

# Inpainting

MARCELO BERTALMIÓ, VICENT CASELLES, SIMON MASNOU, GUILLERMO SAPIRO

## Synonyms

- Disocclusion
- Completion
- Filling-in
- Error concealment

## Related Concepts

- Texture synthesis

## Definition

Given an image and a region  $\Omega$  inside it, the inpainting problem consists in modifying the image values of the pixels in  $\Omega$  so that this region does not stand out with respect to its surroundings. The purpose of inpainting might be to restore damaged portions of an image (e.g. an old photograph where folds and scratches have left image gaps) or to remove unwanted elements present in the image (e.g. a microphone appearing in a film frame). See figure 1. The region  $\Omega$  is always given by the user, so the localization of  $\Omega$  is not part of the inpainting problem. Almost all inpainting algorithms treat  $\Omega$  as a hard constraint, whereas some methods allow some relaxing of the boundaries of  $\Omega$ .

This definition, given for a single-image problem, extends naturally to the multi-image case therefore this entry covers both image and video inpainting. What is not however considered in this text is *surface* inpainting (e.g. how to fill holes in 3D scans), although this problem has been addressed in the literature.



**Fig. 1.** The inpainting problem. Left: original image. Middle: inpainting mask  $\Omega$ , in black. Right: an inpainting result. Figure taken from [20].

## Background

The term *inpainting* comes from art restoration, where it is also called *retouching*. Medieval artwork started to be restored as early as the Renaissance, the motives being often as much to bring medieval pictures “up to date” as to fill-in any gaps. The need to retouch the image in an unobtrusive way extended naturally from paintings to photography and film. The purposes remained the same: to revert deterioration (e.g. scratches and dust spots in film), or to add or remove elements (e.g. the infamous “airbrushing” of political enemies in Staliner-era U.S.S.R). In the digital domain, the inpainting problem first appeared under the name “error concealment” in telecommunications, where the need was to fill-in image blocks that had been lost during data transmission. One of the first works to address automatic inpainting in a general setting dubbed it “image disocclusion,” since it treated the image gap as an occluding object that had to be removed, and the image underneath would be the restoration result. Popular terms used to denote inpainting algorithms are also “image completion” and “image fill-in”.

## Application

The extensive literature on digital image inpainting may be roughly grouped into three categories: patch-based, sparse, and PDEs/variational methods.

### From texture synthesis to patch-based inpainting

Efros and Leung [14] proposed a method that, although initially intended for texture synthesis, has proven most effective for the inpainting problem. The image gap is filled-in recursively, inwards from the gap boundary: each “empty” pixel  $P$  at the boundary is filled with the value of the pixel  $Q$  (lying outside the image gap, i.e.  $Q$  is a pixel with valid information) such that the neighborhood  $\Psi(Q)$  of  $Q$  (a square patch centered in  $Q$ ) is most similar to the (available) neighborhood  $\Psi(P)$  of  $P$ . Formally, this can be expressed as an optimization problem:

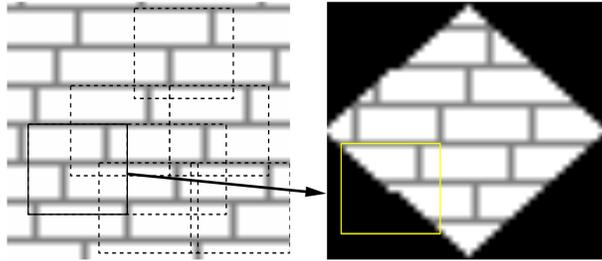
$$\text{Output}(P) = \text{Value}(Q), \quad P \in \Omega, \quad Q \notin \Omega, \quad Q = \arg \min d(\Psi(P), \Psi(Q)), \quad (1)$$

where  $d(\Psi(P), \Psi(Q))$  is the Sum of Squared Differences (SSD) among the patches  $\Psi(P)$  and  $\Psi(Q)$  (considering only available pixels):

$$d(\Psi_1, \Psi_2) = \sum_i \sum_j |\Psi_1(i, j) - \Psi_2(i, j)|^2, \quad (2)$$

and the indices  $i, j$  span the extent of the patches (e.g. if  $\Psi$  is an  $11 \times 11$  patch then  $0 \leq i, j \leq 10$ ). Once  $P$  is filled-in, the algorithm marches on to the next pixel at the boundary of the gap, never going back to  $P$  (whose value is, therefore, not altered again). See Figure 2 for an overview of the algorithm and Figure 3 for an example of the outputs it can achieve. The results are really impressive for a wide range of images. The main shortcomings of this algorithm are its computational cost, the selection of the neighborhood size (which in the original paper is a global user-selected parameter, but which should change locally depending on

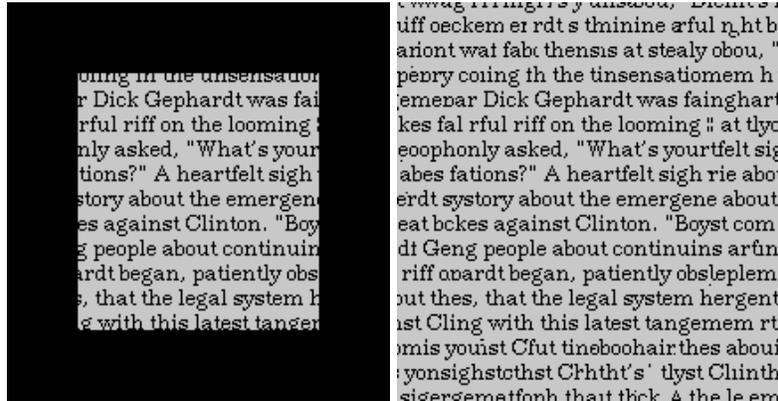
image content), the filling order (which may create unconnected boundaries for some objects) and the fact that it cannot deal well with image perspective (it was intended to synthesize frontal textures, hence neighborhoods are compared always with the same size and orientation). Also, results are poor if the image gap is very large and disperse (e.g. an image where 80% of the pixels have been lost due to random *salt and pepper noise*).



**Fig. 2.** Eros and Leung’s algorithm overview (figure taken from [14]). Given a sample texture image (left), a new image is being synthesized one pixel at a time (right). To synthesize a pixel, the algorithm first finds all neighborhoods in the sample image (boxes on the left) that are similar to the pixels neighborhood (box on the right) and then randomly chooses one neighborhood and takes its center to be the newly synthesized pixel.

Criminisi et al. [12] improved on this work in two aspects. Firstly, they changed the filling order from the original “onion-peel” fashion to a priority scheme where empty pixels at the edge of an image object have higher priority than empty pixels on flat regions. Thus, they are able to correctly inpaint straight object boundaries which could have otherwise ended up disconnected with the original formulation. See Figure 4. Secondly, they copy entire patches instead of single pixels, so this method is considerably faster. Several shortcomings remain, though, like the inability to deal with perspective and the need to manually select the neighborhood size (here there are two sizes to set, one for the patch to compare with and another for the patch to copy from). Also, objects with curved boundaries may not be inpainted correctly.

