SIAM J. IMAGING SCIENCES
Vol. 4, No. 4, pp. 1143–1179

Geometrically Guided Exemplar-Based Inpainting*

Frédéric Cao†, Yann Gousseau‡, Simon Masnou§, and Patrick Pérez¶

Abstract. Exemplar-based methods have proven their efficiency for the reconstruction of missing parts in a digital image. Texture as well as local geometry are often very well restored by such methods. Some applications, however, require the ability to reconstruct nonlocal geometric features, e.g., long edges. In order to do so, we propose to first compute a geometric sketch, which is then interpolated and used as a guide for the global reconstruction. In comparison with other related approaches, the originality of our work relies on the following points: (1) The geometric sketch computation is parameter-free and based on level lines, which provides a complete, reliable, and stable representation of the image. (2) The completion of the geometric sketch is fully automatic. It is done using a new—and interesting on its own—geometric inpainting approach that interpolates level lines with Euler spirals. Euler spirals are natural curves for shape completion and have been used already for edge completion and inpainting. It is the first time, however, that these curves are used for completing the whole level lines structure. (3) The general reconstruction is performed using a guided version of a classical exemplar-based method. However, we do not constrain the exemplar-based reconstruction to strictly follow the geometric guide. We actually use a new metric between blocks that consists of the sum of the classical L^2 metric between any two blocks of the general image plus an L^2 metric between the corresponding blocks in the completed geometric image. This is equivalent to a Lagrangian relaxation of a strictly guided reconstruction. We discuss in the paper the details of the method and some related mathematical issues, and we illustrate its efficiency on several examples.

Key words. inpainting, texture synthesis, exemplar-based inpainting, level lines, image geometry, Euler spirals

AMS subject classifications. 68U10, 94A08, 65D05, 65D17, 65D18, 65K10

DOI. 10.1137/110823572

1. Introduction and motivations.

The last 12 years have witnessed many contributions to the *inpainting* problem in digital images, i.e., the problem of recovering entire regions where the information either has been lost or is partially occluded by undesired objects. The two main applications of inpainting methods are image restoration (e.g., the suppression of scratches and blotches in old pictures and movies) and image manipulation (e.g., the removal of objects in photo editing or movie postproduction), and some inpainting strategies have also been used for digital zooming, color demosaicing, video deinterlacing, superresolution, and compression.

There are basically three categories of methods in the literature. The variational or PDE-

*Received by the editors February 7, 2011; accepted for publication (in revised form) August 5, 2011; published electronically December 1, 2011. This work was supported by ANR Project FREEDOM.
http://www.siam.org/journals/siims/4-4/82357.html
†DxO Labs, 92100 Boulogne-Billancourt, France (fcao@dxo.com).
‡Telecom ParisTech, LTCI CNRS, 75013 Paris, France (gousseau@telecom-paristech.fr).
§Université de Lyon, CNRS, Université Lyon 1, Institut Camille Jordan UMR 5208, 69622 Villeurbanne Cedex, France (masnou@math.univ-lyon1.fr).
¶Technicolor Corporate Research, Rennes Laboratory, F-35576 Cesson-Sévigné Cedex, France (Patrick.Perez@thomson.net).

based approaches enable in favorable cases a good reconstruction of the global geometry of an image but fail at correctly synthesizing the texture parts. In contrast, the methods of the second category were initially designed for texture synthesis. Some of these that we will describe later, called *exemplar-based*, can produce visually striking results but often do not permit the interpolation of nonrepetitive global image structures such as long edges. Finally, methods in the third category do not involve any explicit interpolation in the image domain but rather in one or several transform spaces, relying, e.g., on wavelets or Fourier transform. They usually perform well for sparse missing data or thin domains.

Recently, there have been several attempts to combine variational/PDE methods with exemplar-based algorithms in order to deal with those situations where the latter alone fail at recovering the geometry. This is the spirit of the method presented here that first completes the geometry for a simplified version of the image and then uses it as a weakly constraining guide for a complete reconstruction. More precisely, a sketch of the image is first obtained by selecting relevant level lines. Then, the missing part of the sketch is interpolated by prolongating interrupted level lines in a variational way. Eventually, the interpolated sketch is used as a guide in an exemplar-based inpainting method. A first advantage of the proposed methodology is that each one of these three steps relies on very few parameters. Second, it is to our knowledge the only approach permitting in favorable cases the simultaneous reconstruction of both the geometry and the texture over very large missing regions. This will be illustrated in the experimental section through challenging experiments.

