Image Inpainting by Correspondence Maps : a Deterministic Approach

Laurent Demanet
California Institute of Technology
Applied and Computational Mathematics, MC 217-50
1200 E. California Blvd, Pasadena, CA 91125, USA
demanet@acm.caltech.edu

Bing Song and Tony Chan
Univeristy of California Los Angeles
Mathematics Department
520 Portola Plaza, Los Angeles, CA 90095-1555, USA
{ songbing, chan }@math.ucla.edu

August 18, 2003

Abstract

The success of some recent texture synthesis methods, see [8, 17], suggests that there exists an underlying formulation explaining their performance and paving the way to more involved modeling. Based on their ideas, we formalize a low-level *global deterministic* solution for image inpainting.

A correspondence map is defined as linking each blank or missing pixel to the pixel where its value is taken from, in the seed image. The above-mentioned algorithms are seen as descent procedures to minimize a functional of this correspondence map, the inpainting energy. We discuss why they should not be seen as procedures to sample a probability distribution on the correspondence maps. We therefore question the claims that probability is anywhere involved at this explanatory level.

The algorithm we use is mostly taken from [17]. The latter however suffers from a strong directional bias, the direction in which texture is grown. We restore rotation-invariance at the level of both the target function and the algorithm. Our encouraging numerical results could not have been obtained by a directional texture-growing algorithm.

1 Introduction

The problem we address in this report is that of image inpainting or disocclusion. A special case is texture synthesis constrained by boundary matching with a seed texture. Given an image u where a few pixels or a whole region Ω is missing, we wish to recover the lacking information in the best possible way to the eye. A few principles come to mind when trying to give this problem a more precise statement :

1. The seed image should provide a guideline to the synthesis of the missing pixels. The result of inpainting should locally be visually close to parts of the known image.

2. In addition to considering the seed image, one might wish to consider a set of images, build a summarizing account of their properties, and use this as learned *a posteriori* knowledge.

3. Our *a priori* knowledge of some typical image properties (there are uniform regions, edges, texture, etc.) should provide other indications for the inpainting.

4. At yet a higher level, our understanding of the scene can give us hints at how to inpaint properly.

The method we present in section 2 is at a 'low level' since it only concentrates on the first approach. Variational or PDE-based inpainting methods [2, 7], on the other hand, are based on some a priori information (e.g. the functional to minimize) as well as data fitting (e.g. the boundary condition), so they fall in between the third and first categories. Markov random field texture models, cfr. [19], consist in building a big probability distribution from a large set of images, and then sampling it in order to synthesize the learned patterns. This follows approach number 2 above.

Section 2 formulates the inpainting problem as the minimization of a new inpainting functional. An algorithm to reach a good local minimum is then described. The core of the paper, sections 3, is a discussion related to the new formulation. Issues related to its deterministic vs. probabilistic interpretation are addressed. Some numerical experiments are presented in section 4. Directions for future research are given in section 5.

2 Inpainting by correspondence maps

2.1 Formulation

We are given a known image (intensity function) $u(\alpha)$ defined on the pixels $\alpha \in I\backslash\Omega$ for some missing region Ω. The problem is to recover the unknown intensities $u(\alpha)$ for $\alpha \in \Omega$. Our strategy is to define a *correspondence map* $F : \Omega \to I\backslash\Omega$ and to paste the pixel values from the seed image as $u(\alpha) := u(F(\alpha))$. This is depicted in Fig. 1.

The map F should be chosen so that the synthesized region looks as much as possible like parts of the seed image. Define the neighborhood N_α of a pixel α as, say, the set of its eight surrounding neighbors (N_α cannot contain α). Visual closeness of two pixel neighborhoods N_α and N_β can be measured using the usual Euclidean distance

$$d^2(N_\alpha, N_\beta) = \sum_{\gamma \in N_0} |u(\alpha+\gamma) - u(\beta+\gamma)|^2, \tag{1}$$

where N_0 denotes the neighborhood of the origin. One can now define a global indicator of performance, the inpainting energy or functional, as

$$E(F) = \sum_{\alpha \in \Omega} d^2(N_\alpha, N_{F(\alpha)}). \tag{2}$$

This is the objective to be minimized with respect to F. In other words, finding F amounts to filling Ω in the visually most faithful way by pixels of the seed image.