Ashikhmin [2] contributed as well to improve on the original method of Eros and Leung [14]. With the idea of reducing the computational cost of the procedure, he proposed to look for the best candidate  $Q$  to copy its value to the empty pixel  $P$  not searching the whole image but only searching among the candidates of the neighbors of  $P$  which have already been inpainted. See Figure 5. The speed-up achieved with this simple technique is considerable, and also there is a very positive effect regarding the visual quality of the output. Other methods reduce the search space and computational cost involved in the candidate patch search by organizing image patches in tree structures, reducing the dimension-



**Fig. 3.** Left: original image, inpainting mask  $\Omega$  in black. Right: inpainting result obtained with Efros and Leung’s algorithm, images taken from their paper [14].

ality of the patches with techniques like Principal Component Analysis (PCA), or using randomized approaches.

While most image inpainting methods attempt to be fully automatic (aside from the manual setting of some parameters), there are user-assisted methods that provide remarkable results with just a little input from the user. In the work by Sun et al. [27] the user must specify curves in the unknown region, curves corresponding to relevant object boundaries. Patch synthesis is performed along these curves inside the image gap, by copying from patches that lie on the segments of these curves which are outside the gap, in the “known” region. Once these curves are completed, in a process which the authors call *structure propagation*, the remaining empty pixels are inpainted using a technique like the one by Ashikhmin [2] with priorities as in Criminisi et al. [12]. Barnes et al. [5] accelerate this method and make it interactive, by employing randomized searches and combining into one step the structure propagation and texture synthesis processes of Sun et al. [27].

### The role of sparsity

After the introduction of patch-based methods for texture synthesis by Efros and Leung [14], and image inpainting by Criminisi et al [12], it became clear that the patches of an image provide a good dictionary to express other parts of the image. This idea has been successfully applied to other areas of image processing, e.g. denoising and segmentation.

More general sparse image representations using dictionaries have proven their efficiency in the context of inpainting. For instance, using overcomplete dictionaries adapted to the representation of image geometry and texture, Elad et al. [15] proposed an image decomposition model with sparse coefficients for



**Fig. 4.** Left: original image. Right: inpainting result obtained with the algorithm of Criminisi et al. [12], images taken from their paper.

the geometry and texture components of the image, and showed that the model can be easily adapted for image inpainting. A further description of this model follows.

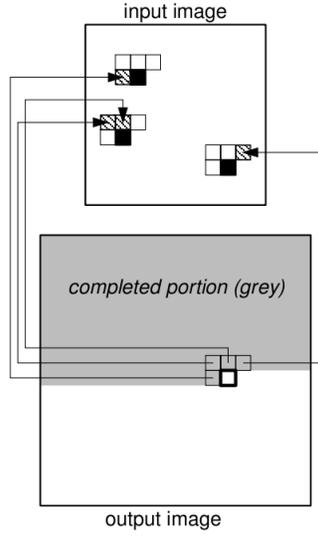
Let  $u$  be an image represented as a vector in  $\mathbb{R}^N$ . Let the matrices  $D_g, D_t$  of sizes  $N \times k_g$  and  $N \times k_t$  represent two dictionaries adapted to geometry and texture, respectively. If  $\alpha_g \in \mathbb{R}^{k_g}$  and  $\alpha_t \in \mathbb{R}^{k_t}$  represent the geometry and texture coefficients, then  $u = D_g\alpha_g + D_t\alpha_t$  represents the image decomposition using the dictionaries collected in  $D_g$  and  $D_t$ . A sparse image representation is obtained by minimizing

$$\min_{(\alpha_g, \alpha_t): u = D_g\alpha_g + D_t\alpha_t} \|\alpha_g\|_p + \|\alpha_t\|_p, \quad (3)$$

where  $p = 0, 1$ . Although the case  $p = 0$  represents the sparseness measure (i.e., the number of non zero coordinates) it leads to a non-convex optimization problem whose minimization is more complex. The case  $p = 1$  yields a convex and tractable optimization problem leading also to sparseness. Introducing the constraint by penalization (thus, in practice, relaxing it) and regularizing the geometric part of the decomposition with a total variation semi-norm penalization, Elad et al [15] propose the variational model:

$$\min_{(\alpha_g, \alpha_t)} \|\alpha_g\|_1 + \|\alpha_t\|_1 + \lambda \|u - D_g\alpha_g - D_t\alpha_t\|_2^2 + \gamma TV(D_g\alpha_g), \quad (4)$$

where  $TV$  denotes the total variation,  $\lambda, \gamma > 0$ . This model can be easily adapted to a model for image inpainting. Observe that  $u - D_g\alpha_g - D_t\alpha_t$  can be interpreted as the noise component of the image and  $\lambda$  is a penalization parameter



**Fig. 5.** Ashikhmin’s texture synthesis method (figure taken from [2]). Each pixel in the current L-shaped neighborhood generates a shifted candidate pixel (black) according to its original position (hatched) in the input texture. The best pixel is chosen among these candidates only. Several different pixels in the current neighborhood can generate the same candidate.

that depends inversely on the noise power. Then the inpainting mask can be interpreted as a region where the noise is very large (infinite). Thus, if  $M = 0$  and  $= 1$  identify the inpainting mask and the known part of the image, respectively, then the extension of (4) to inpainting can be written as

$$\min_{(\alpha_g, \alpha_t)} \|\alpha_g\|_1 + \|\alpha_t\|_1 + \lambda \|M(u - D_g \alpha_g - D_t \alpha_t)\|_2^2 + \gamma TV(D_g \alpha_g). \quad (5)$$

Writing the energy in (5) using  $u_g := D_g u$ ,  $u_t := D_t u$  as unknown variables, it can be observed that  $\alpha_g = D_g^+ u_g + r_g$ ,  $\alpha_t = D_t^+ u_t + r_t$ , where  $D_g^+$ ,  $D_t^+$  denote the corresponding pseudoinverse matrices and  $r_g, r_t$  are in the null spaces of  $D_g$  and  $D_t$ , respectively. Assuming for simplicity, as in Elad et al [15], that  $r_g = 0$ ,  $r_t = 0$ , the model (5) can be written as

$$\min_{(\alpha_g, \alpha_t)} \|D_g^+ u_g\|_1 + \|D_t^+ u_t\|_1 + \lambda \|M(u - u_g - u_t)\|_2^2 + \gamma TV(u_g). \quad (6)$$

This simplified model is justified in Elad et al [15] by several reasons: it is an upper bound for (5), is easier to solve, it provides good results, it has a Bayesian interpretation, and is equivalent to (5) if  $D_g$  and  $D_t$  are non-singular, or when using the  $\ell^2$  norm in place of the  $\ell^1$  norm. The model has nice features

since it permits to use adapted dictionaries for geometry and texture, treats the inpainting as missing samples and the sparsity model is included with  $\ell^1$  norms that are easy to solve.

