Image and video inpainting

MVA 2024-2025

Yann Gousseau
Telecom Paris

Image Inpainting

Inpainting (disocclusion) : How to fill missing regions in images ?
Should be done in a plausible way.

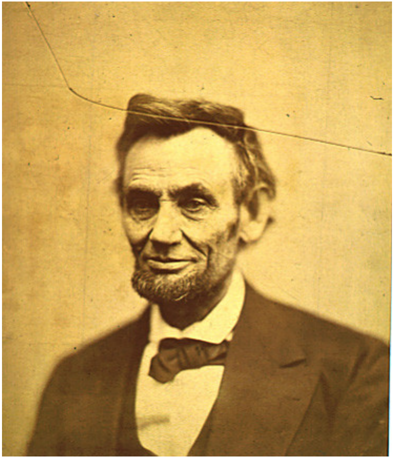


Image Inpainting

Inpainting (disocclusion) : How to fill missing regions in images ?
Should be done in a plausible way.

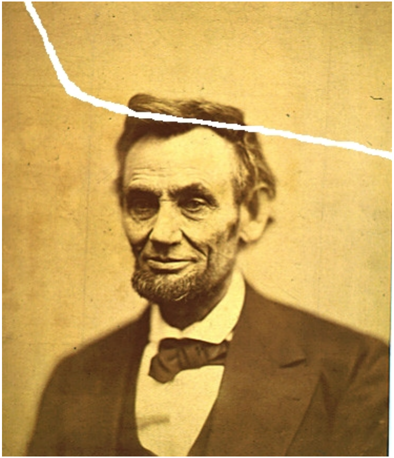


Image Inpainting

Inpainting (disocclusion) : How to fill missing regions in images ?
Should be done in a plausible way.

Image Inpainting

Inpainting (disocclusion) : How to fill missing regions in images ?
Should be done in a plausible way.

Source Bertalmio et al. 2000

Image Inpainting

Inpainting (disocclusion) : How to fill missing regions in images ?
Should be done in a plausible way.

Image Inpainting

Inpainting (disocclusion) : How to fill missing regions in images ?
Should be done in a plausible way.

Image Inpainting

Inpainting (disocclusion) : How to fill missing regions in images ?
Should be done in a plausible way.

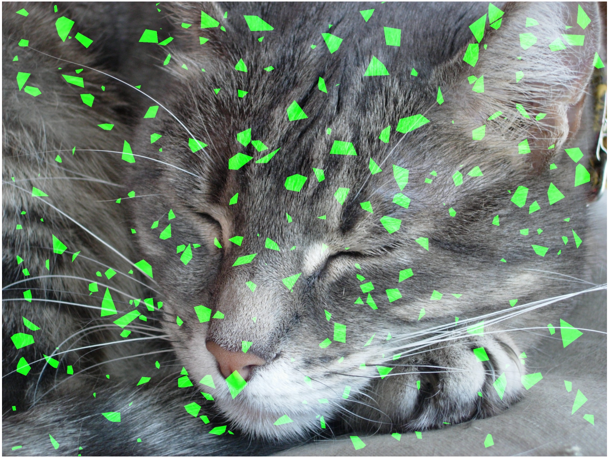
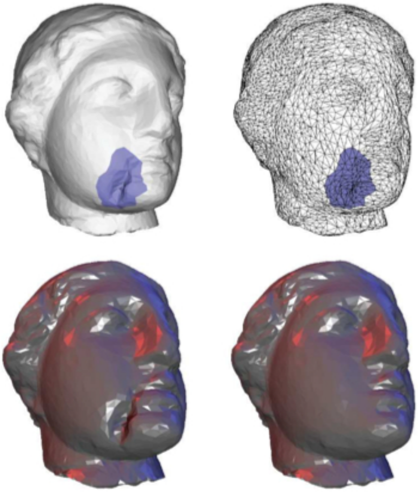


Image Inpainting

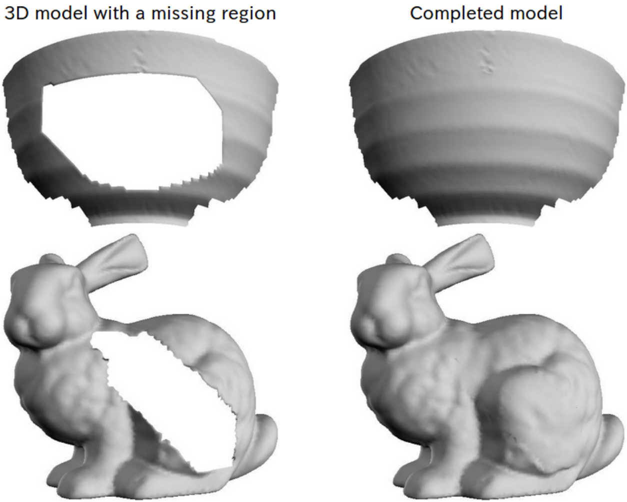
Inpainting (disocclusion) : How to fill missing regions in images ?
Should be done in a plausible way.

3D inpainting



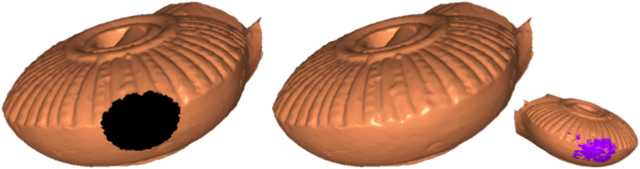
(Bobenko, Schroder, 2005)

3D inpainting



(Kawai, Sato, Yokoya, 2009)

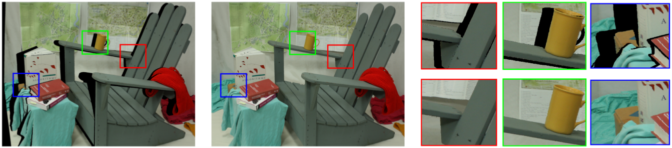
3D inpainting



(a) hole (b) our completion

(Harary et al., 2014)

Virtual view synthesis



(Buyssens et al., 2015)

Video inpainting

Inpainted video

Video inpainting

Original video

Historical example

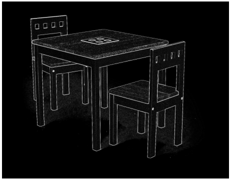
The commissar vanishes (from www.newseum.org)

Applications

- Image and video editing
- Video post-production, visual effects
- Restoration of old materials (photographs and movies)
- Zoom, super-resolution, deinterlacing
- Multi-image restoration (moving objects)
- Etc.

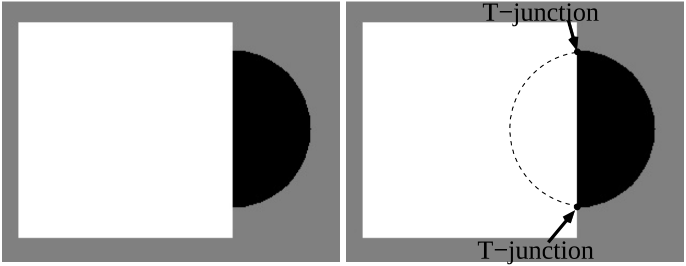
Human visual system and occlusions

Objects are (mostly) opaque $\rightarrow$ most objects are only partially visible !



Our visual system is able to infer missing parts by amodal completion.