An ambiguity appears in this definition when the neighborhood $N_{F(\alpha)}$ of the target pixel is not fully contained in the image I or if it overlaps with unknown pixels in Ω. The most obvious way to deal with this problem is to restrict the range of F to 'acceptable' pixels β, in the sense that N_β is fully contained in $I\backslash\Omega$.

Classical variational formulations (see [2, 7]) are based on the smoothness of the interpolant u on the occluded region Ω. They are local in the sense that only information at the boundary $\partial\Omega$ of Ω is really taken into account. Here the situation is very different. The synthesized image need not be smooth : the objective is a functional of F, not directly of u. A minimizing correspondence map F has itself no reason to be smooth. Inpainting from a correspondence map is obviously a global (in contrast to local) method.

Our inpainting energy is at a much more elementary level than a regularization functional like TV, in the sense that it is defined with poor learned (a posteriori) knowledge or subjective (a priori) assumption on what the solution should look like. Minimizing $E(F)$ is not really a model, it is hardly more than one possible low-level formulation of the inpainting problem. A set of pixel neighborhoods built from a single texture or image sample does not qualify satisfactorily as a set of model parameters.

Let us remark that this inpainting functional $E(F)$ does not lend itself easily to minimization, unlike the nicer convex functionals sometimes found in image processing. It possesses lots of local minima. Finding a global minimum is probably a hard combinatorial problem.

2.2 Algorithm

The following algorithm is a descent for $E(F)$. It is based on local minimizations at every unknown pixel. Information is grown from the border to the inside of Ω in multiple sweeps. The main ideas are taken from [8, 17]. How our implementation differs from theirs is detailed below.

We follow the following steps :

1. Initialization of the map F at random.

2. Select a pixel α in Ω immediately near the border of Ω.

3. Update $F(\alpha)$ by neighborhood matching. This means finding a pixel $\beta \in I \backslash \Omega$ minimizer of $d(N_\alpha, N_\beta)$ so that $F(\alpha) := \beta$. Then assign $u(\alpha) := u(F(\alpha))$.

4. Repeat 2 and 3 for all the pixels inside Ω and near the border. Then go to the next row of pixels, not immediately adjacent to the border, etc. until every pixel of Ω is visited.

5. Sweep again the totality of Ω, i.e. repeat 2 through 4, until $E(F)$ stagnates at a minimum value.

This basic scheme can be made faster and more efficient in a variety of different ways. We have implemented the following improvements.

- *Codebook Pruning*. An exhaustive search is far from being necessary when looking for a good match among the seed pixels (step 3 above). One can restrict the search for a good neighborhood match to a subset of the seed pixels, e.g. chosen randomly and not too far from the current α. Typically a decimation of a factor 10 still produces visually similar results. If we call ‘codebook’ the set of neighborhoods N_α of the candidates α, this is codebook pruning.

- *More likely candidates*. There are a few natural pixels to include in the set of candidates in the seed image. They should not all be chosen randomly.

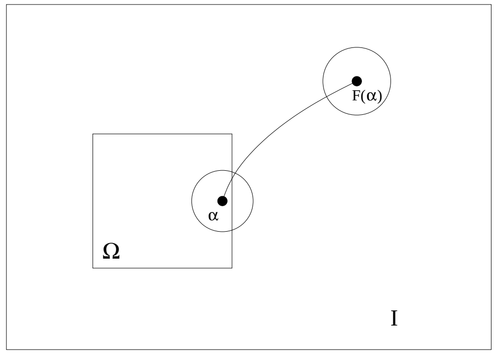


Figure 1: The correspondence map $F : \Omega \to I \backslash \Omega$

Suppose all the neighbors of a pixel α have already been inpainted, then a good candidate for $F(\alpha)$ would be $F(\alpha - \beta) + \beta$ for β close to the origin. This way, larger regions of the known image are likely to be copied in Ω. We learned that this strategy has also been adopted in [1].