This framework has been adapted to the use of dictionaries of patches and has been extended in several directions like image denoising, filling-in missing pixels (Aharon et al [1]), color image denoising, demosaicing and inpainting of small holes (Mairal et al [21], and further extended to deal with multiscale dictionaries and to cover the case of video sequences in Mairal et al [22]. To give a brief review of this model some notation is required. Image patches are squares of size  $n = \sqrt{n} \times \sqrt{n}$ . Let  $D$  be a dictionary of patches represented by a matrix of size  $n \times k$ , where the elements of the dictionary are the columns of  $D$ . If  $\alpha \in \mathbb{R}^k$  is a vector of coefficients, then  $D\alpha$  represents the patch obtained by linear combination of the columns of  $D$ . Given an image  $v(i, j)$ ,  $i, j \in \{1, \dots, N\}$ , the purpose is to find a dictionary  $\hat{D}$ , an image  $\hat{u}$  and coefficients  $\hat{\alpha} = \{\hat{\alpha}_{i,j} \in \mathbb{R}^k : i, j \in \{1, \dots, N\}\}$  which minimize the energy

$$\min_{(\alpha, D, u)} \lambda \|v - u\|_2 + \sum_{i,j=1}^N \mu_{i,j} \|\alpha_{i,j}\|_0 + \sum_{i,j=1}^N \|D\alpha_{i,j} - R_{i,j}u\|_2, \quad (7)$$

where  $R_{i,j}u$  denotes the patch of  $u$  centered at  $(i, j)$  (dismissing boundary effects), and  $\mu_{i,j}$  are positive weights. The solution of the nonconvex problem (7) is obtained using an alternate minimization: a sparse coding step where one computes  $\alpha_{i,j}$  knowing the dictionary  $D$  for all  $i, j$ , a dictionary update using a sequence of one rank approximation problems to update each column of  $D$  (Aharon, Elad, and Bruckstein [1]), and a final reconstruction step given by the solution of

$$\min_u \lambda \|v - u\|_2 + \sum_{i,j=1}^N \|\hat{D}\alpha_{i,j} - R_{i,j}u\|_2. \quad (8)$$

Again, the inpainting problem can be considered as a case of non-homogeneous noise. Defining for each pixel  $(i, j)$  a coefficient  $\beta_{i,j}$  inversely proportional to the noise variance, a value of  $\beta_{i,j} = 0$  may be taken for each pixel in the inpainting mask. Then the inpainting problem can be formulated as

$$\min_{(\alpha, D, u)} \lambda \|\beta \otimes (v - u)\|_2 + \sum_{i,j=1}^N \mu_{i,j} \|\alpha_{i,j}\|_0 + \sum_{i,j=1}^N \|(R_{i,j}\beta) \otimes (D\alpha_{i,j} - R_{i,j}u)\|_2, \quad (9)$$

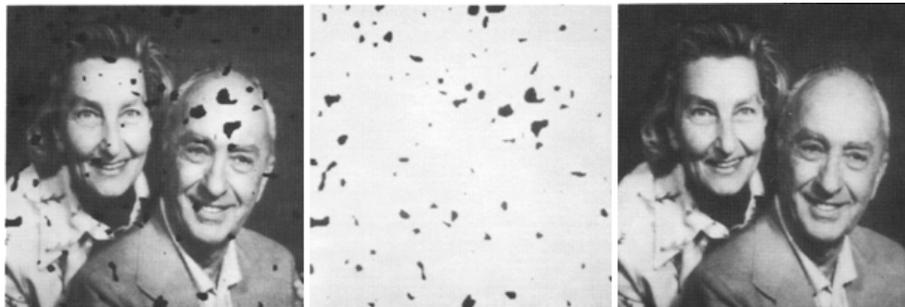
where  $\beta = (\beta_{i,j})_{i,j=1}^N$ , and  $\otimes$  denotes the elementwise multiplication between two vectors.

With suitable adaptations, this model has been applied to inpainting (of relatively small holes), to interpolation from sparse irregular samples and super-resolution, to image denoising, demosaicing of color images, and video denoising and inpainting, obtaining excellent results, see Mairal et al [22].

## PDEs and variational approaches

All the methods mentioned so far are based on the same principle: a missing/corrupted part of an image can be well synthesized by suitably sampling and copying uncorrupted patches (taken either from the image itself or built from a dictionary). A very different point of view underlies many contributions involving either a variational principle, through a minimization process, or a (non necessarily variational) partial differential equation (PDE).

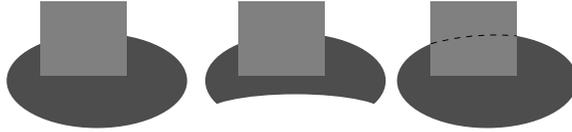
An early interpolation method that applies for inpainting is due to Ogden, Adelson, Bergen, and Burt [24]. Starting from an initial image, a Gaussian filtering is built by iterated convolution and subsampling. Then, a given inpainting domain can be filled-in by successive linear interpolations, downsampling and upsampling at different levels of the Gaussian pyramid. The efficiency of such approach is illustrated in Figure 6.



**Fig. 6.** An inpainting experiment taken from Ogden et al [24]. The method uses a Gaussian pyramid and a series of linear interpolations, downsampling, and upsampling.

Masnou and Morel proposed in [23] to interpolate a gray-valued image by extending its isophotes (the lines of constant intensity) in the inpainting domain. This approach is very much in the spirit of early works by Kanizsa, Ullman, Horn, Mumford and Nitzberg to model the ability of the visual system to complete edges in an occlusion or visual illusion context. This is illustrated in Figure 7. The general completion process involves complicated phenomena that cannot be easily and univocally modeled. However, experimental results show that, in simple occlusion situations, it is reasonable to argue that the brain extrapolates broken edges using elastica-type curves, i.e., curves that join two given points with prescribed tangents at these points, a total length lower than a given  $L$ , and minimize the Euler elastica energy  $\int |\kappa(s)|^2 ds$ , with  $s$  the curve arc-length and  $\kappa$  the curvature.