2. State of the art.

2.1. Geometric methods. The different approaches to geometrical inpainting are all based on the prior that geometry can be recovered from the close neighborhood of the region to be filled (referred to as the *inpainting domain* hereafter). They divide into two categories:

1. the variational methods that involve a criterion of regularity of the reconstruction;
2. the pure PDE methods that try to diffuse and/or advect progressively the information from the boundary.

The first variational approach to geometrical inpainting was proposed in 1998 in [65] (see also [64]), following an inspiring work by Nitzberg, Mumford, and Shiota [69] on amodal completion and depth computing based on Kanizsa's vision theory [56]. The basic idea of [65] is that the geometry can be reconstructed in the inpainting domain by simply interpolating—using short and not too oscillating curves—all level lines that touch the domain boundary. This happens to be equivalent to minimizing a functional involving the curvature to a power p. Further details will be given in section 4. A globally minimizing scheme is proposed in [65] for the case $p = 1$, while the local minimization in the case $p > 1$ is addressed in [21] using a fourth-order equation. An interesting relaxed formulation of a slightly different model is proposed in [6] that yields a numerically more convenient second-order Euler–Lagrange equation.

A totally different approach has been proposed by Bertalmio et al. in [9], where the term *inpainting* has been proposed for the first time in this context. The idea is to mimic the way professionals do inpainting for real painting restoration. It consists in progressively advecting the valid information from the boundary of the inpainting domain inwards using a third-order advection-type equation. This equation transports the image values along continuations of

edges, an additional anisotropic diffusion being used to avoid shocks. An interesting interpretation of the model is given in [14], and the connections with the classical Navier–Stokes equation of fluid dynamics are shown in [8].

Using a more global approach, Chan and Shen propose in [22] a denoising/interpolation model based on the joint minimization of a quadratic fidelity term and a total variation criterion. This latter actually makes sense only for thin inpainting domains because its minimizers are images having the shortest level lines. A variant of the associated Euler–Lagrange equation is studied in [23].

Elder and Goldberg propose in [37] an approach to image editing based on the manipulation of the edge map. A possible application is the removal of objects in some simple situations: the corresponding edges are first removed from the edge map, the remaining edges are completed if necessary with, possibly, the help of the user, and the final image is obtained by a linear interpolation out of the edges.

Esedoglu and Shen propose in [39] interpolating in the inpainting domain using a piecewise smooth function that minimizes the Mumford–Shah functional with an additional penalization, so that the discontinuity set has small Euler elastica energy (see below), i.e., is short and not too curvy; see [38] for an interesting implementation of the model.

Grossauer and Scherzer study in [43] an inpainting model based on the Ginzburg–Landau equation. In [84], Tschumperlé and Deriche propose an efficient second-order anisotropic diffusion equation that preserves curvature and gives good results. The approach of Auroux and Masmoudi in [5] uses the PDE techniques that have been developed for the inverse conductivity problem in the context of crack detection. In [3, 76], a prior segmentation of the edges outside the inpainting domain is performed. Then edges are interpolated—using splines in [3], arc of circles in [76]—and the inpainting is completed using a smooth interpolation between edges. The tricky part of such an approach is the careful choice of pairs of edges that will be connected. Interestingly, the interpolation of large domains in [3] involves an additional *patch repetition procedure* that turns the method into one of the very first geometry-guided inpainting methods.

Finally, in an attempt to improve the fast but limited method proposed by Telea [81], Bornemann and März describe in [14] a first-order equation that advects the image information along the integral curves of a coherence vector field that extends inside the inpainting domain the dominant directions of the outside gradient. They propose a very fast numerical scheme based on fast marching. In terms of speed and quality of the results, it is one of the best PDE approaches to the inpainting problem, together with [84].

There is, however, a common drawback to all PDE or variational methods that we have presented so far, and that explains why they are used mainly for inpainting thin regions: they are unable to properly restore textured regions, in contrast to the approaches that we will now describe.

2.2. From texture synthesis to inpainting: Exemplar-based methods. For a long time, texture synthesis has been formulated as a problem of estimating and then sampling a probability distribution [11, 29, 92]. This approach may provide good results but is usually computationally expensive, and, more importantly, the choice of a suitable model is a very difficult task. De Bonet studies in [12] a multiresolution technique where interscale dependencies of

texture samples are constrained. Drawing their inspiration from psychophysics experiments, Heeger and Bergen [51] propose synthesizing a texture by constraining the marginals of overcomplete wavelet decompositions. This work has been followed by others in which not only the marginals but also more complicated statistics on wavelet decompositions are constrained; see [79]. All these methods may provide good results but not for all situations: highly structured patterns are often badly synthesized.