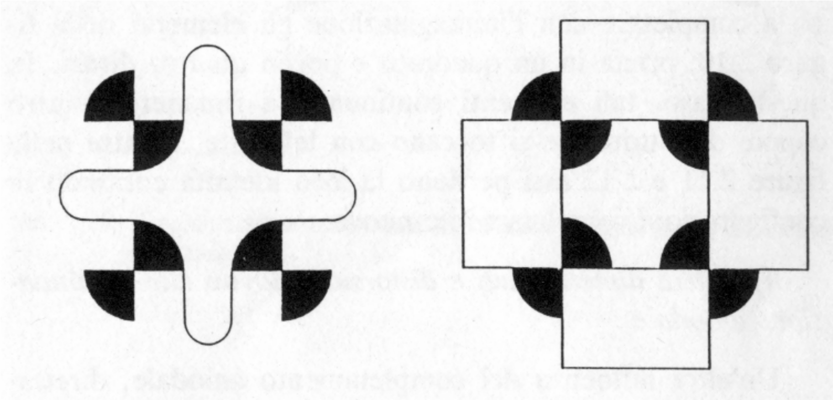
Amodal completion



Curves are interpolated smoothly between T-junctions

G. Kanizsa, Organization in Vision: Essays on Gestalt Perception, Pr¨ager, 1979

Amodal completion



Curves are interpolated smoothly between T-junctions

G. Kanizsa, Organization in Vision: Essays on Gestalt Perception, Pr¨ager, 1979

Amodal completion

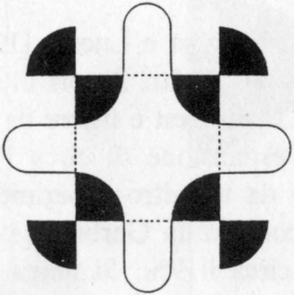


FIG. 2.13. Così si completa la figura 2.11.

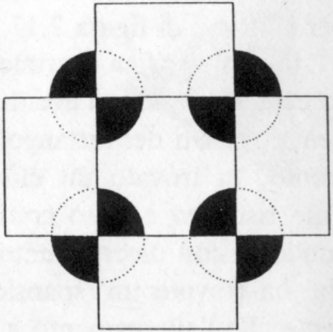


FIG. 2.14. Così si completa la figura 2.12.

Amodal completion

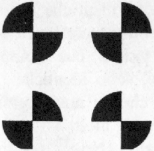


FIG. 2.10. Quattro coppie di settori neri. I vari completamenti mentalmente possibili non incidono sulla loro identità.

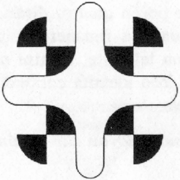


FIG. 2.11. Un quadrato con quattro appendici, come in figura 2.13.

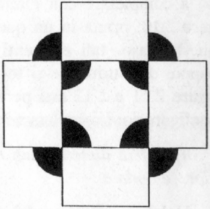
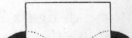
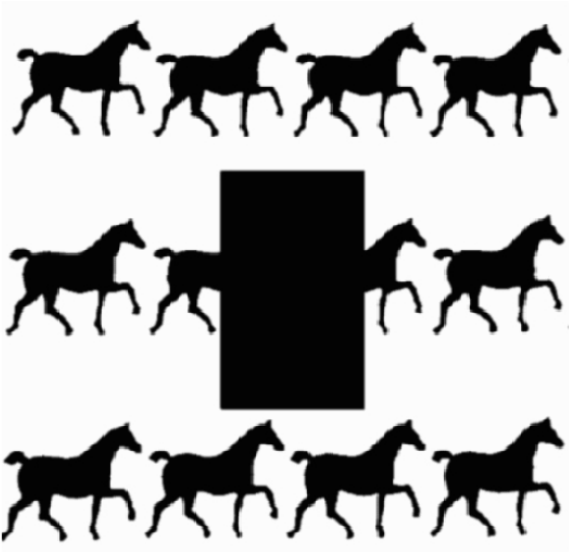


FIG. 2.12. Quattro dischi come in figura 2.14.





Amodal completion supersede common sense/previous knowledge !

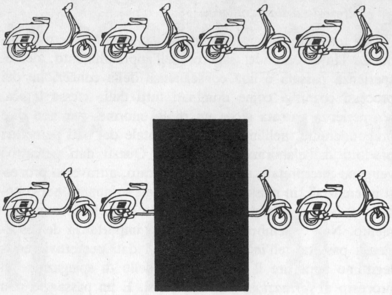


FIG. 2.29. Lo scooter «allungato».

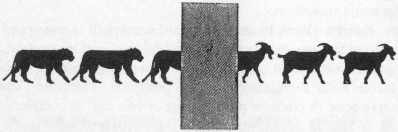
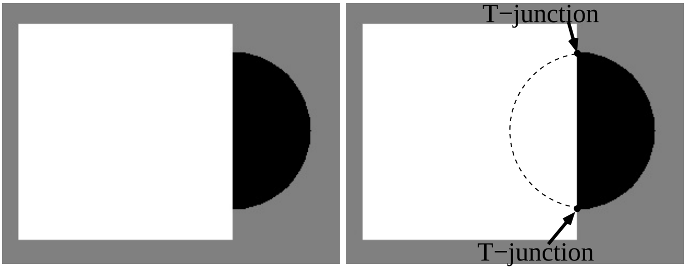


FIG. 2.30. Un ibrido creato dal completamento amodale.

First approach to inpainting (desocclusion) : Masnou-Morel 1998
Mimicks the human visual system

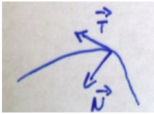


Virtual contour model = Euler elastica = argmin $\int_0^{\mathcal{L}} (\alpha + \beta|\kappa|^2)ds$
with boundary constraints of order 1

Express differential geometry reminder

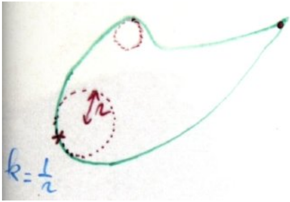
- $C : [a, b] \to \mathbb{R}^2$ is a **Jordan curve**
  if $C(p_1) \neq C(p_2)$ for $p_1 \neq p_2$.

- the choice of the parametrization C is of course not unique
- $L(a, p)$: length of C between a and p
  parametrization is called **arc-length parametrization** if
  $\frac{dL}{dp} = 1$
- If C is twice differentiable and $C'(p) \neq 0$
  the tangent vector is defined as $\overrightarrow{T} = \frac{C'(p)}{|C'(p)|}$
  the normal vector $\overrightarrow{N}$ is such that $(\overrightarrow{T}, \overrightarrow{N})$ is a direct orthonormal basis

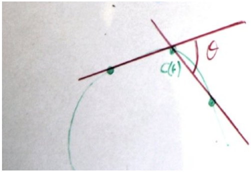


- **curvature**
  $\exists k$ such that
  $$\frac{1}{|C'|}\frac{d\overrightarrow{T}}{dp} = k\overrightarrow{N}$$
  and k is independent of the parametrization
  $k\overrightarrow{N}$ is called the curvature vector
- For an arc-length parametrization :
  $$\overrightarrow{T} = C'(s),$$
  $$\frac{d\overrightarrow{T}}{ds} = k\overrightarrow{N} = C''(s)$$
  (because $L(a,p) = \int_a^p |C'(u)|du$, so that $|C'(s)| = 1$.
- writing $C(p) = (x(p), y(p))$, we get
  $$k = \frac{y''x' - x''y'}{(x'^2 + y'^2)^{3/2}}$$

- the curvature satisfies $k(p) = r(p)^{-1}$, where $r(p)$ is the radius of the circle that best approximate the curve at $C(p)$ (osculating circle)



- In practice, one can approximate the curvature by the difference between two consecutive direction of the tangent vector (more robust than direct second order derivatives).



$$k \approx \frac{\theta}{\Delta s}$$

From curves to images: using the level set framework

Level lines of a (gray level) image are lines of constant intensity or, "equivalently", the boundaries of $\{x,\ u(x) \geq t\}$.