The model by Masnou and Morel [23] generalizes this principle to the isophotes of a gray-valued image. More precisely, denoting  $\hat{\Omega}$  a domain slightly larger than  $\Omega$ , it is proposed in [23] to extrapolate the isophotes of an image  $u$ , known out-



**Fig. 7.** Amodal completion: the visual system automatically completes the broken edge in the left figure. The middle figure illustrates that, here, no global symmetry process is involved: in both figures, the same edge is synthesized. In such simple situation, the interpolated curve can be modeled as a Euler’s elastica, i.e. a curve with clamped points and tangents at its extremities, and with minimal oscillations.

side  $\Omega$  and valued in  $[m, M]$ , by a collection of curves  $\{\gamma_t\}_{t \in [m, M]}$  with no mutual crossings, that coincide with the isophotes of  $u$  on  $\tilde{\Omega} \setminus \Omega$  and that minimize the energy

$$\int_m^M \int_{\gamma_t} (\alpha + \beta |\kappa_{\gamma_t}|^p) ds dt. \quad (10)$$

Here  $\alpha, \beta$  are two context-dependent parameters. This energy penalizes a generalized Euler’s elastica energy, with curvature to the power  $p > 1$  instead of 2, of all extrapolation curves  $\gamma_t, t \in [m, M]$ .

An inpainting algorithm, based on the minimization of (10) in the case  $p = 1$ , is proposed by Masnou and Morel in [23]. A globally minimal solution is computed using a dynamic programming approach that reduces the algorithmical complexity. The algorithm handles only simply connected domains, i.e., those with no holes. In order to deal with color images, RGB images are turned into a luma/chrominance representation, e.g. YCrCb, or Lab, and each channel is processed independently. The reconstruction process is illustrated in Figure 8.

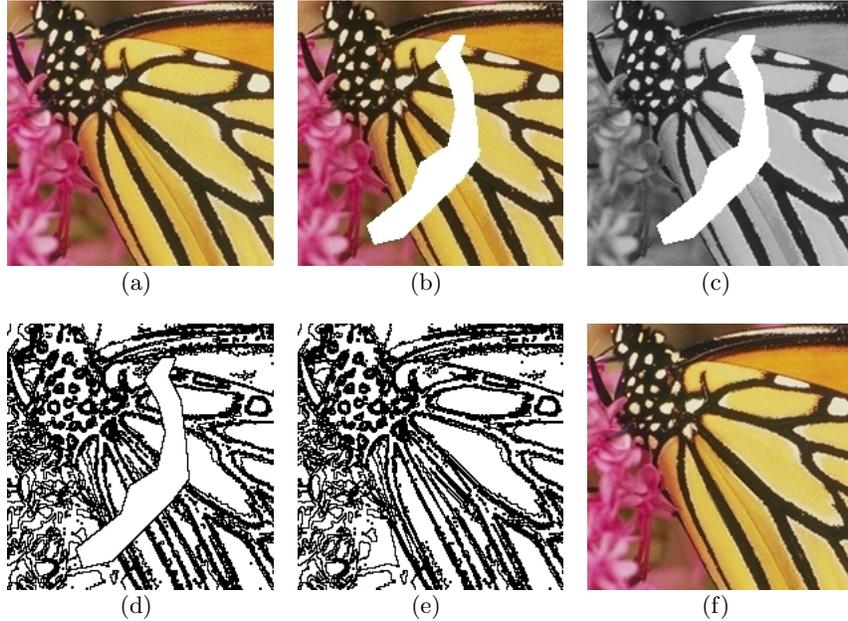
The word *inpainting*, in the image processing context, has been coined first by Bertalmío, Sapiro, Caselles, and Ballester in [7], where a PDE model is proposed in the very spirit of real paintings restoration. More precisely, being  $u$  a gray-valued image to be inpainted in  $\Omega$ , a time stepping method for the transport-like equation

$$\begin{aligned} u_t &= \nabla^\perp u \cdot \nabla \Delta u \quad \text{in } \Omega, \\ u &\text{ given in } \Omega^c, \end{aligned} \quad (11)$$

is combined with anisotropic diffusion steps that are interleaved for stabilization, using the following diffusion model

$$u_t = \varphi_\epsilon(x) |\nabla u| \nabla \cdot \frac{\nabla u}{|\nabla u|}, \quad (12)$$

where  $\varphi_\epsilon$  is a smooth cut-off function that forces the equation to act only in  $\Omega$ , and  $\nabla \cdot (\nabla u / |\nabla u|)$  is the curvature along isophotes. This diffusion equation, that



**Fig. 8.** 8(a) is the original image and 8(b) the image with occlusions in white. The luminance channel is shown in Figure 8(c). A few isophotes are drawn in Figure 8(d) and their reconstruction by the algorithm of Masnou and Morel [23] is given in Figure 8(e). Applying the same method to the luminance, hue, and saturation channels, yields the final result of Figure 8(f).

has been widely used for denoising an image while preserving its edges, compensates any shock possibly created by the transport-like equation. What is the meaning of Equation (11)? Following Bertalmío et al [7],  $\Delta u$  is a measure of image smoothness, and stationary points for the equation are images for which  $\Delta u$  is constant along the isophotes induced by the vector field  $\nabla^\perp u$ . Equation (11) is not explicitly a transport equation for  $\Delta u$ , but, in the equivalent form,

$$u_t = -\nabla^\perp \Delta u \cdot \nabla u \quad (13)$$

it is a transport equation for  $u$  being convected by the field  $\nabla^\perp \Delta u$ . Following Bornemann and März [9], this field is in the direction of the level lines of  $\Delta u$ , which are related to the Marr-Hildreth edges. Indeed, the zero crossings of (a convoluted version of)  $\Delta u$  are the classical characterization of edges in the celebrated model of Marr and Hildreth. In other words, as in the real paintings restoration, the approach of Bertalmío et al [7] consists in conveying the image intensities along the direction of the edges, from the boundary of the inpainting domain  $\Omega$  towards the interior. The efficiency of such approach is illustrated in Figure 9. From a numerical viewpoint, the transport equation and the anisotropic diffusion can be implemented with classical finite difference schemes. For color images, the coupled system can be applied independently to each channel of

any classical luma/chrominance representation. There is no restriction on the topology of the inpainting domain.



**Fig. 9.** An experiment taken from Bertalmío et al [7]. Left: original image. Middle: a user-defined mask. Right: the result with the algorithm of [7].

Another perspective on this model is provided by Bertalmío, Bertozzi, and Sapiro in [6], where connections with the classical Navier-Stokes equation of fluid dynamics are shown. Indeed, the steady state equation of Bertalmío et al [7],

$$\nabla^\perp u \cdot \nabla \Delta u = 0,$$

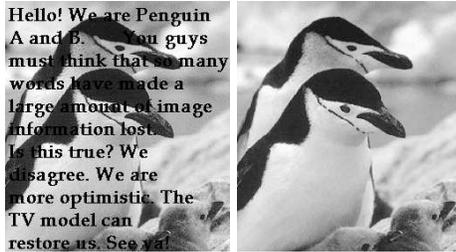
is exactly the equation satisfied by steady state inviscid flows in the two-dimensional incompressible Navier-Stokes model. Although the anisotropic diffusion equation (12) is not the exact counterpart of the viscous diffusion term used in the Navier-Stokes model for incompressible and Newtonian flows, yet a lot of the numerical knowledge on fluid mechanics seems to be adaptable to design stable and efficient schemes for inpainting. Results in this direction are shown in Bertalmío, Bertozzi, and Sapiro [6].