- *Multiresolution*. The construction of the correspondence map can be done in a multiresolution way, by successive refinements on embedded grids. Inpainting (steps 2 through 5) is done at a very coarse scale first, on a downsampled image I and inpainting mask Ω. Using an appropriate interpolation procedure one can then recursively refine this guess at finer and finer resolutions. This way interactions can occur at a much larger scale than the size of the neighborhood (a 3 by 3 square, say).

- *Extension to color images*. This is straightforward if we understand that for vector-valued intensities the absolute value $|u|$ becomes the l^2-norm $\|u\| = \sqrt{u_1^2 + u_2^2 + u_3^2}$.

The algorithm presented in [17] is very similar to the one presented here with one noticeable difference. Their neighborhood is ‘causal’ in the sense that only known or already inpainted pixels are taken into account in the neighborhood. This has the big computational advantage of producing good results after only one sweep of Ω, for example in raster-scan order, but is highly anisotropic. In contrast, the functional (2) is rotation-invariant and the algorithm presented above is meant to correct the bias of causality. A more detailed discussion of this causal vs. non-causal aspect will be given in the next section. Let us also mention that [17] uses a clever tree structure of the codebook in order to speed up the search (by making it approximate).

3 Discussion

3.1 Remarks on the algorithm

This section aims at clarifying the deeper link between the formulation in terms of inpainting energy and the 'pixel-pasting-by-neighborhood-matching' algorithm we presented.

Let us go back for a moment to *causal* neighborhoods as in [8, 17]. We number the pixels of Ω in raster-scan order (from left to right and top to bottom) as $\alpha_1, \alpha_2, \dots, \alpha_n$. When visiting pixel α_i, step 3 of our algorithm consists in solving the following problem,

$$\begin{gathered} \min_{F(\alpha_i)} d(N_{\alpha_i}, N_{F(\alpha_i)}) \\ \text{given } F(\alpha_1), F(\alpha_2), \dots, F(\alpha_{i-1}). \end{gathered} \tag{3}$$

The seed image u on $I \backslash \Omega$ is also hidden in the constraints, we drop it for notational convenience. That the values of $F(\alpha_{i+1}), \dots$ are not present in the condition is what we refer to as the 'causality' property of the neighborhood.

This can also be seen as the maximization of the conditional probability

$$P(F(\alpha_i)|F(\alpha_1), \dots, F(\alpha_{i-1})) = Ce^{-d^2(N_{\alpha_i}, N_{F(\alpha_i)})}. \tag{4}$$

Here C is an appropriate normalization constant. The choice of an exponential is not compulsory at this level but will prove useful below when we take logarithms. It is now natural to consider the corresponding joint probability distribution $P(F(\alpha_1), \dots, F(\alpha_n))$. Causality of the neighborhood is indeed a clever way to factorize it via Bayes' rule as

$$\begin{aligned} &P(F(\alpha_1))P(F(\alpha_2)|F(\alpha_1)) \dots \\ &\qquad P(F(\alpha_n)|F(\alpha_1), F(\alpha_2), \dots, F(\alpha_{n-1})). \end{aligned} \tag{5}$$

The global 'inpainting energy' is therefore nothing but

$$\begin{aligned} E(F) &= -\log P(F(\alpha_1), \dots, F(\alpha_n)) \\ &= \sum_{\alpha_i \in \Omega} d^2(N_{\alpha_i}, N_{F(\alpha_i)}) + C, \end{aligned} \tag{6}$$

for some unimportant constant C.

Obtaining the global minimum of (6) is a different and much more difficult problem than performing the successive neighborhood matchings (3). This is true regardless of whether the neighborhood is causal or not. When designing a method to solve such a hard optimization problem there is a trade-off to solve between the ability to make the energy decrease and the ability to escape from local minima. The successive minimizations (3) can be considered as a rather naive but very fast greedy procedure to reach a good point of low energy. This is how we understand the success of the algorithms in [8, 17].