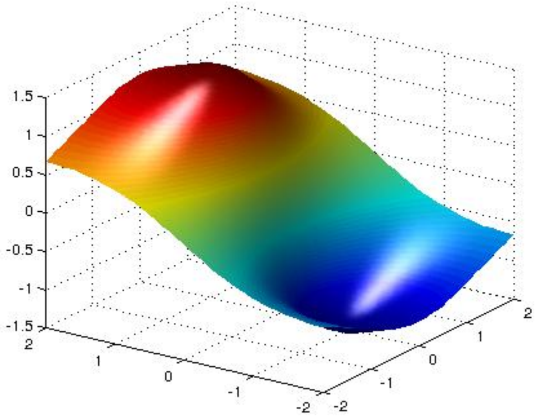


Figure: Graph of $f(x,y) = \frac{3y}{x^2+y^2+1}$

From curves to images: using the level set framework

Level lines of a (gray level) image are lines of constant intensity or, "equivalently", the boundaries of $\{x,\ u(x) \geq t\}$.

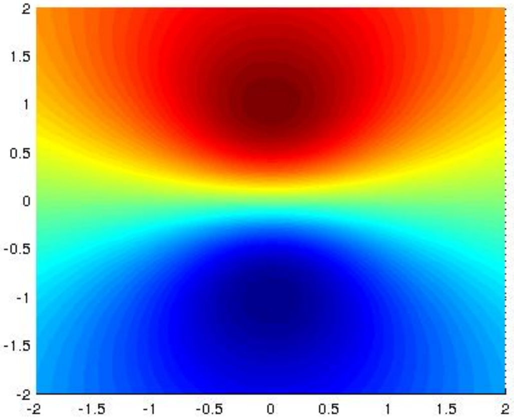


Figure: Graph of $f(x,y) = \frac{3y}{x^2+y^2+1}$ (viewed from above)

From curves to images: using the level set framework

Level lines of a (gray level) image are lines of constant intensity or, "equivalently", the boundaries of $\{x,\ u(x) \geq t\}$.

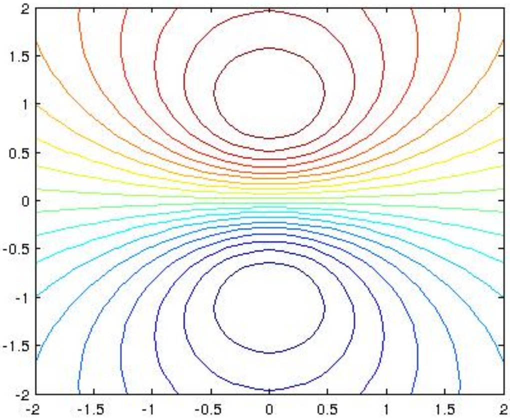


Figure: Some level lines

From curves to images: using the level set framework

Set of lines : "the topographic map" of the image

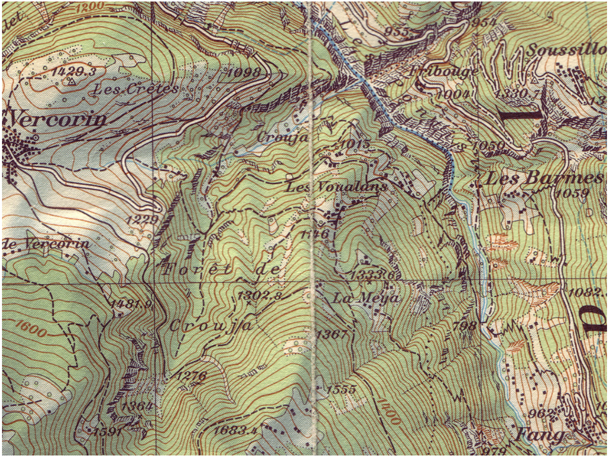


Figure: A topographic map

From curves to images: using the level set framework

Set of lines : "the topographic map" of the image

Adaptation to inpainting: using the level set framework (Masnou-Morel 1998)

Level sets $X_t^u = \{y : u(y) \geq t\} \Longleftrightarrow u(x) = \sup\left\{t : x \in X_t^u\right\}$

Level lines = Boundaries of level sets

Level lines reconstruction $\Longleftrightarrow$ Image restoration

Adaptation to inpainting: using the level set framework (Masnou-Morel 1998)

Level sets $X_t^u = \{y : u(y) \geq t\} \Longleftrightarrow u(x) = \sup\left\{t : x \in X_t^u\right\}$

Level lines = Boundaries of level sets

Level lines reconstruction $\Longleftrightarrow$ Image restoration

Adaptation to inpainting: using the level set framework (Masnou-Morel 1998)

Level sets $X_t^u = \{y : u(y) \geq t\} \iff u(x) = \sup\left\{t : x \in X_t^u\right\}$

Level lines = Boundaries of level sets

Level lines reconstruction $\iff$ Image restoration

Adaptation to inpainting: using the level set framework (Masnou-Morel 1998)

$$\int_{-m}^{+M}\Big(\sum_{\substack{\text{Paired}\\ \text{junctions}}} \int (\alpha + \beta|\kappa_{X_t^u}|^p)\, d\mathcal{H}^1\Big)\, dt \iff \int |\nabla u|\Big(\alpha + \beta\Big|\mathrm{div}\,\frac{\nabla u}{|\nabla u|}\Big|^p\Big)\, dx$$

Minimization over collections of curves

$\Downarrow$

Constructive approach

Classical minimization

$\Downarrow$

Global approach

Many works have followed :

- variational/PDE approaches
  (Masnou-Morel 1998, Chan-Shen 2001, Caselles et al. 2001, Bertalmio et al 2001, Tschumperlé-Deriche 2004, Bornemann and Märtz 2007, Schönlieb - Bertozzi 2011, Chizhov et al. 2021, etc.)
- examplar-based, patch-based
  (Efros-Leung 1999, Wei-Levoy 2000, Efros-Freeman 2001, Ashikmin et al 2001, Harrison 2001, Criminisi-Pérez-Toyama 2004, Pérez-Gangnet-Blake 2004, de Bonet 1997, Igehy-Pereira 1997, Komodakis 2007, Kawai et al. 2009, Arias et al. 2011, Liu-Caselles 2013, Wang 2013, Newson et al. 2014, Daisy et al. 2015, etc.)
  Two main trends:
  - greedy (sequential)
  - global, patch-based optimization (parallel)
- inpainting in transform domains
  (Elad et al. 2005, Chan et al. 2006, Fadili et al. 2007, Cai 2008)
- Convolutional neural networks
  (Pathak et al. 2016, Iizuka et al. 2017, Yu et al. 2018, 2019, Liu et al. 2018, Nazeri et al. 2019, Yi et al. 2020, Saharia et al. 2021, Lugmayr et al. 2022, etc.)