Chan and Shen propose in [11] a denoising/inpainting first-order model based on the joint minimization of a quadratic fidelity term outside  $\Omega$  and a total variation criterion in  $\Omega$ , i.e., the joint energy

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

with  $A \supset \Omega$  the image domain and  $\lambda$  a Lagrange multiplier. The existence of solutions to this problem follows easily from the properties of functions of bounded variation. As for the implementation, Chan and Shen look for critical points of the energy using a Gauss-Jacobi iteration scheme for the linear system associated to an approximation of the Euler-Lagrange equation by finite differences.

More recent approaches to the minimization of total variation with subpixel accuracy should nowadays be preferred. From the phenomenological point of view, the model of Chan and Shen [11] yields inpainting candidates with the smallest possible isophotes. It is therefore more suitable for thin or sparse domains. An illustration of the model's performances is given in Figure 10



**Fig. 10.** An experiment taken from Chan and Shen [11]. Left: original image. Right: after denoising and removal of text.

Turning back to the criterion (10), a similar penalization on  $\tilde{\Omega}$  of both the length and the curvature of all isophotes of an image  $u$  yields two equivalent forms, in the case where  $u$  is smooth enough (see Masnou and Morel [23]):

$$\int_{-\infty}^{+\infty} \int_{\{u=t\} \cap \tilde{\Omega}} (\alpha + \beta |\kappa|^p) ds dt = \int_{\tilde{\Omega}} |\nabla u| \left( \alpha + \beta \left| \nabla \cdot \frac{\nabla u}{|\nabla u|} \right|^p \right) dx. \quad (14)$$

There have been various contributions to the numerical approximation of critical points for this criterion. A fourth-order time-stepping method is proposed by Chan, Kang, and Shen in [10] based on the approximation of the Euler-Lagrange equation, for the case  $p = 2$ , using upwind finite differences and a min-mod formula for estimating the curvature. Such high-order evolution method suffers from well-known stability and convergence issues that are difficult to handle.

A model, slightly different from (14), is tackled by Ballester, Bertalmío, Caselles, Sapiro, and Verdera in [4] using a relaxation approach. The key idea is to replace the second-order term  $\nabla \cdot \frac{\nabla u}{|\nabla u|}$  with a first-order term depending on an auxiliary variable. More precisely, Ballester et al study in [4] the minimization of

$$\int_{\tilde{\Omega}} |\nabla \cdot \theta|^p (a + b |\nabla k * u|) dx + \alpha \int_{\tilde{\Omega}} (|\nabla u| - \theta \cdot \nabla u) dx,$$

under the constraint that  $\theta$  is a vector field with subunit modulus and prescribed normal component on the boundary of  $\tilde{\Omega}$ , and  $u$  takes values in the same range as in  $\Omega^c$ . Clearly,  $\theta$  plays the role of  $\nabla u / |\nabla u|$  but the new criterion is much less singular. As for  $k$ , it is a regularizing kernel introduced for technical reasons in order to ensure the existence of a minimizing couple  $(u, \theta)$ . The main difference

between the new relaxed criterion and (14), besides singularity, is the term  $\int_{\tilde{\Omega}} |\nabla \cdot \theta|^p$  which is more restrictive, despite the relaxation, than  $\int_{\tilde{\Omega}} |\nabla u| \left| \nabla \cdot \frac{\nabla u}{|\nabla u|} \right|^p dx$ . However, the new model has a nice property: a gradient descent with respect to  $(u, \theta)$  can be easily computed and yields two coupled second-order equations whose numerical approximation is standard. Results obtained with this model are shown in Figure 11.



**Fig. 11.** Two inpainting results obtained with the model proposed by Ballester et al [4]. Observe in particular how curved edges are restored.

The Mumford-Shah-Euler model by Esedoglu and Shen [17] is also variational. It combines the celebrated Mumford-Shah segmentation model for images and the Euler's elastica model for curves, i.e., denoting  $u$  a piecewise weakly smooth function, that is a function with integrable squared gradient out of a discontinuity set  $K \subset \tilde{\Omega}$ , the proposed criterion reads

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

Two numerical approaches to the minimization of such criterion are discussed in Esedoglu and Shen [17]: first, a level set approach based on the representation of  $K$  as the zero-level set of a sequence of smooth functions that concentrate, and the explicit derivation, using finite differences, of the Euler-Lagrange equations associated with the criterion; the second method addressed by Esedoglu and Shen is a  $\Gamma$ -convergence approach based on a result originally conjectured by De Giorgi and recently proved by Schätzle. In both cases, the final system of discrete equations is of order four, facing again difficult issues of convergence and stability.

More recently, following the work of Grzibovskis and Heintz on the Willmore flow, Esedoglu, Ruuth, and Tsai [16] have addressed the numerical flow associated with the Mumford-Shah-Euler model using a promising convolution/thresholding method that is much easier to handle than the previous approaches.

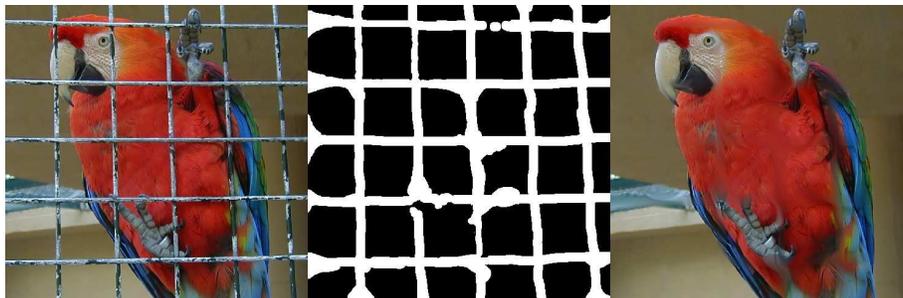
Tschumperlé proposes in [28] an efficient second-order anisotropic diffusion model for multi-valued image regularization and inpainting. Given a  $\mathbb{R}^N$ -valued image  $u$  known outside  $\Omega$ , and starting from an initial rough inpainting obtained by straightforward advection of boundary values, the pixels in the inpainting domain are iteratively updated according to a finite difference approximation to the equations

$$\frac{\partial u_i}{\partial t} = \text{trace}(T \nabla^2 u_i), \quad i \in \{1, \dots, N\}.$$

Here,  $T$  is the tensor field defined as

$$T = \frac{1}{(1 + \lambda_{\min} + \lambda_{\max})^{\alpha_1}} v_{\min} \otimes v_{\min} + \frac{1}{(1 + \lambda_{\min} + \lambda_{\max})^{\alpha_2}} v_{\max} \otimes v_{\max},$$

with  $0 < \alpha_1 \ll \alpha_2$ , and  $\lambda_{\min}, \lambda_{\max}, v_{\min}, v_{\max}$  are the eigenvalues and eigenvectors, respectively, of  $G_\sigma * \sum_{i=1}^N \nabla u_i \otimes \nabla u_i$ , being  $G_\sigma$  a smoothing kernel and  $\sum_{i=1}^N \nabla u_i \otimes \nabla u_i$  the classical structure tensor, that is known for representing well the local geometry of  $u$ . Figure 12 reproduces an experiment taken from Tschumperlé [28].



