Image Completion with Structure Propagation

(or: Do NOT trust digital images [anymore]!)

(ACM SIGGRAPH 2005)

[Jian Sun, Lu Yuan, Jiaya Jia, Heung-Yeung Shum]

The things you have to hear:

- Introduction/Motivation:
  - In what context to put it
  - Related work | building blocks
  - That's cold coffee! (really?)
- Algorithmic Details:
  - Overall *usage*
  - The method in detail

... (cont.)

- Performance/comparison
  - Advantages/disadvantages
  - limits
- Conclusion
  - Applicability
  - Future use

Introduction/Motivation

- Context
  - 2D image reconstruction
    - No additional dimension, such as time.
  - Data sources: pictures; or, in a more general view: measurements from the real world.
- Related work
  - Image inpainting: *small gaps, thin structures*
  - Example-based approaches
  - Approaches with interaction

Cold coffee?

- The “usual” approach
  - Development over years, every time a little improvement
  - Several paths finally join to form an “exceptional” result
- This algorithm's base observations:
  - Only few well-defined curves are necessary
  - There exists an synthesis ordering

Is it really a milestone?

It is really a milestone?

The method's steps

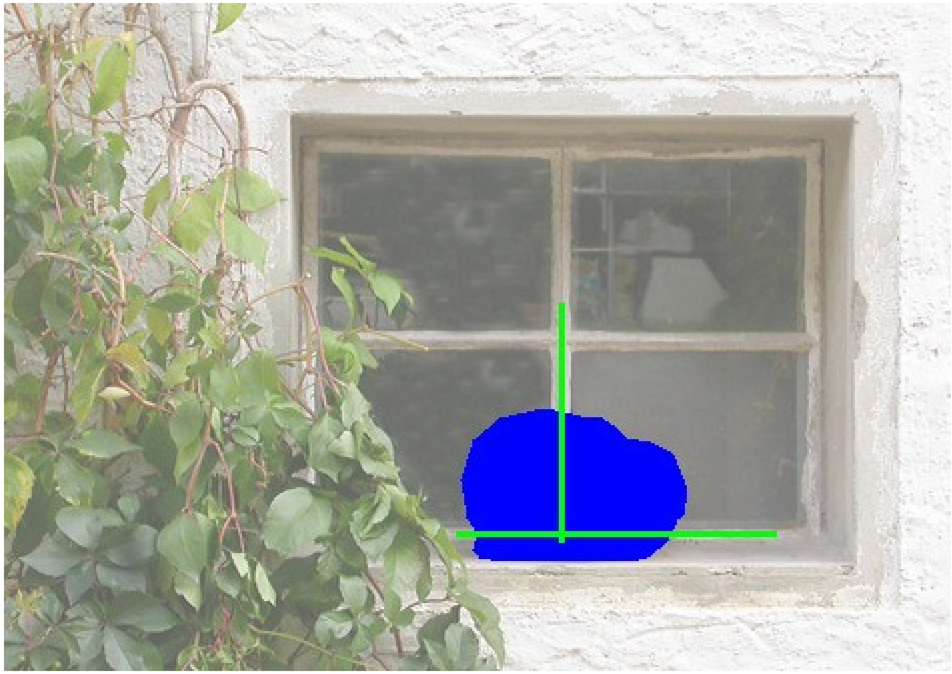
- User Interaction:
  - Giving some “hints” --> define curve for salient structures
- Structure Propagation:
  - Reconstruct curve in unknown region --> create missing part of salient structures
- Texture Propagation:
  - Fill the holes: perform texture synthesis in the unknown regions.