The reader might wonder where we needed causality of the neighborhood in the above reasoning. If for example each neighborhood N_α is symmetric around the pixel α, we can still introduce the inpainting energy as 6 and define a corresponding total probability distribution as $Ce^{-E(F)}$. But the latter would not neatly factorize anymore via Bayes' rule. It would actually be forbidden to consider each $Ce^{-d^2(N_\alpha, N_{F(\alpha)})}$ as a conditional probability. The formal application of Bayes' rule would give different values for the joint probability depending on the order in which pixels are taken. Instead, the right conditional probabilies to consider in the non-causal case are

$$P(F(\alpha)|F(N_\alpha)) = Ce^{-\tilde{d}^2(N_\alpha, N_{F(\alpha)})},$$

where the notation $F(N_\alpha)$ refers to all the values of F for the pixels in N_α, and with the distance $\tilde{d}$ defined as

$$\begin{aligned} \tilde{d}^2(N_\alpha, N_{F(\alpha)}) = &\sum_{\gamma \in N_0} |u(\alpha+\gamma) - u(F(\alpha)+\gamma)|^2 \\ &+ \sum_{\gamma \in N_0} |u(F(\alpha)) - u(F(\alpha-\gamma)+\gamma)|^2. \end{aligned} \tag{7}$$

This expression can be obtained by isolating the dependence on a given α in the expression of the total energy. The application of Bayes' rule does not extend much beyond

$$P(F(\alpha)|F(N_\alpha)) = \frac{P(F(\alpha) \text{ and } F(N_\alpha))}{P(F(N_\alpha))}.$$

It is instructive to compare equations (1) and (7). The first term in (7) is exactly d^2 and measures neighborhood closeness at α vs. $F(\alpha)$. The second term is present because α is itself a neighbor of the pixels which are in its neighborhood. It measures how much changing F at α disrupts the neighborhood matches for the pixels surrounding α. Of course the algorithm of section 2 can be modified by using this new distance $\tilde{d}$ instead of d. It has the advantage of making the global energy decrease at every step (this is not necessarily the case with d), and is therefore guaranteed to reach a local minimum. However this idea comes with a big drawback. The algorithm becomes a pure descent and loses all its 'creativity' to find a good local minimum. Typically it would produce uniformly gray or colored regions by copy-pasting pixels from a flat smooth region in the seed image. This is why we kept the original d in our algorithm. The true problem is the need for clever algorithms to minimize the functional $E(F)$.

3.2 Deterministic vs. Probabilistic

So far we have associated probability and energy, through $E = -\log p + C$ or $p = Ce^{-E}$, but this identification is only formal. The models people build from these two concepts can be very different.

- An 'energy' functional $E(F)$ should be *minimized*. For our purposes it is a criterion that sorts every possible argument F in a definite order, from the best one (lowest energy) to the worst ones (highest energy). We are therefore interested in the (often unique) argument of the minimum. Examples related to our purpose include regressions in statistics, denoising-deblurring problems addressed by variational or Bayesian methods in image processing, and of course variational inpainting [7].

- A probability distribution $p(F)$ should be *sampled*. For our purposes it is a criterion that sorts each event into categories, e.g. as typical or non-typical. All the typical events have approximately the same probability. All the non-typical events have approximately zero probability, or have a very high probability but then are by far outnumbered by the typical events. We are interested in any one of these typical events. Examples include the simulation of Markov Random Field models in statistical mechanics or texture synthesis [4, 19].

In some sense a probability model is 'weaker' than an energy model. It does not manage to rank all the events into a significant one-dimensional scale. Moreover, the most probable outcomes of a random vector need not look like the vast majority of typical outcomes, so it might turn out to be a bad idea to try and maximize the probability. The classical example is a multivariate random variable made of i.i.d. $N(0, 1)$, which typical samples look like 'noise', but which most probable outcome is the vector that is identically equal to zero (this is not 'typical').

The two paradigms are usually used in very different contexts to address very different questions. However it is not clear yet where solutions to inpainting should belong. As far as the simple 'neighborhood-matching' algorithms are concerned, most authors seem to classify them as sampling from a distribution. A hint in this direction is the complete factorization of the probability distribution (5) into conditional probabilities in the case of a causal neighborhood. This is an ideal setting for sampling the joint distribution : just sample the successive conditional distributions one after the other. This is called Gibbs sampling and is the core of the approach in [13].