**Fig. 12.** An inpainting experiment (the middle image is the mask defined by the user) taken from Tschumperlé [28].

The approach of Auroux and Masmoudi in [3] uses the PDE techniques that have been developed for the inverse conductivity problem in the context of crack detection. The link with inpainting is the following: missing edges are modeled as cracks and the image is assumed to be smooth out of these cracks. Given a crack, two inpainting candidates can be obtained as the solutions of the Laplace equation with Neumann condition along the crack and either a Dirichlet, or a Neumann condition on the domain's boundary. The optimal cracks are those for which the two candidates are the most similar in quadratic norm, and they

can be found through topological analysis, i.e. they correspond to the set of points where putting a crack mostly decreases the quadratic difference. Both the localization of the cracks and the associated piecewise smooth inpainting solutions can be found using fast and simple finite differences schemes.

Finally, Bornemann and März propose in [9] a first-order model to advect the image information along the integral curves of a coherence vector field that extends in  $\Omega$  the dominant directions of the image gradient. This coherence field is explicitly defined, at every point, as the normalized eigenvector to the minimal eigenvalue of a smoothed structure tensor whose computation carefully avoids boundary biases in the vicinity of  $\partial\Omega$ . Denoting  $c$  the coherence field, Bornemann and März show that the equation  $c \cdot \nabla u = 0$  with Dirichlet boundary constraint can be obtained as the vanishing viscosity limit of an efficient fast-marching scheme: the pixels in  $\Omega$  are synthesized one at a time, according to their distance to the boundary. The new value at a pixel  $p$  is a linear combination of both known and previously generated values in a neighborhood of  $p$ . The key ingredient of the method is the explicit definition of the linear weights according to the coherence field  $c$ . Although the Bornemann-März model requires a careful tune of four parameters, it is much faster than the PDE approaches mentioned so far, and performs very well, as illustrated in Figure 13

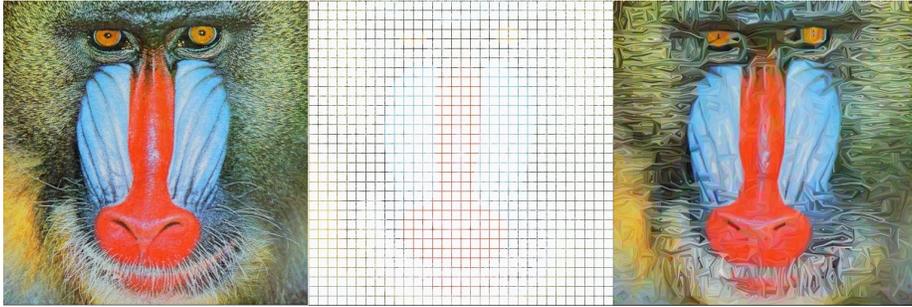


**Fig. 13.** An inpainting experiment taken from Bornemann and März [9], with a reported computation time of 0.4 sec.

### Combining and extending PDEs and patch models

In general, most PDE/variational methods that have been presented so far perform well for inpainting either thin or sparsely distributed domains. However, there is a common drawback to all these methods: they are unable to restore texture properly, and this is particularly visible on large inpainting domains, like for instance in the inpainting result of Figure 12 where the diffusion method is not able to recover the parrot's texture. On the other hand, patch-based methods are not able to handle sparse inpainting domains like in Figure 14, where no valid squared patch can be found that does not reduce to a point. On the contrary, most PDE/variational methods remain applicable in such situation, like in Figure 14 where the model proposed by Masnou and Morel [23] yields the

inpainting result. Obviously, some geometric information can be recovered, but no texture.

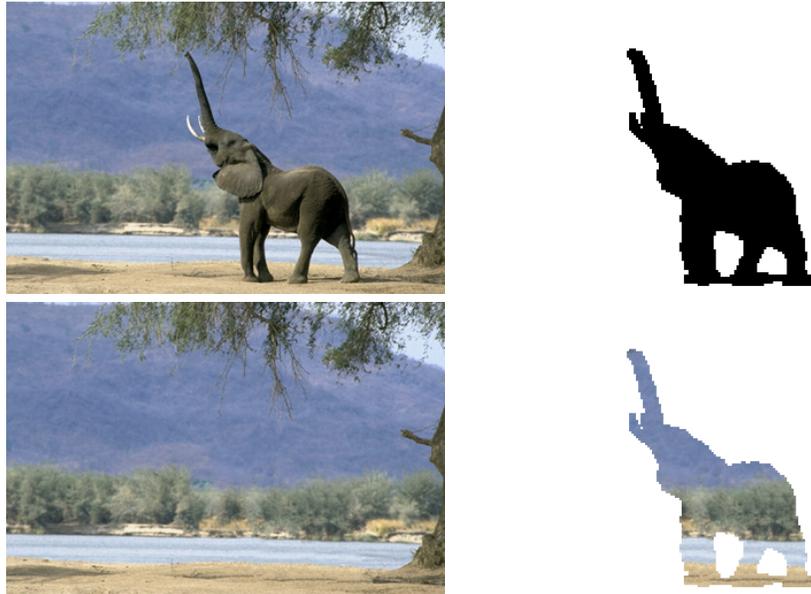


**Fig. 14.** A picture of a mandrill, the same picture after removal of  $15 \times 15$  squares (more than 87% of the pixels are removed), and the reconstruction with the method introduced by Masnou and Morel [23] using only the one-pixel wide information at the squares' boundaries.

There have been several attempts to explicitly combine PDEs and patch-based methods in order to handle properly both texture and geometric structures. The contribution of Criminisi, Pérez, and Toyama [12] was mentioned already. The work of Bertalmío, Vese, Sapiro, and Osher [8] uses an additive decomposition of the image to be inpainted into a geometric component that contains all edges information, and a texture component. Then the texture image is restored using the Efros and Leung's algorithm of [14], while the geometric image is inpainted following the method proposed in Bertalmío et al [7] (several subsequent works have proposed other methods for the individual reconstruction of each component). The final image is obtained by addition of the restored texture and geometric components. In a few situations where the additive decomposition makes sense, this approach does indeed improve the result and extends the applications domain of inpainting.