User Interaction

- Given: Image with bothering area:

User Interaction

- User selects object and defines edges:



Algorithmic process

- Structure Propagation:

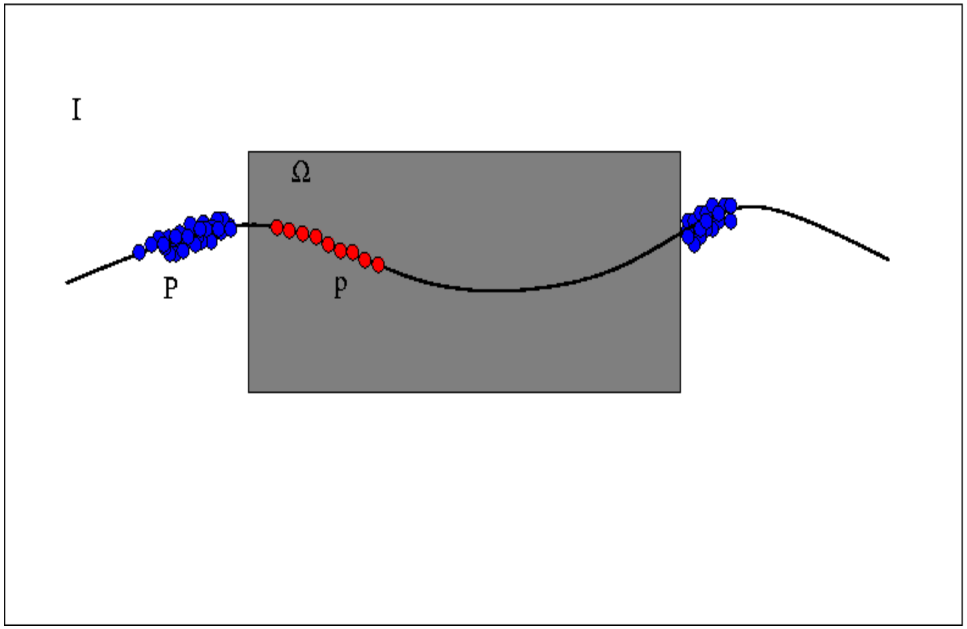
Algorithmic process

- Final result:

The algorithm's pillar no. 1

- Structure Propagation
  - Given an image with
    - Unknown region Ω
    - User-defined curve C
  - Sampling of
    - known region along user-defined curve
      - $P = \{P(1), P(2), \ldots, P(N)\}$
    - unknown region along the user-defined curve:
      - Anchor points $\{p_i\}_{i=1}^{L}$

Illustration



Graph $G=\{V,E\}$ --> labeling problem:

for each anchor point we want to find the label x_i out of {1,...,N} and "paste" the corresponding patch $P(x_i)$ at position p_i

Energy minimization

- We want to minimize the energy globally

$$E(X)=\sum_{i\in V} E_1(x_i)+\sum_{(i,j)\in E} E_2(x_i,x_j) \qquad X=\{x_i\}_{i=1}^{L}$$

$$E_1(x_i)=k_s\cdot E_S(x_i)+k_i\cdot E_I(x_i)$$

$$E_S(x_i)=d(c_i,c_{x_i})+d(c_{x_i},c_i)$$

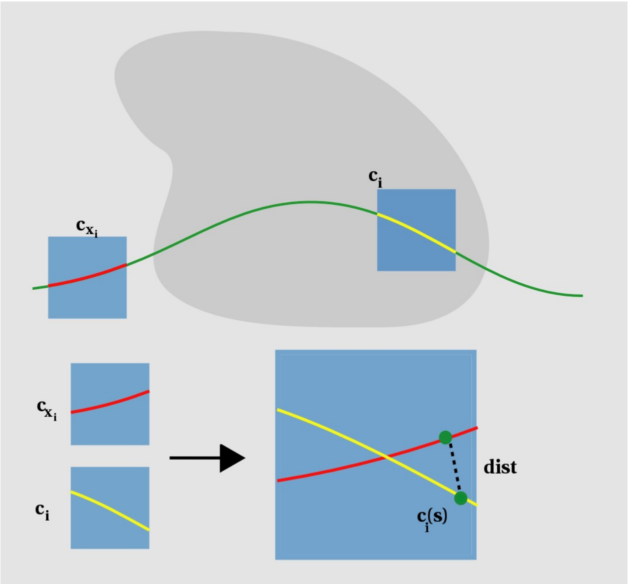
$$d(c_i,c_{x_i})=\frac{1}{N_s}\sum_s \|dist(c_i(s),c_{x_i})\|^2$$