Variational/PDE methods

Simplest approach : heat equation
Image I and hole (occlusion) Ω

$$\frac{\partial I}{\partial t} = \Delta I \qquad \text{inside } \Omega$$

and

$$I = I_0 \qquad \text{outside } \Omega$$

Information is propagated by averaging :

Blurred results

In a discrete setting :

$$\Delta(u)(i,j) \approx u(i{+}1,j){+}u(i{-}1,j){+}u(i,j{+}1){+}u(i,j{-}1){-}4u(i,j)$$

$$u^{n+1}(i,j) - u^n(i,j) = \delta t \Delta u^n(i,j)$$

$$u^{n+1}(i,j) = (1-5\delta t)u^n(i,j)$$

$$+\delta t\left(u^n(i+1,j) + u^n(i-1,j) + u^n(i,j+1) + u^n(i,j-1) + u^n(i,j)\right),$$

$\rightarrow$ local smoothing

Variational/PDE methods

Bertalmío, Sapiro, Caselles, Ballester (2000)

- Introduce the term "inpainting"
- Evolution equation :

$$\frac{\partial u}{\partial t} = \nabla \Delta u \cdot \nabla^{\perp} u$$

(+anisotropic diffusion for stabilization)

- Idea : a measure of smoothness (Δu) is "transported" along the isophotes directed by $\nabla^{\perp} u$
  by analogy with a transport equation $\frac{\partial u}{\partial t} = -\text{div}(u\vec{v})$,
  where $\vec{(v)}$ is the speed. If $\vec{v}$ is constant, then

$$\frac{\partial u}{\partial t} = -\nabla(u).\vec{v}$$

- Efficient for small and non-textured occlusions
- Many variants and follow-up
  (see e.g. *Partial Differential Equation Methods for Image Inpainting*, Schoenlieb, 2016)

From Bertalmio et al. 2000

From Bertalmio et al. 2000

Variational/PDE methods

Alternative: the denoising viewpoint

- Chan, Shen (2001) (Total variation)

$$\int_A |\nabla u| dx + \frac{\lambda}{2} \int_\Omega^c |u - u_0|^2 dx,$$

  A being the image domain

- Chan, Kang, Shen (2002) & Esedoglu, Shen (2002) (Mumford-Shah-Euler)

$$\int_{\Omega \setminus A} |u - u_0|^2 \, dx + \int_{\Omega \setminus K} |\nabla u|^2 dx + \int_K (\alpha + \beta \, k^2) ds.$$

- and many, many other contributions (higher-order inpainting, topological analysis, fractional-order inpainting, etc.) !

Variational/PDE methods

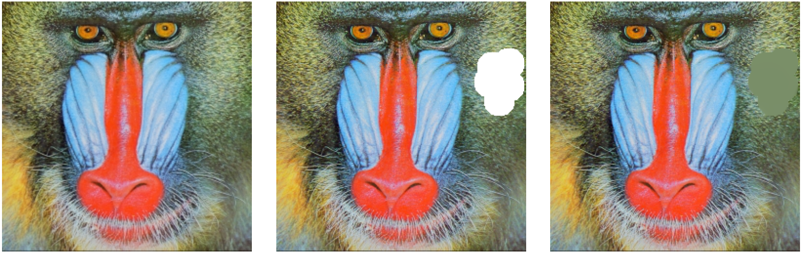
Advantages : fast, mathematical interpretation (strong geometrical property)

Main limitation : no texture

15×15 patches are removed (from Masnou et al. 2011)

Variational/PDE methods

Advantages : fast, mathematical interpretation (strong geometrical property) Main limitation : no texture



Large inpainting using the Total Variation

Nonlocal methods : from texture synthesis to inpainting

- The patch-based texture synthesis method of Efros and Leung (that we saw in the texture synthesis lecture) can be straightforwadly applied to the inpainting problem
- Many patch-based methods followed from the 2000's

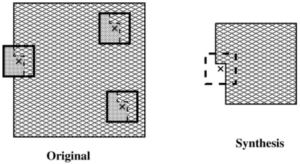
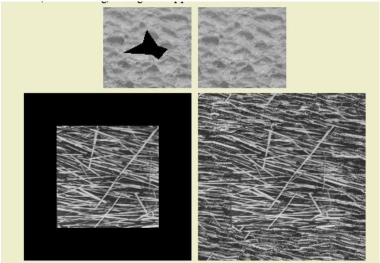


FIG. 1. *The nonparametric resampling algorithm.*



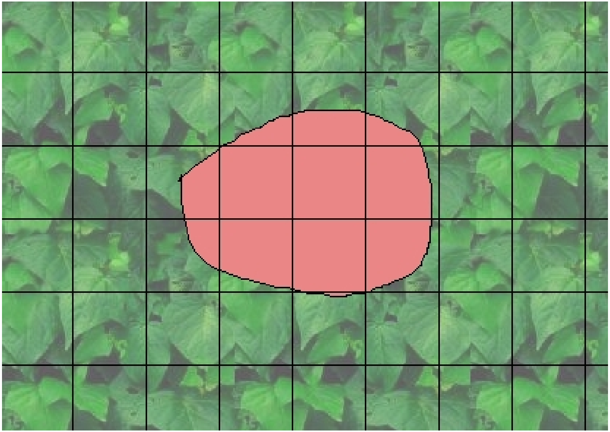
Ill. Levina-Bickel 2006 - Efros-Leung 1999

Application to inpainting

Many papers, and many methods :

- Drori et al. 2003 (multiscale sampling)
- Criminisi et al. 2004, Pérez et al 2014 (greedy approach, priority order for the filling-in) → next slides
- Sun et al. 2005 (user-assisted method to help the recovery of geometric structures)
- Wexler et al. 2005, Newson et al 2017 (global patch-based energy, heuristic for the minimisation) → second part of the lecture
- Komodakis et al. 2007 (variational and patch-based strategy, minimization with belief propagation)
- Cao et al. 2011 (patch-based strategy with automatic geometrical guide)
- Arias et al. 2011 (variational framework for non-local patch-based inpainting)
- Liu-Caselles 2013 (multi-scale graph-cut)
- and a lot more...

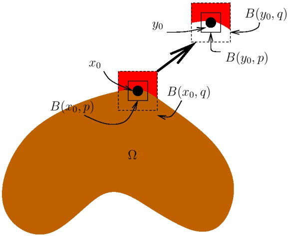
→ synthèse par "patchs" (Efros-Freeman 2000, Pérez-Gangnet-Blake 04)



Principe (Pérez-Gangnet-Blake 04)

Soit Ω la région à reconstruire, $0 < p < q$ deux paramètres.
Soit $B(x,p)$ le patch centré sur x de rayon p et
$C(x,p,q) = B(x,q) \setminus B(x,p)$.