In Komodakis and Tziritas [20] the authors combine variational and patch-based strategies by defining an inpainting energy over a graph whose nodes are the centers of patches over the image. The inpainting energy has two terms, one being a texture synthesis term and the other measuring the similarity of the overlapping area of two neighboring patches (centered on nodes which are neighbors in the graph). By minimizing this energy with Belief Propagation, a label is assigned to each node, which amounts to copying the patch corresponding to the label over the position of the node. The results are very good on a variety of different images (e.g. Fig. 1) and the method is fast. Some potential issues: there is no assurance that the iterative process converges to a global minimum, and visual artifacts may appear since the method uses a fixed grid and entire patches are copied for each pixel of the mask.

The work by Drori, Cohen-Or, and Yeshurun in [13] does not involve any explicit geometry/texture decomposition, but the search for similar neighborhoods is guided by a prior rough estimate of the inpainted values using a multiscale sampling and convolution strategy, in the very spirit of Ogden et al [24]. In addition, in contrast with many patch-based methods, the dictionary of valid patches is enriched using rotations, rescalings, and reflections. An example extracted from Drori et al [13] is shown in Figure 15.



**Fig. 15.** An experiment from Drori et al [13] illustrating the proposed multiscale diffusion/patch-based inpainting method. The upper-left image is the original, the upper-right image contains the mask defined by the user, the bottom-left image is the result, and the bottom-right image shows what has been synthesized in place of the elephant.

### Beyond single image inpainting

All the methods mentioned above involve just a single image. For the multi-image case, there are two possible scenarios: video inpainting, and inpainting a single image using information from several images.

Basic methods for video inpainting for data transmission (where the problem is known as “error concealment” and involves restoring missing image blocks) and for film restoration applications (dealing with image gaps produced by dust, scratches or the abrasion of the material) assume that the missing data changes

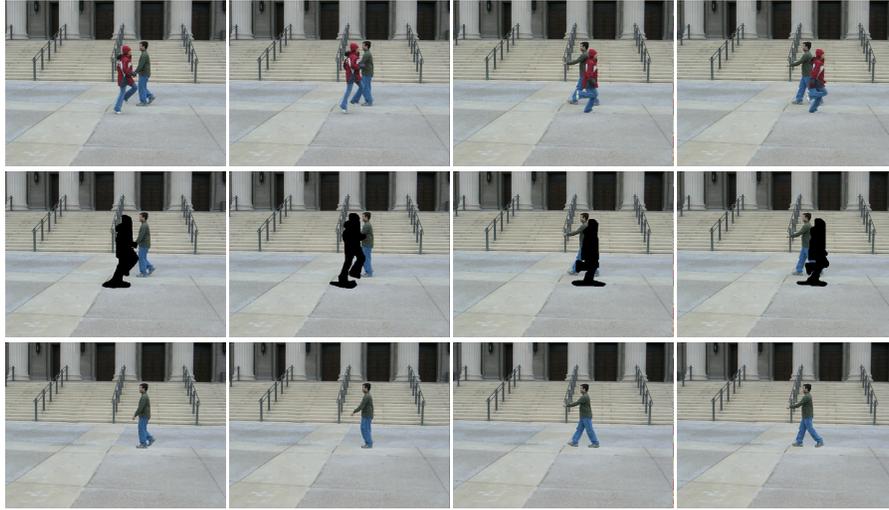
location in correlative frames, and therefore use motion estimation to copy information along pixel trajectories. A particular difficulty in video inpainting for film restoration is that, for good visual quality of the outputs, the detection of the gap and its filling-in are to be tackled jointly and in a way which is robust to noise, usually employing probabilistic models in a Bayesian framework, see for example the book by Kokaram [19].

Wexler et al. [29] propose a video inpainting algorithm that extends to space-time the technique of Efros and Leung [14] and combines it with the idea of coherence among neighbors developed by Ashikhmin [2]. First, for each empty pixel  $P$  they consider a space-time cube centered in  $P$ , compare it with all possible cubes in the video, find the most similar and keep its center pixel  $Q$ , which will be the correspondent of  $P$ . For each cube the information considered and compared is not only color but also motion vectors. Then, instead of copying the value of  $Q$  to  $P$ , they copy to  $P$  the average all the values of the shifted correspondents of the neighbors of  $P$ : for instance, if  $R$  is at the *right* of  $P$ , and  $S$  is the correspondent of  $R$ , then the pixel to the *left* of  $S$  will be involved in the average to fill-in  $P$ . This is based on the idea by Ashikhmin [2], see Fig. 5. The shortcomings of this video inpainting method are that the results present significant blur (due to the averaging), it seems to be limited only to static-camera scenarios (probably due to the simple motion estimation procedure involved) and periodic motion without change of scale, and the computational cost is quite high (due to the comparison of 3D blocks).

Shiratori et al. [26] perform video inpainting by firstly inpainting the motion field with a patch based technique like that of Efros and Leung [14] and then propagating the colors along the (inpainted) motion trajectories. The method assumes that motion information is sufficient to fill-in holes in videos, which isn't always the case (e.g. with a static hole over a static region). The results present some blurring, due to the bilinear interpolation in the color propagation step.

Patwardhan et al. [25] propose a video inpainting method consisting of three steps. In the first step they decompose the video sequence into binary motion layers (foreground and background), which are used to build three image *mosaics* (a mosaic is the equivalent of a panorama image created by stitching together several images): one mosaic for the foreground, another for the background and a third for the motion information. The other two steps of the algorithm perform inpainting, first from the foreground and then from the background: these inpainting processes are aided and sped-up by using the mosaics computed in the first step. See Figure 16 for some results. The algorithm is limited to sequences where the camera motion is approximately parallel to the image plane, and foreground objects move in a repetitive fashion and do not change size: these restrictions are imposed so that a patch-synthesis algorithm like that of Efros and Leung [14] can be used.

Hays and Efros [18] perform inpainting of a single image using information from a database with several millions of photographs. They use a scene-descriptor to reduce the search space from two million to two hundred images, those images from the database which are *semantically* closer to the image the user wants



**Fig. 16.** Top row: some frames from a video. Middle row: inpainting mask  $\Omega$  in black. Bottom row: video inpainting results obtained with the algorithm of Patwardhan et al. [25].