New models were proposed in the late 90's that sample the values of a new texture directly from a set of neighborhoods of a given sample image [33, 87]. Exploiting the locality and the stationarity at a certain scale of the texture, they achieve in favorable cases a degree of realism that was beyond the reach of previous approaches. The first celebrated algorithms in the class of exemplar-based methods, due to Efros and Leung [33] and Wei and Levoy [87], both involve the notion of *patch*, i.e., a square window of size $r \times r$. The basic idea is that in order to synthesize the value at a pixel whose neighborhood is partially known, one looks for similar patches in a sample image (when performing texture synthesis) or in a valid part of the image (when performing inpainting). The gray or color value at the pixel is simply chosen as the value of the central pixel of the most resembling patch or sampled from a set of very similar patches, the similarity between patches generally being measured with a weighted L^2 metric. Such approaches enable the reproduction of textures even when they are highly structured. They are, however, prone to either *garbage growing* (the production of erratic structures) or plain repetitions of the input sample. This core algorithm is proposed with different variants in [33, 87] (see also [50, 13]), involving, for instance, a multiscale approach and a careful synthesis order.

All these papers are essentially dedicated to texture synthesis, yet an example of inpainting is given in [33]. Previously, a *parametric* synthesis approach had been proposed in [54] for illustrating an inpainting application. To the best of our knowledge, the first explicit and systematic dedication to inpainting of a *nonparametric* synthesis method can be found in [13]. Despite the great improvements with respect to the previous contributions to texture synthesis, a well-known problem of the exemplar-based methods that we have mentioned comes from the synthesis of only one pixel at a time, which may sometimes result in "cycling" effects like the constant propagation of an erroneous synthesis or the formation of much too repetitive patterns. Surprisingly, because it amounts to reducing the space of interpolation possibilities, better results can be obtained by synthesizing not only the central pixel but entire patches. The use of such methods also considerably reduces the computational time. Many contributions have been proposed in this direction with several variants [34, 47, 61, 28, 32, 72, 55, 90, 59, 2]:

- on the location of the patches to be synthesized (using a fixed grid or not, with either no overlap, a smooth overlap, or a sharp overlap between adjacent patches),
- on the filling order (raster scan, concentric layers, incorporation of geometric or intensity constraints),
- on the searching domain (reinitialized at each iteration or constrained by the previously chosen sample patch),
- on the similarity measure between patches and the way to find a minimizing candidate (enumeration, belief propagation, etc.),
- on the artificial extension of the space of samples by introducing rotated and rescaled versions of the existing patches.

Recently, the *PatchMatch* method [7] that involves random associations between patches yielded spectacular improvements of the computational complexity. Let us finally mention the recent contributions [40, 73, 57, 1], where the exemplar-based inpainting is formulated as a variational problem with two unknowns, the image color and the similarity weights between pairs of points, resulting in a more tractable numerical task.

State-of-the-art exemplar-based methods are able to produce, in a very reasonable time, results that are often amazing. Most approaches are not only able to reproduce a texture but are also very good at restoring the local geometric information whenever it can be obtained elsewhere in the image. However, nonlocal geometric features cannot be well restored in general, which was the main motivation for this paper.

2.3. Optimizing in a transform domain. There have been in recent years several contributions where the inpainting problem is formulated as an optimization task in a transform domain, e.g., Fourier, wavelets, framelets, etc. It usually amounts to finding the reconstructed image whose representation in the transform domain is optimal. For instance, a simultaneous geometry/texture inpainting is performed in [36] (see also [45, 46]) by optimizing the decomposition in two different transform domains, one giving a sparse representation of the geometry (the curvelet domain) and the other adapted to the texture (the DCT domain). Other approaches in this category are [52, 17, 24]. All these methods are usually very efficient when the unknown part of the image is a sparse set of pixels. In cases where the inpainting domain is large, however, they are clearly outperformed by exemplar-based methods.

2.4. Combining texture synthesis and geometric reconstruction. Several papers on inpainting have been concerned with the combination of a PDE or variational model for the reconstruction of the geometry together with an exemplar-based algorithm for the reconstruction of texture. In [10], the image is first decomposed into a geometric component and a texture component using a technique described in [86]. The geometric component is inpainted by the model of [9], the texture component is restored using the Efros–Leung algorithm [33], and the final image is the sum of both reconstructions. An analogous approach is used in [74] in the very specific context of detecting and removing text on an image. A limitation when using such decompositions to perform the reconstruction is the nonindependence of the two components. Independently inpainting these two components yields artifacts.