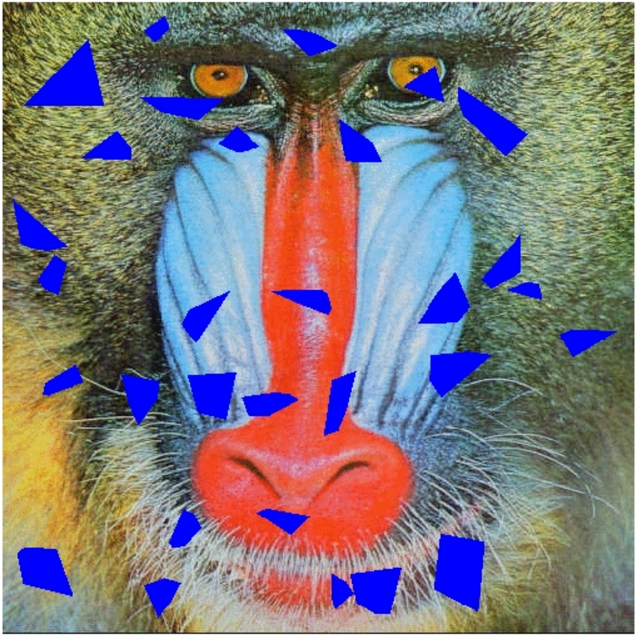
- 1) soit $x_0 \in \partial\Omega$ ayant un nombre de voisins maximum dans Ω^c.
- 2) soit $y_0 \in \Omega^c$ qui minimise la norme L^2 entre $C(x_0,p,q) \setminus \Omega$ et $C(y_0,p,q) \setminus (\Omega + y_0 - x_0)$.
- 3) pour chaque $x \in B(x_0,p) \cap \Omega$ soit $I(x) = I(x + y_0 - x_0)$.
- 4) remplacer Ω par $\Omega \setminus B(x_0,p)$ et itérer.



Nombreuses variantes de cet algorithme (choisir au hasard un patch proche, injecter de l'invariance en considérant des rotations des patchs, etc.)

Les résultats dépendent de

1. paramètres p, q
   en général préférable de choisir $p > 0$ (patchs au lieu de pixels)
   p grand $\rightarrow$ meilleur respect de la géométrie, moins d'"innovation"
   q grand $\rightarrow$ meilleures transitions entre patchs
2. ordre de remplissage:
   quel $x_0 \in \partial\Omega$ choisir à chaque itération ?
   [Criminisi-Pérez-Toyama '04] : x_0 minimise une fonctionnelle dépendant de
   i) la géométrie de Ω,
   ii) la géométrie des lignes de niveau autour de x_0.



Application to inpainting

Application to inpainting

Application to inpainting

Application to inpainting

However these methods may fail at reconstructing long-range geometric features, e.g. long edges

Different approaches have been proposed for the simultaneous restoration of geometry and texture (Bertalmio et al '03, Fadili-Starck '05, Tschumperlé et al. 2006, Cao et al. 2011, etc.)

Usually rely on hybrid approaches

$\Longrightarrow$ Possible solution : use a geometric guide computed on a simplified image

Geometrically guided exemplar-based inpainting (Cao et al. 2011)

Step 1: compute a geometric sketch

Step 2: restore the geometric sketch by interpolation of the level lines with Euler spirals

Euler elasticae are solutions of

$$\mathrm{Min} \int_0^L (1 + |\Psi'(s)|^2) ds \qquad (\Psi = \text{angle(tangent,horizontal axis)})$$

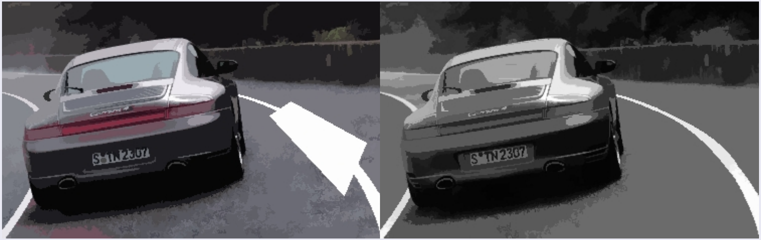
under endpoints and end-tangents constraints.

Euler-Lagrange equation: $(\Psi')^2 = 1 + \lambda \cos \Psi + \mu \sin \Psi$.

After linearization $(\Psi')^2 = 1 + \lambda + \mu\Psi$ whose solutions are Euler spirals:

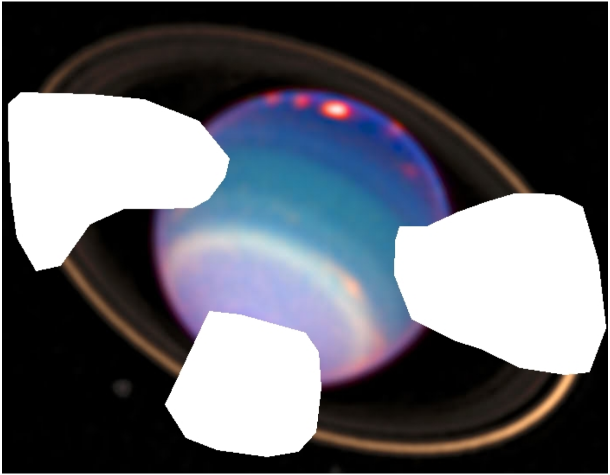
- Curvature= affine function of arc-length
- Very useful in civil engineering, industrial design, typography. Also used as a model for shape completion in vision.

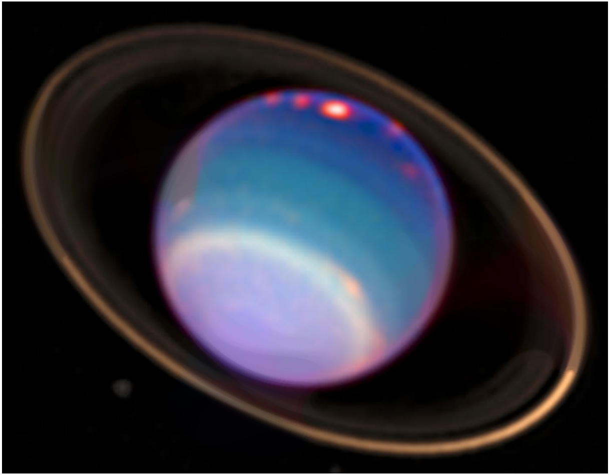
Step 2: restore the geometric sketch by interpolation of the level lines (e.g. with Euler spirals)



Step 3: use the reconstructed sketch as a geometric guide

New metric between patches $=$ Linear combination of a L^2 metric on the original image (conditioned by the inpainting domain) and a L^2 metric on the (complete) geometric sketch (Many possible variants)