Instead, we believe that $E(F)$ should be minimized rather than $Ce^{-E(F)}$ sampled. As far as we experienced, no harm was done when trying to reach a good minimum. The descent never seems to reach atypical highly probable states. Thus no need for a careful sampling. The pixel-pasting-by-neighborhood-matching approach is presumably not involved enough to require a true probabilistic modeling. The numerical experiments supporting this conclusion are shown in the next section.

Other thought experiments confirm this. Take for instance an infinite periodic pattern, occluded by any reasonable mask. It is perfectly inpainted by the original pattern, which is visually reasonable and therefore qualifies as ‘typical’. In that case the inpainting energy is zero, as low as it can get. There is no room for atypical solutions, no need for a probabilistic modeling.

This provides an explanation for the difference of performance of the algorithms in [8] and [17] : the former approach insists on sampling the conditional probabilities whereas the latter approach simply maximizes them. This results in a much faster algorithm without loss of visual quality of the synthesized texture. The numerical experiments supporting this conclusion are shown in the next section.

There is a way to re-introduce probability in our setting, not at a modeling level but rather as a tool to solve the optimization problem. When getting trapped at a nonsatisfactory local minima is the inevitable faith of naive descent procedures, people often resort to ’stochastic descents’ to have a better chance of reaching the global minimum. For instance, simulated annealing intuitively keeps the system from freezing at a high-energy state by “shaking it hard enough” during a slow cooling. For the particular case of our pixel synthesis algorithm, it could mean randomly assigning a disadvantageous target $F(\alpha)$ to the current pixel α. Of course such a procedure would considerably slow down the descent but could prove useful to output a better looking result. One reference for these probabilistic optimization algorithms is [4]. These tools turn out to be closely related to sampling strategies, but our message here is that the deterministic vs. probabilistic nature of the model investigated is a different question.

3.3 Continuous images

If we model the image u as a function from $I \subset \mathbb{R}^2$ to $\mathbb{R}$, and the correspondence map as a function F from $\Omega \subset R^2$ to $I \backslash \Omega \subset \mathbb{R}^2$, the total inpainting energy to be minimized over F is expressed as

$$E(F) = \int_\Omega dy \int_{\mathbb{R}^2} dx \chi(x) |u(F(y-x)) - u(F(y)-x)|^2, \tag{8}$$

where $\chi(x)$ is an indicator function of the neighborhood of 0. Then, as before, $u(x) = u(F(x))$ for every $x \in \Omega$. Well-posedness, existence, uniqueness and regularity of the minimizers of this highly nonconvex functional seem quite challenging questions.

4 Numerical experiments

Fig. 2 is a synthetic geometric example. On the left is the image to be inpainted, the 'noise' indicating the mask where the pixel intensity values are occluded or missing. On the right is the result after inpainting. In this example, the minimum of the inpainting energy is zero. Fig. 3 is de Bonet's texture sample nr. 161. Again, (a) is the occluded image, (b) is the inpainting result and (c) is the energy vs. the number of sweeps. The energy decreases quickly in a few sweeps and then stagnates till some approximately steady state is reached. Note that plot (c) was obtained only from running the algorithm at the finest resolution. Fig. 4 shows that the algorithm can also be very successfully applied to textured images. Fig. 6 is the 'Barbara' image that contains both a texture and a cartoon part : (a) is the degraded image, (b) is the result of inpainting by correspondence map. We observe that the texture part is mostly recovered. (c) is the result of TV inpainting. We can see that the texture part is not recovered at all. Another very efficient way to process this image would be to decompose the image as a texture plus cartoon image and then use a different inpainting method for each part. See [3] for details.

In all the above examples and many others the quality of the result does not degrade as the number of sweeps gets very large. This supports the conclusion that the model behind the algorithm is deterministic and not probabilistic, in the sense discussed above.

Another interesting experiment to test this claim is to start with the *occluded original image* as initial guess, and apply the algorithm[1]. We observe that, on toy inpainting problems such as the one in Fig. 7, the 'steady-state' is visually close to the original image and the inpainting energy is not significantly lowered. Once again, this validates the claim that minimizing an energy is the right framework behind the algorithm. Had the algorithm degraded the image substantially, we could have resorted to sampling strategies to avoid that phenomenon ; but this is not the case here.