$$E_I(x_i)=\frac{1}{N_{s'}}\sum_{s'}(diff(P(x_i)(s')-b_i(s')))^2$$

Energy minimization

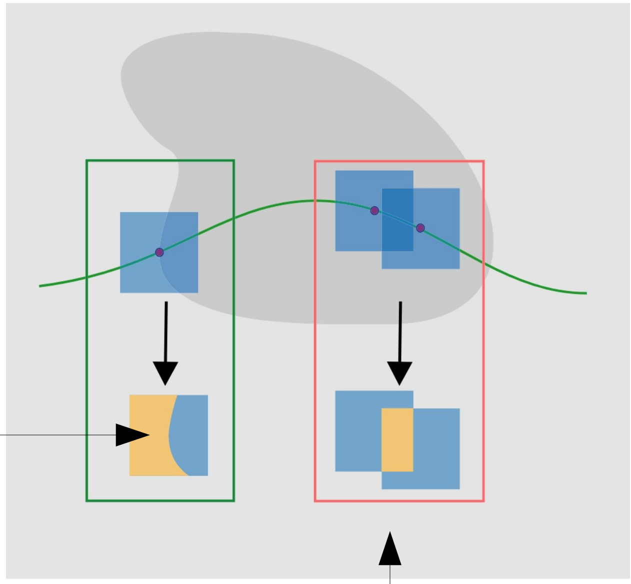
$$E_s(x_i) = d(c_i, c_{x_i}) + d(c_{x_i}, c_i)$$

$$d(c_i, c_{x_i}) = \sum_s \|dist(c_i(s), c_{x_i})\|^2$$



Energy minimization

$$E(X) = \sum_{i \in V} E_1(x_i) + \sum_{(i,j) \in E} E_2(x_i, x_j)$$

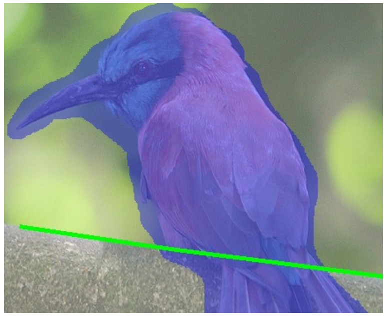


$E_I(x_i)$

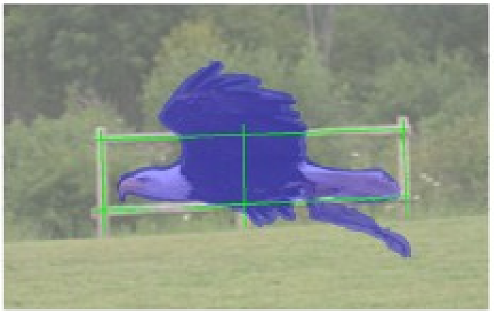
$E_2(x_i, x_j)$

Two approaches

- i *Dynamic Programming* for a single user defined curve



- ii *Belief Propagation* for multiple intersecting curves



Dynamic Programming

- Big table (remember Info II)
- Define the cumulative minimal cost from node 1 to node i for all possible x_i

$$M_i(x_i) = E_1(x_i) + min_{x_{i-1}} \{ E_2(x_{i-1}, x_i) + M_{i-1}(x_{i-1}) \}$$

- The optimal label of node L is then

$$x_L^* = argmin_{x_L} M_L(x_L)$$

Now the harder part:
multiple intersecting curves

- Belief Propagation
  - is a probability inference algorithm
  - uses message passing between two neighboring nodes
  - iterative procedure
  - uses only a neighborhood of each node in order to calculate minimal energy. But do this for a maximal iteration count of T where T = max. dist. between two nodes

Belief Propagation

- Initialization: $M_{ij}^{0} = \mathbf{0}$
- Iteration from t=1 to T:

$$M_{ij}^{t} = min_{x_i} \{ E_1(x_i) + E_2(x_i, x_j) + \sum_{k \neq j, k \in N(i)} M_{ki}^{t-1} \}$$