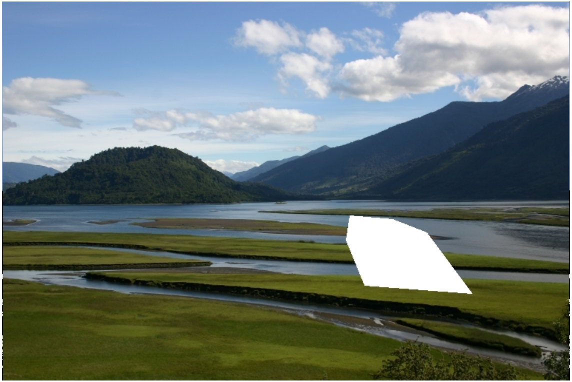


Image with missing region

Method from Cao et al. ’11

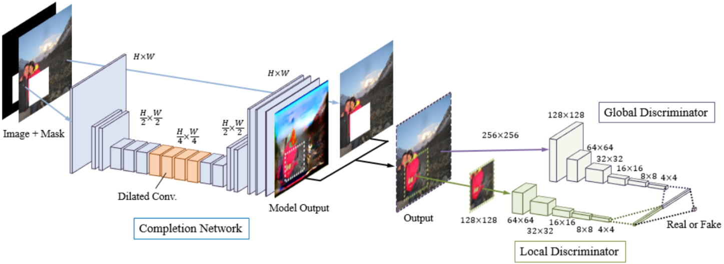
Global optimization-based approaches

- Initiated by the works of Demanet et al. 2003, Wexler et al. 2005
- The missing region is reconstructed by stiching patches from the images, whose coherence is ensured by an iterative approach
- $\rightarrow$ texture $+$ relatively good global geometric coherence (best approaches to date without learning)
- $\rightarrow$ detailed for *video inpainting* in the second part of the course

And of course ... CNNs

Many approaches developped from Convolutional Neural Networks (CNN) (Pathak et al. 2016, Iizuka et al. 2017, Yang et al. 2017, Liu et al. 2018, 2019, Yi el al. 2020, Suvorov et al. 2022, Lugmayr et al. 2022, etc.)

- Use ideas from autoencoders, Generative Adversarial Networks (Goodfellow et al. 2014) or more recently diffusion models
- Implicitely use information not from the inpainted image (this was sometimes done explicitely before, see Hays-Efros 2007)
- Training can involve several millions images and weeks of computation



Global architecture of the method from Iizuka et al. 2017

From Iizuka et al. 2017

From Iizuka et al. 2017

Typically outside the reach of patch-based methods

ccn-based method ; the image is aligned, as in the training dataset

after a 10 pixels translation

patch-based method

Experiment courtesy of E. Bonnail

May yield artefacts

May yield artefacts

Contextual attention (DeepFill, Yu et al. 2018)

Hybrid method (CNN / patch-based) Take into account patches near the missing region at training time

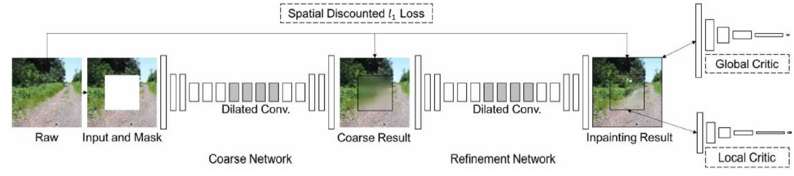


Figure 2: Overview of our improved generative inpainting framework. The coarse network is trained with reconstruction loss explicitly, while the refinement network is trained with reconstruction loss, global and local WGAN-GP adversarial loss.

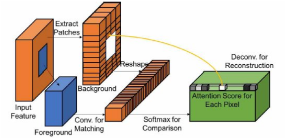
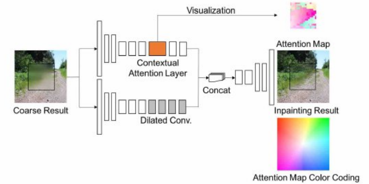


Figure 3: Illustration of the contextual attention layer.

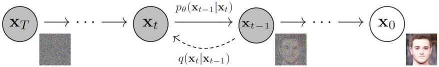


Illustrations from Yu et al. 2018

Other evolutions

- Edge connect (Nazeri et al. 2019)
  Learn a sketch reconstruction component
- Free-form image inpainting (DeepFill v2, Yu et al. 2019)
  Use of Gated convolution
- Contextual residual aggregation (Yi et al. 2020)
- Local inpainting in the Fourier domain(LAMA, Suvorov et al. 2022)
- Diffusion models (REPAINT Lugmayr et al. 2022, PALETTE Saharia et al. 2022, latent diffusion models, aka stable diffusion, Rombach et al 2022, text-guided inpainting, Smartbrush 2023, etc.)

Diffusion models for inpainting



- Rely on Denoising Diffusion Probabilistic Models (DDPM)
- DDPMs generate images by progressive denoising of a noise input (Sohl-Dickstein et al. 2015, Ho et al. 2020)
- "Denoising" rely on a CNN trained to reverse the following process

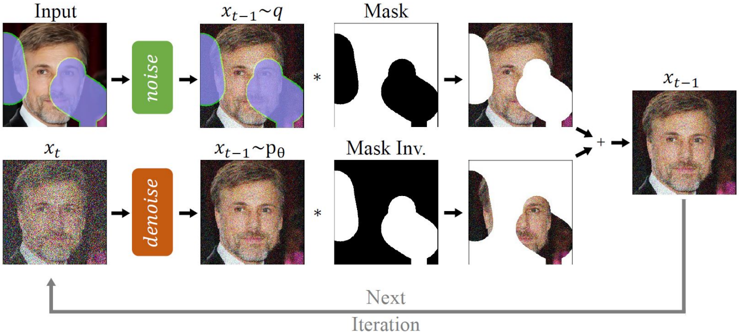
$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$$

- More precisely, the network is trained to learn μ_θ and Σ_θ of the process

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t,t), \Sigma_\theta(x_t,t))$$

REPAINT, Lugmayr et al. 2022

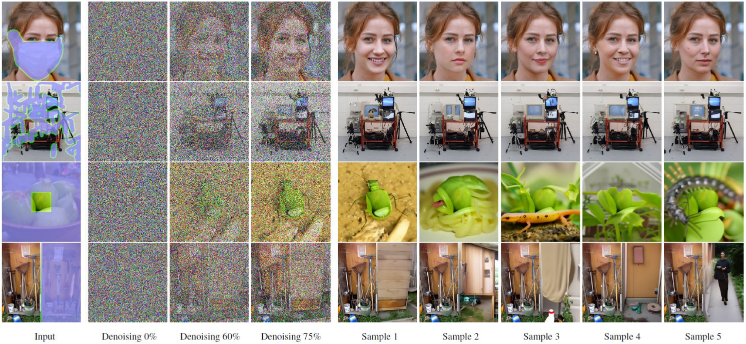
- The framework is adapted to the inpainting task



$$x_{t-1}^{\text{known}} \sim \mathcal{N}(\sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t)\mathbf{I})$$
$$x_{t-1}^{\text{unknown}} \sim \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$
$$x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}$$

REPAINT, Lugmayr et al. 2022

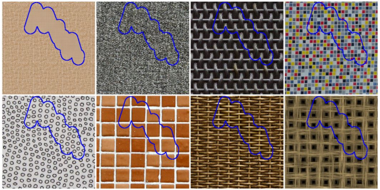
- enables unprecedented quality and diversity



REPAINT, Lugmayr et al. 2022

Reasonable generalization capacity

- trained on a relatively generic scene database (Places2)



- trained on a face database (CelebA-HQ)

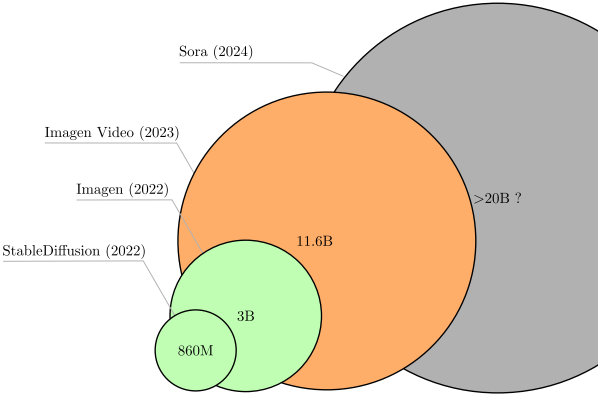
Experiments courtesy of N. Cherel

REPAINT, Lugmayr et al. 2022

But rely on huge networks ...