Note that running the algorithm with the 'forbidden' occluded part of the image as initial data could be used in practice to erase unwanted information.

[1]No multiresolution strategy is adopted here. The initial inpainting map is computed in an obvious way from the original image by neighborhood matching.

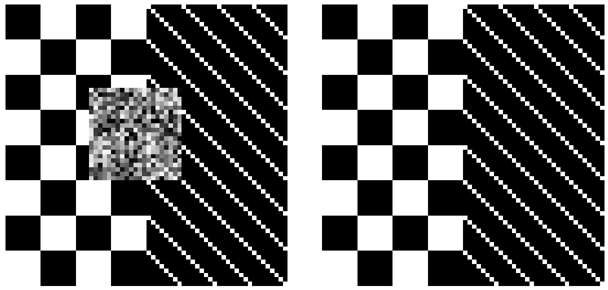
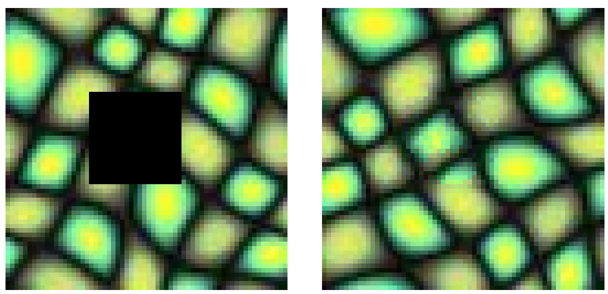
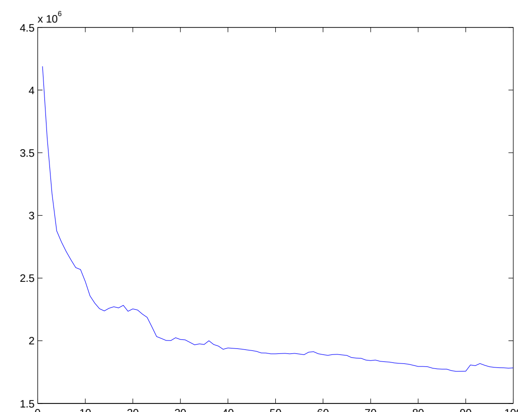


Figure 2: Inpainting a synthetic image. Left: The image to be inpainted. The 'noisy square' in the middle indicates the occluding mask. Right: After inpainting.

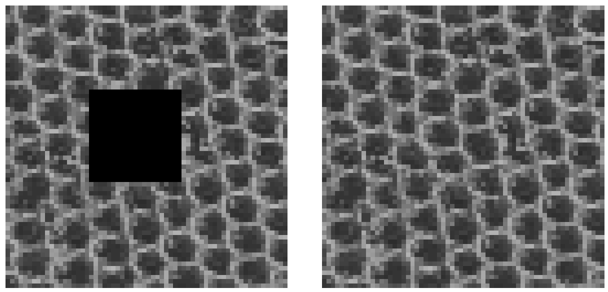


(a) (b)



(c)

Figure 3: Inpainting a textured image. *This is a color image* (a) The occluded image. (b) After inpainting (c) Inpainting energy vs. number of iterations.



(a) (b)

Figure 4: Inpainting a textured image. (a) The image to be inpainted. (b) After inpainting.