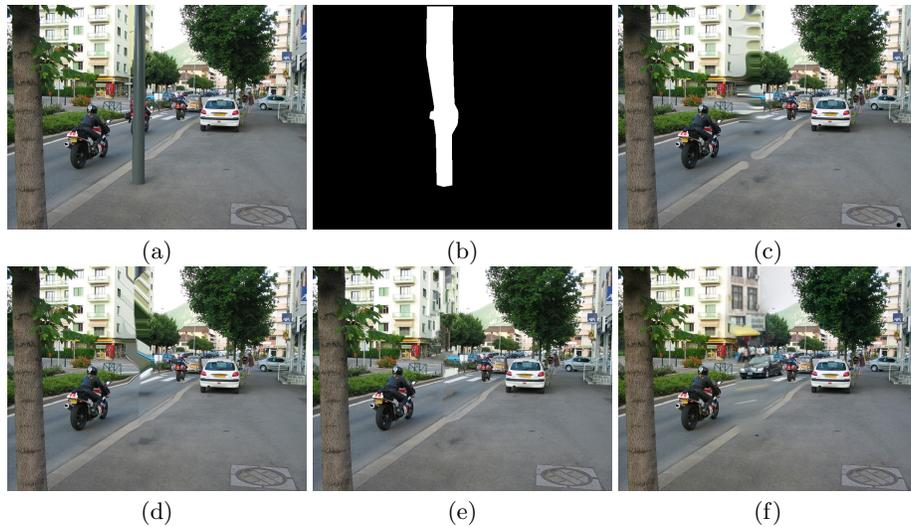


**Fig. 17.** Left: original image. Middle: inpainting mask  $\Omega$ , in white. Right: inpainting result obtained with the method by Hays and Efros [18], images taken from their paper.

to inpaint. Using template matching they align the two hundred best matching scenes to the local image around the region to inpaint. Then they composite each matching scene into the target image using seam finding and image blending. Several outputs are generated so the user may select among them, and the results can be outstanding, see Figure 17. The main shortcoming of this method is that it relies on managing and operating a huge image database. When the algorithm fails, it can be due to a lack of good scene matches (if the target image is atypical), or because of *semantic violations* (e.g. failure to recognize people hence copying only part of them), or in the case of uniformly textured backgrounds (where this algorithm might not find the precise same texture in another picture of the database).

## Open Problems

Inpainting is a very challenging problem and it is far from being solved, see Figure 18. Patch-based methods work best in general, although for some applications (e.g. very spread, sparsely distributed gap  $\Omega$ ) geometry-based methods might be better suited. And when the image gap lies on a singular location, with surroundings that can't be found anywhere else, then all patch-based methods give poor results, regardless if they consider or not geometry. For video inpainting the situation is worse, the existing algorithms are few and with very constraining limitations on camera and object motion. Because video inpainting is very relevant in cinema post-production, in order to replace the current typical labor intensive systems, important developments are expected in the near future.



**Fig. 18.** An example where no inpainting method seems to work. (a) Original image, from the database provided by Hays and Efros [18]. (b) In white, the mask to be inpainted, that is not the initial mask proposed by Hayes and Efros, but derives from the fuzzy mask actually used by their algorithm. (c) Result courtesy of D. Tschumperlé using the algorithm from [28]. (d) Result courtesy of T. März and F. Bornemann using the algorithm from [9]. (e) Result using a variant of the algorithm from Criminisi et al. [12]. (f) Result from Hays and Efros [18].

## Recommended Readings

- [1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on*

- signal processing*, 54(11):4311, 2006.
- [2] M. Ashikhmin. Synthesizing natural textures. *Proc. ACM Symp. Interactive 3D Graphics*, ACM Press, pages 217–226, 2001.
  - [3] D. Auroux and M. Masmoudi. A one-shot inpainting algorithm based on the topological asymptotic analysis. *Comp. Appl. Math.*, 25:1–17, 2006.
  - [4] C. Ballester, M. Bertalmío, 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, 2001.
  - [5] C. Barnes, E. Shechtman, A. Finkelstein, and D.B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3):2, 2009.
  - [6] M. Bertalmío, A. Bertozzi, and G. Sapiro. Navier-Stokes, fluid dynamics, and image and video inpainting. In *Proc. IEEE Int. Conf. on Comp. Vision and Pattern Recog.*, Hawaiï, 2001.
  - [7] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. SIGGRAPH'00, New Orleans, USA*, pages 417–424, 2000.
  - [8] M. Bertalmío, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003.
  - [9] F. Bornemann and T. März. Fast image inpainting based on coherence transport. *J. Math. Imaging and Vision*, 28(3):259–278, 2007.
  - [10] T.F. Chan, S.H. Kang, and J. Shen. Euler’s elastica and curvature based inpainting. *SIAM Journal of Applied Math.*, 63(2):564–592, 2002.
  - [11] T.F. Chan and J. Shen. Mathematical models for local deterministic inpaintings. *SIAM Journal of Applied Math.*, 62(3):1019–1043, 2001.
  - [12] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004.
  - [13] I. Drori, D. Cohen-Or, and H. Yeshurun. Fragment-based image completion. In *Proc. SIGGRAPH'03*, volume 22(3), pages 303–312, 2003.
  - [14] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision*, volume 2, page 1033. Citeseer, 1999.
  - [15] M. Elad, J.L. Starck, P. Querre, and D.L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Applied and Computational Harmonic Analysis*, 19(3):340–358, 2005.
  - [16] S. Esedoglu, S. Ruuth, and R. Tsai. Threshold dynamics for high order geometric motions. *Interfaces and Free Boundaries*, 2007.
  - [17] S. Esedoglu and J. Shen. Digital image inpainting by the Mumford-Shah-Euler image model. *European J. Appl. Math.*, 13:353–370, 2002.
  - [18] J. Hays and A.A. Efros. Scene completion using millions of photographs. *Communications of the ACM*, 51(10):87–94, 2008.
  - [19] A.C. Kokaram. *Motion picture restoration: digital algorithms for artefact suppression in degraded motion picture film and video*. Springer-Verlag London, UK, 1998.

- [20] N. Komodakis and G. Tziritas. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing*, 16(11):2649, 2007.
- [21] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53, 2008.
- [22] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *SIAM Multiscale Modeling and Simulation*, 7(1):214–241, 2008.
- [23] S. Masnou and J.-M. Morel. Level lines based disocclusion. In *5th IEEE Int. Conf. on Image Processing, Chicago, Illinois, October 4-7, 1998*.
- [24] J.M. Ogden, E.H. Adelson, J.R. Bergen, and P.J. Burt. Pyramid-based computer graphics. *RCA Engineer*, 30(5):4–15, 1985.
- [25] KA Patwardhan, G. Sapiro, and M. Bertalmío. Video inpainting under constrained camera motion. *IEEE Transactions on Image Processing*, 16(2):545–553, 2007.
- [26] T. Shiratori, Y. Matsushita, X. Tang, and S.B. Kang. Video completion by motion field transfer. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, 2006.
- [27] J. Sun, L. Yuan, J. Jia, and H.Y. Shum. Image completion with structure propagation. In *ACM SIGGRAPH 2005 Papers*, page 868. ACM, 2005.
- [28] D. Tschumperlé. Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's. *International Journal of Computer Vision*, 68(1):65–82, 2006.
- [29] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1. Citeseer, 2004.