- about 500M parameters
- memory impact is about 3GB for 256x256 images
- heavy environmental impact of the training stage :
  for celebA-HQ (30000 images):
  500h + training time, about 10kg CO2

And it is getting bigger and bigger !

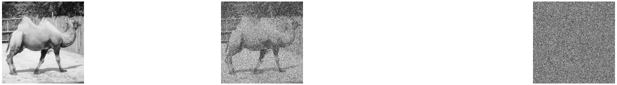


Strong need for frugal or at least lightweight approaches

A possible solution : *internal approaches*
The model is learned for the image/video at hand.

Framework from Denoising Diffusion Probabilistic Models [1].

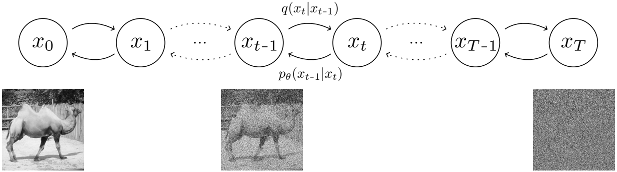
$$x_0 \rightarrow x_1 \cdots x_{t-1} \xrightarrow{q(x_t|x_{t-1})} x_t \cdots x_{T-1} \rightarrow x_T$$



Forward process:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(\sqrt{1-\beta_t}x_{t-1}, \beta_t I\right)$$

Framework from Denoising Diffusion Probabilistic Models [1].



Forward process:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(\sqrt{1-\beta_t}x_{t-1}, \beta_t I\right)$$

Backward process:

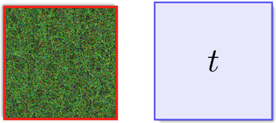
$$p_\theta\left(x_{t-1}|x_t\right) = \mathcal{N}(\mu_\theta(x_t, t), \sigma_t^2 I)$$

Problem setting 2/2

Neural network f_θ is used to predict the mean of $p_\theta(x_{t-1}|x_t \quad)$ and is optimized for a denoising L2 loss.

$$\mathbb{E}_{x_0, x_t, t}\left[w(t)\|x_0 - f_\theta(x_t, t)\|_2^2\right]$$

Where x_t is the noisy image.



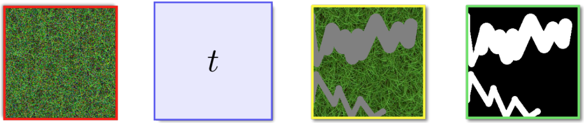
x_t t

Problem setting 2/2

Neural network f_θ is used to predict the mean of $p_\theta(x_{t-1}|x_t, y)$ and is optimized for a denoising L2 loss. For image inpainting, we have additional inputs:

$$\mathbb{E}_{x_0, x_t, t, M}\left[w(t)\|x_0 - f_\theta(x_t, y, M, t)\|_2^2\right]$$

Where x_t is the noisy image. $y = x \circ (1 - M)$ is the clean masked image, M the mask.



x_t $\quad$ t $\quad$ y $\quad$ M

Network architecture

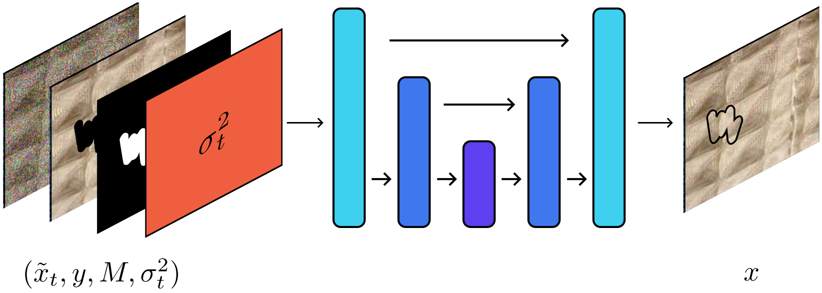
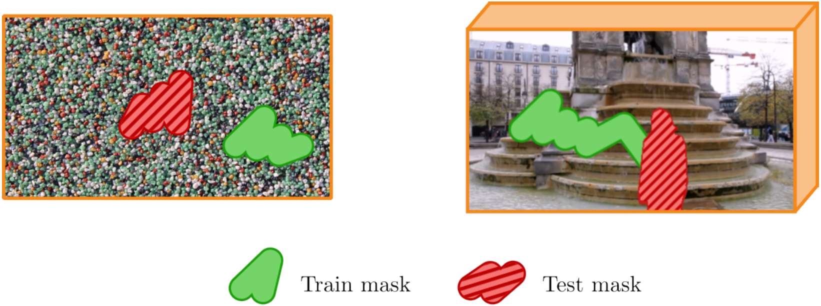


Figure: UNet with 160k parameters for image inpainting

Internal learning



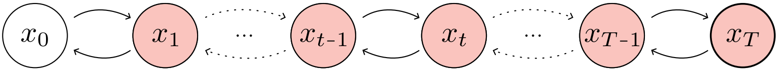
Baseline training

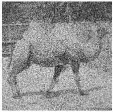
Training

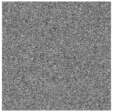
repeat
  $x_0 \sim q(x_0)$, $t \sim \mathcal{U}([1, T])$
  $x_t \sim q(x_t | x_0)$
  Take gradient descent step on
   $\nabla_\theta \| x_0 - f_\theta(x_t, t) \|^2$
until converged

Inference

$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $x_{t-1} \sim \mathcal{N}\left(\mu_\theta(x_t, t), \sigma_t^2 \mathbf{I}\right)$
end for
return x_0







Baseline training

Training

repeat

  $x_0 \sim q(x_0)$, $t \sim \mathcal{U}([1,T])$

  $x_t \sim q(x_t|x_0)$

  Take gradient descent step on

   $\nabla_\theta \|x_0 - f_\theta(x_t, t)\|^2$

until converged

Inference

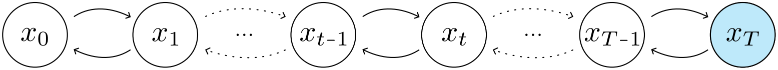
$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