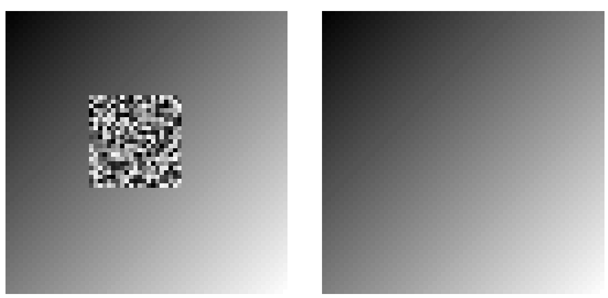
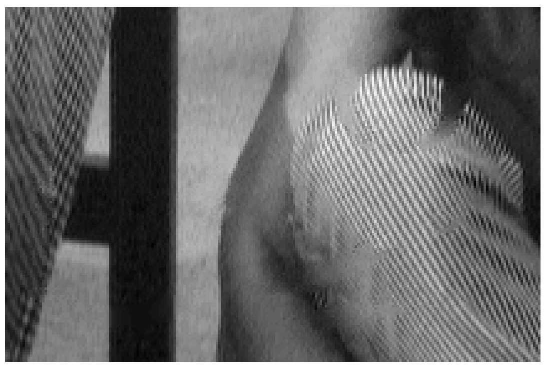
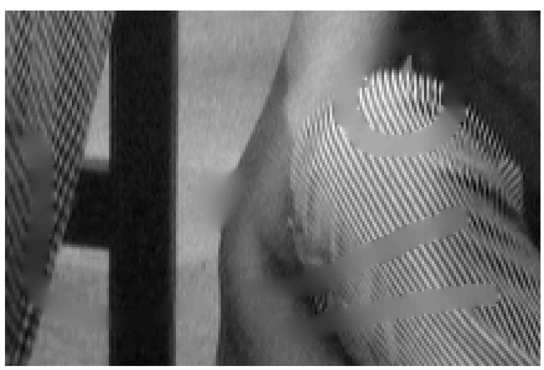


Figure 5: Inpainting a linear smooth image. Left: the image to be inpainted. Right: After inpainting.

(a)

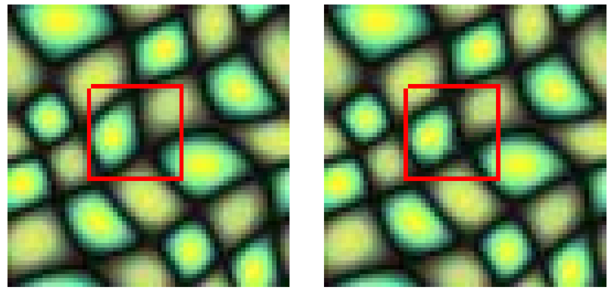


(b)



(c)

Figure 6: Inpainting a real-life image. (a)Image with missing information, (b) Result of inpainting by correspondence map. We can see that the texture part is well recovered. (c) Result of TV inpainting.



(a) (b)

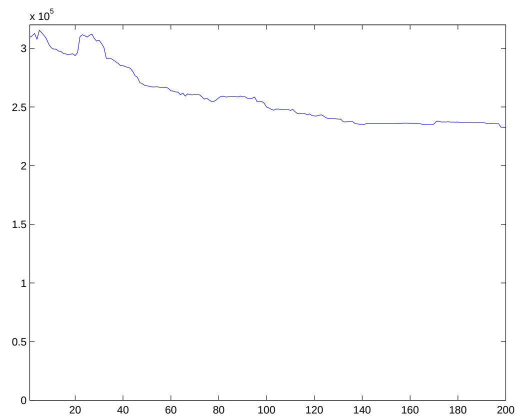


Figure 7: Taking original occluded image as initial guess and applying inpainting. This is a color image. The point is to illustrate the claim that the algorithm does not degrade the image or ‘go too far’ if the number of sweeps is large. (a) Original image. (b) After inpainting. (c) Inpainting energy vs. number of iterations. Note that the energy does not always decrease. *This example is not supposed to show the performance of the method, see explanations in the text.*

5 Extensions

The low-level inpainting solution consisting of minimizing (8) will probably gain from being formulated using some additional a priori information. A few paths can be followed.

- There is a notion of visual closeness associated with the energy (8). Namely, for every synthesized pixel there exists a pixel in the seed image in good agreement in terms of similarity of the neighborhoods. It would however make little sense if the correspondence map took the observer to a very distant pixel everytime a step is taken in the missing region. Rather than isolated pixels, we expect reasonably large patches to be pasted into that region. This amounts to requiring 'smoothness' of the correspondence map and suggests adding the following penalization term,

$$\int_\Omega dy \int_{\mathbb{R}^2} dx \chi(x) |u(F(y-x)) - u(F(y)-x)|^2$$
$$+\lambda \int_\Omega dy \, \|\nabla F(y) - I\|_F,$$
$$u(y) = u(F(y)).$$

In other words the correspondence map should locally look like the identity ($\| \cdot \|_F$ is the Frobenius norm). The parameter λ weights the importance of each term. How to efficiently implement the minimization of this new energy is the interesting problem. A step in this direction is [1] where the author does not just select the pixel candidates randomly in the seed image but also according to what has already been synthesized. Preference is precisely given to these pixels that extend the correspondence map so as to copy larger patches into the missing region. See also comments in section 2.