There is no explicit geometry/texture decomposition in [32] but rather a prior fast estimate of the colors of the hidden region by a multiscale filtering approach that guides the exemplar-based inpainting. Let us mention that in a related direction, an early multiscale inpainting method relying on the Laplacian pyramid was proposed in [70]. Using a guide for the reconstruction is also the spirit of [31], except that the guide is obtained with the very efficient PDE model of [84] (that we presented above) for geometrical inpainting. Analogously, [75] proposes an algorithm for structure and texture restoration that incorporates a texture synthesis algorithm to the geometric algorithm described in [76].

The exemplar-based approach of [28] has been an inspiration for our paper. It indeed involves a local geometrical guiding through a priority term that forces the reconstruction of strong edges first. In the current paper, we build instead a global geometrical guiding, enabling the restoration of missing parts over large distances.

Similarly to the approach presented in this paper, several works propose performing in-

painting by relying on a segmentation of the image, either automatic or not, to infer the missing geometry. The method presented in [55] consists of a prior edge segmentation in the known part of the image followed by a completion of the edges in the inpainting domain. This yields a partition of the domain that guides the final reconstruction done with a tensor voting technique, yielding excellent results. In [80], the user manually draws the geometric structure in the inpainting domain; then exemplar-based inpainting is performed, first along the edges and then in the rest of the domain. Eventually, the method proposed in [25] also relies on a prior computation of a sketch, and the sketch edges are interpolated using straight lines or conic curves. This work is along the same lines as ours, but each part of the algorithm is described in an imprecise way, precluding its practical use. For a different purpose, the authors of [48] introduced a way to compute Marr's primal sketch associated with an image but also define a notion of sketchability and a way to resynthesize an image given a sketch and a parsimonious modeling of texture.

Last, the method proposed in [16] uses a decomposition of image into a texture component and a structure component [86], and combines the geometric model from [84] and the Efros–Leung model [33] to inpaint, using a local isotropy criterion to decide whether a point should be treated as texture or structure.

2.5. Contribution of this paper and reader's guide. The method presented in this paper is in the spirit of some approaches described above. Our motivation is to propose an algorithm that exploits the performances of the best exemplar-based methods in addition to the ability to recover long-range geometric structures, possibly more complex than straight lines. The first step of our method is the creation of a simplified representation of the valid part of the image, a sketch, that essentially contains the important geometric features (section 3). This is done by a suitable selection of a few level lines wherever the image is significantly contrasted. Following [30], a contrast is declared significant depending on a threshold that is automatically, and parameter-freely, computed over the image. In contrast with the method in [25], our sketch is easy and fast to compute, and it does not require any edge precompletion since all curves in the image are level lines and are therefore closed curves.

In a second step (section 4), the sketch is reconstructed within the inpainting domain using an inpainting method that is interesting on its own. It consists of the application to all level lines of the same principle that guides the amodal completion process, a well-known ability of our visual system. A similar strategy has been proposed in [65, 64] using straight lines (or shortest polygonal curves in case of nonconvex inpainting domain) for completing the level lines. The novelty here is the use of Euler spirals and the possibility of handling crossings. As will be further explained in section 4, Euler spirals are natural curves for shape completion. They have been used in [58] to design a simple inpainting model that consists of a prior completion of broken edges followed by a smooth interpolation in between the reconstructed edges. This method seems applicable only for images with few strong edges. In contrast, our model performs the completion of *all* level lines and is therefore applicable, in theory, to any image. Yet the completion principle that is the basis of our approach makes sense mostly for the recovery of geometric features but is essentially not adapted for textures.

A key property of the new inpainting model of section 4.3 is the ability to handle sharp discontinuities correctly, which is a very difficult task for any usual PDE approach. The

method in [65, 64] already had this ability, but the new method has the additional capacity to handle curvy discontinuities and discontinuity crossings.

We saw in the previous section that joint texture/geometry approaches are of two types: the class of methods that independently process the geometry component and the texture component, and the class of methods that use the prior reconstruction of the geometry as a strict guide for the final reconstruction. We propose in section 6 a kind of Lagrangian relaxation of the guiding, where the completed geometric image is used in an energy term that is added to the classical patch metric of [72]; see section 5. We will provide the reader with a few experiments showing that this Lagrangian guiding significantly improves the results when some long-range, global geometry indeed has to be reconstructed, while it is more flexible than other approaches where the guiding is strict.

We also address in this paper a theoretical question. As we previously mentioned, the exemplar-based methods seem to have the ability to restore the geometry only *locally*. This is, however, a heuristic statement that had not been theoretically justified so far. We give a partial theoretical argument in that direction in the appendix, where we investigate the ability to asymptotically recover a curve starting from patches made of straight lines.