- Optimal label computation:

$$x_i^{*} = argmin_{x_i} \{ E_1(x_i) + \sum_{k \in N(i)} M_{ki}^{T} \}$$

Belief Propagation

iteration count

- $M_{ij}^{0} = \mathbf{0}$
- $M_{ij}^{t} = min_{x_i}\{E_1(x_i) + E_2(x_i, x_j) + \sum_{k \neq j, k \in N(i)} M_{ki}^{t-1}\}$
- $x_i^{*} = argmin_{x_i}\{E_1(x_i) + \sum_{k \in N(i)} M_{ki}^{T}\}$

local neighborhood

Grasping breath: where are we?

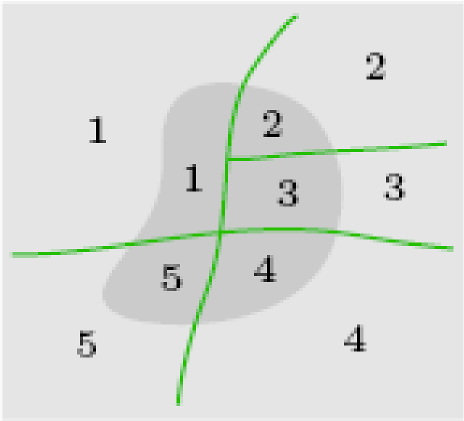
- The algorithm now has reconstructed the image along the user-defined curve
- Now we need to
  - Fill the "holes"
  - Do photometric correction

Texture Propagation

- Other approaches use the whole image as a "pool" for their texture synthesis...

  --> irrelevant information may gain importance...

- This algorithm is brighter:



Photometric correction

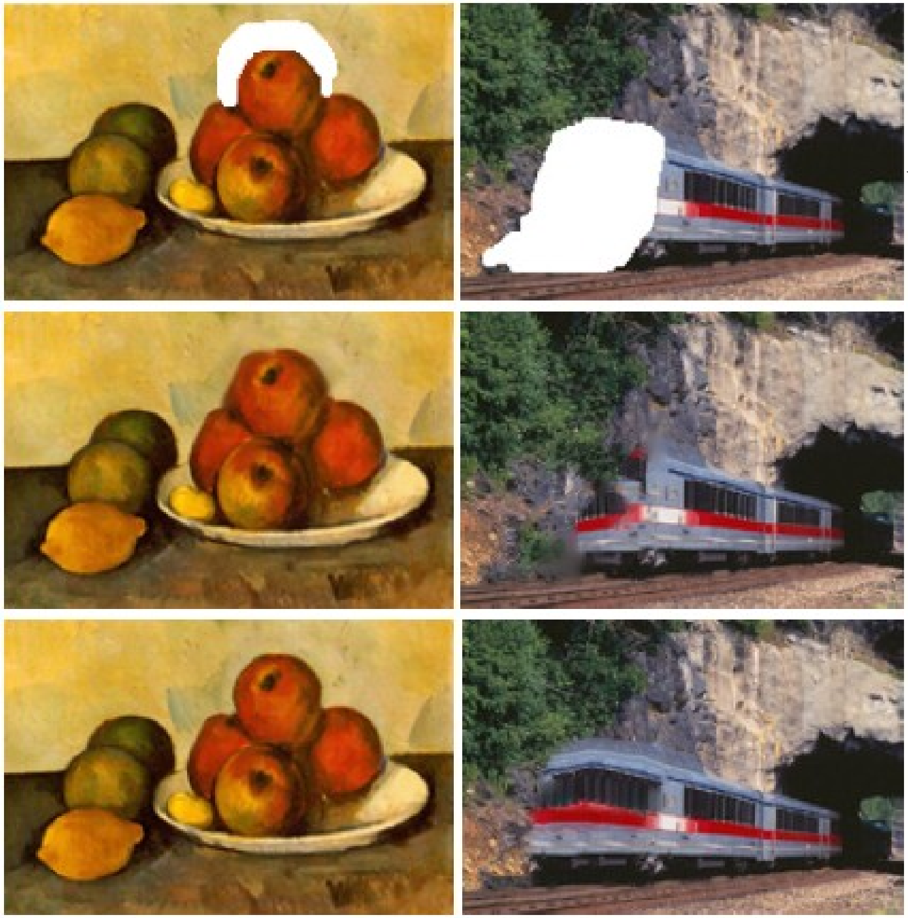
- Neighboring sampled patches may produce a “seam”
  - Binary masks
  - Defining Poisson Eqs.
  - Blending the color-edges.