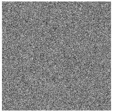
for $t = T, \ldots, 1$ **do**

  $x_{t-1} \sim \mathcal{N}\left(\mu_\theta(x_t, t), \sigma_t^2 \mathbf{I}\right)$

end for

return x_0





Interval training

Training for interval i

repeat

$x_0 \sim q(x_0)$, $t \sim \mathcal{U}([\tau_{i+1}, \tau_i])$

$x_t \sim q(x_t|x_0)$

Take gradient descent step on

$\nabla_\theta \|x_0 - f_\theta(x_t, t)\|^2$

until converged

Inference for interval i

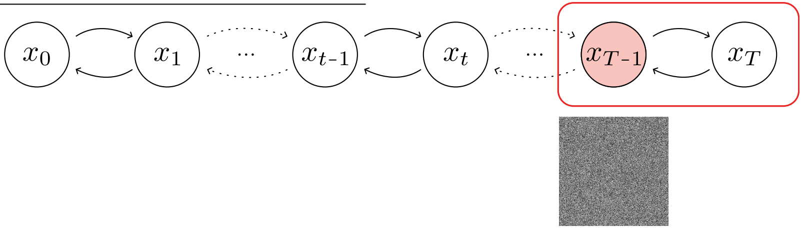
$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

for $t = \tau_i, \ldots, \tau_{i+1}$ **do**

$x_{t-1} \sim \mathcal{N}\left(\mu_\theta(x_t, t), \sigma_t^2 \mathbf{I}\right)$

end for

return x_0



Interval training

Training for interval i

repeat

$x_0 \sim q(x_0)$, $t \sim \mathcal{U}([\tau_{i+1}, \tau_i])$

$x_t \sim q(x_t|x_0)$

Take gradient descent step on

$$\nabla_\theta \|x_0 - f_\theta(x_t, t)\|^2$$

until converged

Inference for interval i

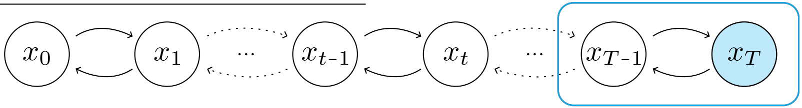
$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

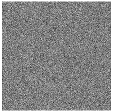
for $t = \tau_i, \ldots, \tau_{i+1}$ **do**

$x_{t-1} \sim \mathcal{N}\left(\mu_\theta(x_t, t), \sigma_t^2 \mathbf{I}\right)$

end for

return x_0





Interval training

Training for interval i

repeat

$x_0 \sim q(x_0)$, $t \sim \mathcal{U}([\tau_{i+1}, \tau_i])$

$x_t \sim q(x_t | x_0)$

Take gradient descent step on

$$\nabla_\theta \| x_0 - f_\theta(x_t, t) \|^2$$

until converged

Inference for interval i

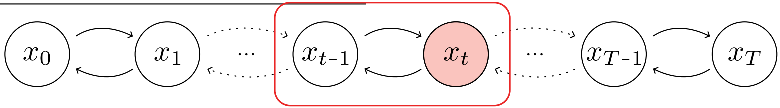
$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

for $t = \tau_i, \ldots, \tau_{i+1}$ **do**

$x_{t-1} \sim \mathcal{N}\left(\mu_\theta(x_t, t), \sigma_t^2 \mathbf{I}\right)$

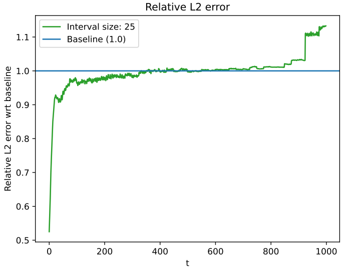
end for

return x_0



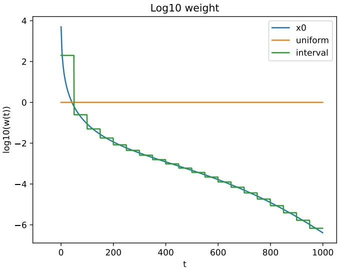
Interval training

+ model specialized for each inference phase
+ remove weighting in the loss
− single use



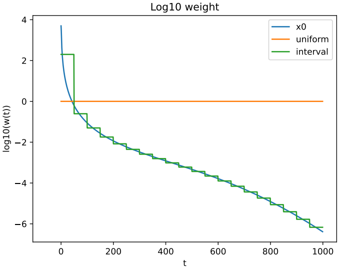
Interval training

+ model specialized for each inference phase
+ remove weighting in the loss
– single use



Interval training

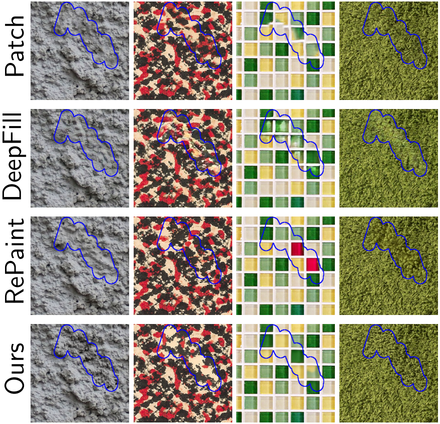
+ model specialized for each inference phase
+ remove weighting in the loss
− single use



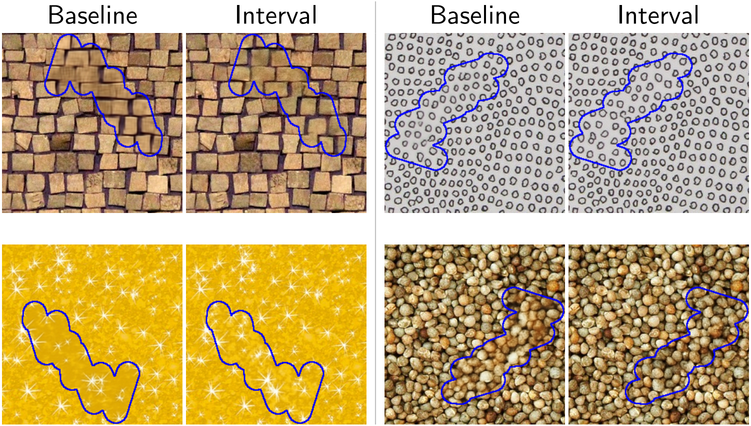
Texture inpainting

Comparison with:

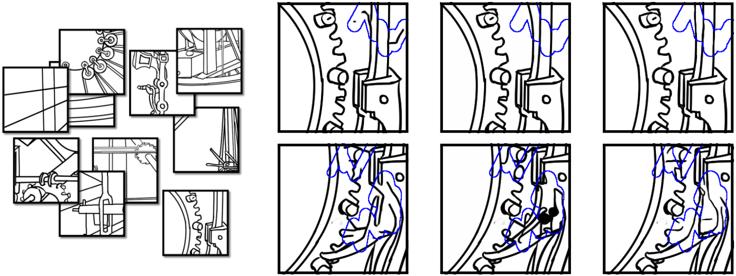
- Patch-based method of Newson *et al.*(2017) [2]
- DeepFill: inpainting network with attention (2018) [3]
- RePaint: large diffusion model (2022) [4]



Interval training - results

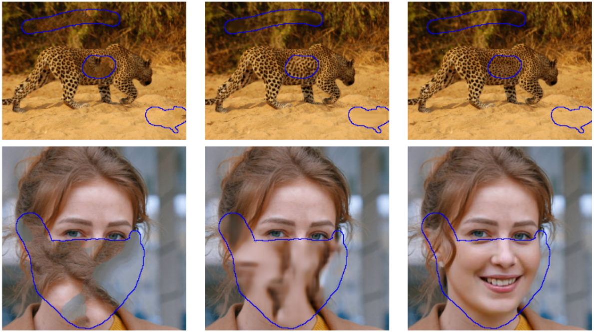


Line drawing



Train set | Patch | RePaint | Ours

Non-stationnary images



Patch Ours RePaint

The method is unable to create new content and to infer completely unseen structures.

Works also for videos ... exemples on next set of slides.