{|\nu|} \sum_{r \in \nu} (|I_x(r)|, |I_y(r)|)
\]
Extended patch

- Include simple texture information (inspired by Liu and Caselles 2013)

\[
W_p = \left[(u(q), \lambda v(q))\right]_{q \in \mathcal{N}_p} \quad v(q) = \frac{1}{|\nu|} \sum_{r \in \nu} (|I_x(r)|, |I_y(r)|)
\]
Texture-aware inpainting

ours with texture

ours with texture
Texture-aware inpainting

ours with texture

ours with texture
Texture-aware inpainting

original
Texture-aware inpainting

w/o texture
Texture-aware inpainting
Dealing with camera motion

- **Camera motion**, even small, is a problem
- **Good patch match** = similar structure with *similar apparent movement*

\[
W_p = \begin{bmatrix}
  \text{frame}_1 & \text{frame}_2 & \text{frame}_3 & \text{frame}_4 \\
  t & & & \\
\end{bmatrix}
\]

\[
W_{p+\phi(p)} = \begin{bmatrix}
  \text{frame}_1 & \text{frame}_2 & \text{frame}_3 & \text{frame}_4 \\
  t + \phi(t) & & & \\
\end{bmatrix}
\]

- **Simple solution**
  - Stabilize (estimate and *compensate dominant motion*)
  - Inpaint video
  - Put dominant motion back
Dealing with camera motion
Dealing with camera motion

w/o stabilization
Dealing with camera motion

with stabilization
Results of complete system
Results of complete system
Results of complete system
Results of complete system
Conclusion and outlooks

- **Proposed video inpainting [Newson et al. 2013, 2014]**
  - Complex dynamic HD scenes
  - Generic and automatic
  - Data and code available ([www.enst.fr/~gousseau/video_inpainting](http://www.enst.fr/~gousseau/video_inpainting))
  - Soon IPOL online demo
  - Still too slow and visible artefacts

- **Beyond texture and structure**
  - Reflectance and transparency
  - Illumination and shading

- **Other signal completion problems**
  - Unconventional images (HDR, plenoptic, multispectral)
  - Geometry (depth maps, point clouds and meshes)
  - Audio (declipping, declicking)
Thanks

- Andrés Almansa (CNRS)
- Andrew Blake (Microsoft Research)
- Frédéric Cao (DxO)
- Antonio Criminisi (Microsoft Research)
- Matthieu Fradet (Technicolor)
- Michel Gangnet (Microsoft Research)
- Yann Gousseau (TelecomParisTech)
- Simon Masnou (University of Lyon)
- Alasdair Newson (Technicolor)
- Kentaro Toyama (Microsoft Research)