- Naturally, the algorithm is well-suited for texture synthesis but sometimes fails on reproducing geometrical features of the image. This is precisely what TV inpainting does reasonably well for us. It is therefore tempting to write combined models such as the following minimization problem.

$$\min_{u,F} E(F) + \lambda_1 TV(u) + \lambda_2 ||u(F(x)) - u(x)||_2^2,$$

where $TV(u)$ is the TV norm of u, $E(F)$ is the total inpainting energy defined in (8). The issue would then again be to find a clever algorithm to minimize this. It is probably a good idea to make λ_1 decrease as the number of iterations (sweeps) increases. This situation would more or less correspond to choosing the TV inpainting as initial guess for the usual algorithm.

Acknowledgements

We are indebted to Yingnian Wu and Jackie Shen for interesting discussions, and grateful to the referees for helping us improve the manuscript. L.D. would like to thank the UCLA imagers for their hospitality. He is a fellow of the Belgian American Educational Foundation and the Francqui Foundation.

References

[1] M. Ashikhmin. Synthesizing natural textures. In *The proceedings of 2001 ACM Symposium on Interactive 3D Graphics*, pages 217–226, March 2001.

[2] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Trans. on Image Processing*, 10(8):1200–1211, August 2001.

[3] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *UCLA CAM Report 02-47*, July 2002.

[4] P. Brémaud. *Markov Chains: Gibbs fields, Monte-Carlo simulation, and queues*. Springer, 1999.

[5] T. F. Chan, S.-H. Kang, and J. Shen. Euler's elastica and curvature-based image inpainting. *SIAM J. Appl. Math*, 63(2):564–592, 2002.

[6] T. F. Chan and J. Shen. Non-texture inpainting by curvature-driven diffusions (cdd). *J. Visual Comm. Image Rep.*, 12(4):436–449, 2001.

[7] T. F. Chan and J. Shen. Mathematical models of local non-texture inpaintings. *SIAM J. Appl. Math*, 62(3):1019–1043, 2002.

[8] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision 2*, pages 1033–1038, September 1999.

[9] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH 2001 Conference Proceedings*, pages 341–346, August 2001.

[10] D. Heeger and J. Bergen. Pyramid-based texture analysis/synthesis. *Computer Graphics (SIGGRAPH 1995)*, pages 229–238, July 1995.

[11] M. Masnou and J. Morel. Level-lines based disocclusion. *IEEE Int. Conf. on Image Processing*, pages 259–263, October 1998.

[12] J.-M. Morel and S. Solimini. *Variational Methods in Image Segmentation*, volume 14 of *Progress in Nonlinear Differential Equations and Their Applications*. Birkhäuser, Boston, 1995.

[13] K. Popat and R. Picard. Novel cluster-based probability model for texture synthesis, classification and compression. In *Visual Communications and Image Processing*, pages 756–768, 1993.

[14] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithm. *Physica D*, 60:259–268, 1992.

[15] G. Turk. Texture synthesis on surfaces. In *SIGGRAPH 2001 Conference Proceedings*, pages 347–354, August 2001.

[16] L. Vese and S. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *UCLA CAM Report 02-19*, May 2002.

[17] L.-Y. Wei and M. Levoy. Fast texture synthesis using Tree-Structured vector quantization. In *Proceedings of SIGGRAPH 2000*, pages 479–488, July 2000.

[18] L.-Y. Wei and M. Levoy. Texture synthesis over arbitrary manifold surfaces. In *SIGGRAPH 2001 Conference Proceedings*, pages 355–360, August 2001.

[19] S. Zhu, Y. Wu, and D. Mumford. Filters, Random Fields and Maximum Entropy (FRAME): Towards a unified theory for texture modeling. *Internatinal Journal of Computer Visison*, 27(2):107–126, March 1998.