Overcome the lack of information

- Sometimes there simply are not enough existing sample patches, i.e. the ratio $\frac{known\,curve}{unknown\,curve}$ is too small.

$$\longrightarrow \theta^* = argmin_\theta \{ d(R(c_{x_i};\theta), c_i) + d(c_i, R(c_{x_i};\theta)) \}$$

Performance analysis (+)



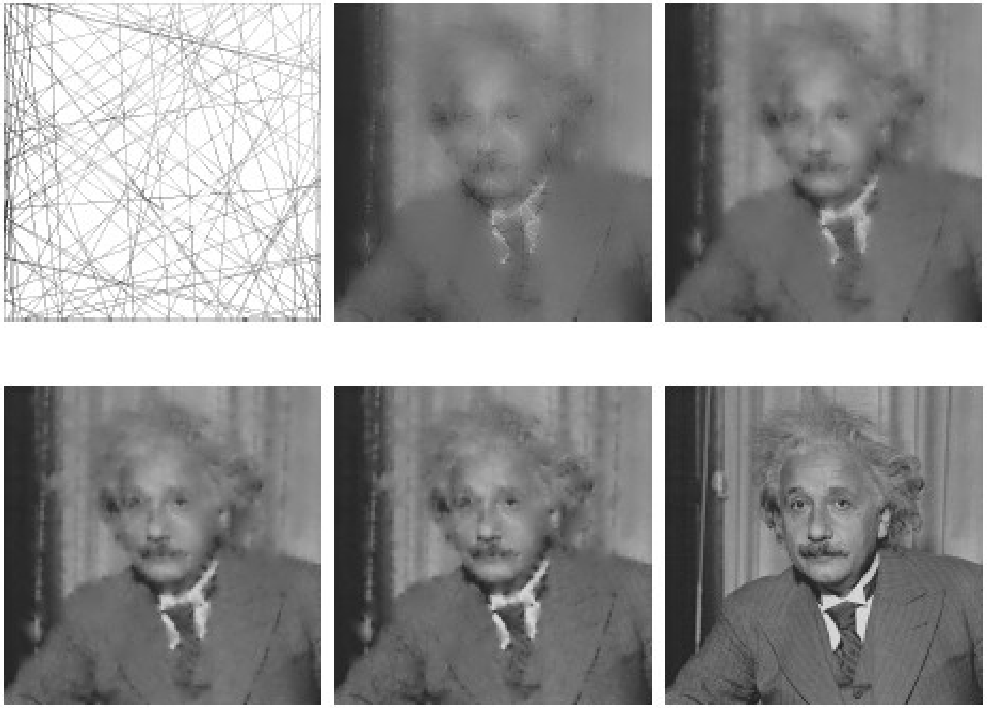
Start point

Drori et al. 03

SP

Performance analysis (-)

(a pathological example)



Limits

- Images with layered objects: 

- Insufficient number of samples (cf. Einstein)
  - But that is an inherent problem of these kind of algorithms...

The future

- The idea almost gets forced on: The authors plan to develop it further, in order to be applicable to video and meshes.
- Other graphics applications:
  - Identification (customs, forensic, etc.)
  - Observation (Main station cameras, criminology, ...)
  - Archives (paintings, later: sculptures/buildings...)
  - ...

Jumping back: "cold coffee"?

- Actually, yes :) - or... not really: another future use.