Overall, the method presented in this paper allows, with very few parameters, the inpainting of a rich class of images containing texture and long-range geometry with possibly sharp, curvy, and mutually crossing discontinuity lines. The experimental results that we obtain confirm the pertinence of our model.

3. Computing a sketch of the image. By a *sketch* of a gray level image (the color case will be addressed later on), we mean a piecewise constant approximation obtained with a suitable segmentation procedure. The segmentation used in this paper is derived from the *topographic map* of the image [19] and, more precisely, from the *meaningful level lines* defined in [30] (other segmentation procedures could be used to compute a sketch). Recall that the topographic map of a gray level image I is the collection of all its level lines that we define here as the connected components of the boundaries of upper levels sets $X_t(I) = \{x : I(x) \geq t\}$ (one could equivalently consider the lower level sets). Since

$$I(x) = \sup\{t : x \in X_t(I)\}, \tag{3.1}$$

the image can be completely reconstructed from its level lines. This complete representation has several advantages. First, it is invariant with respect to local increasing contrast changes, and, second, the topographic map can be associated with a nested tree structure that is very useful not only for image segmentation [30] but also for denoising [67], shape extraction, and comparison [63] or image compression [42].

A sketch of the initial gray level image is obtained by keeping only the most significant level lines, i.e., those that are contrasted enough according to an *a contrario* criterion [30]. Roughly speaking, meaningful lines are lines having a contrast that is very unlikely to be encountered in a white noise image. This amounts to keeping lines L for which

$$N.H(c(L))^{l(L)} \leq \epsilon, \tag{3.2}$$

where N is the total number of level lines in the image, H is the cumulative histogram of the modulus of the gradient over the image, $l(L)$ is the length of the line, $c(L)$ is the minimum

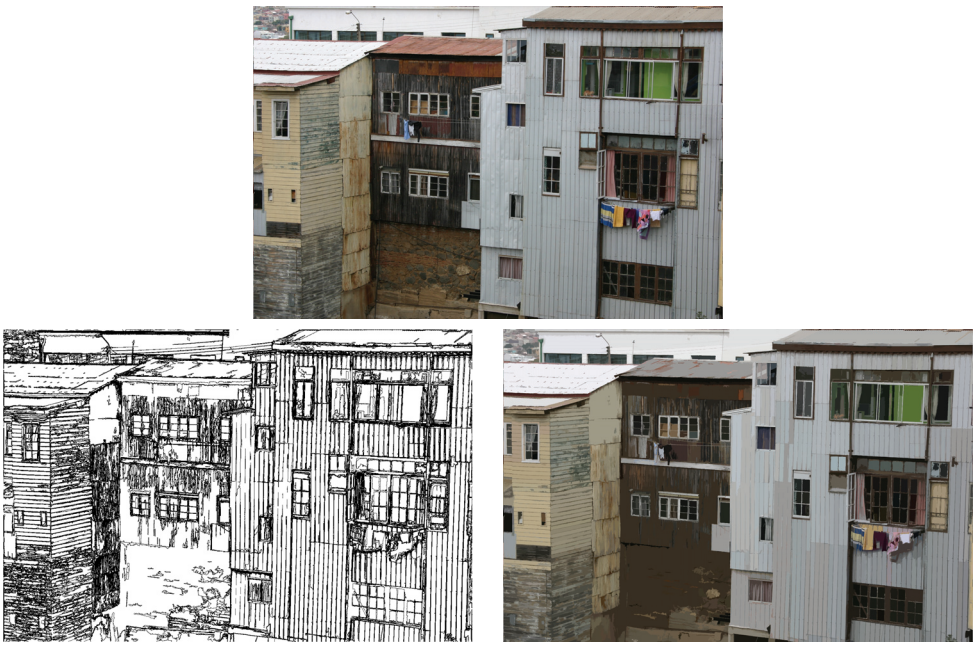

Figure 1. *A digital image (top), its meaningful level lines (bottom left), and the sketch reconstructed only from these lines (bottom right). The texture has partly disappeared, but the geometric structures and the dynamics remain.*

contrast along the line, and ϵ is a positive number. In practice, we use a refinement of this approach as presented in [18]. It consists in using a local estimation of the cumulated gradient histogram H to assert the meaningfulness of a curve. We chose to use this refinement because it yields less detection in textured area, a desirable property for the application considered in this paper. We took $\epsilon = 1$ for all the experiments to be displayed in this paper, drawing on the robustness of *a contrario* approaches to the choice of this parameter. Eventually, we obtain a piecewise constant image (a sketch) from the collection of selected lines as follows. Relying on the nested structure of the topographic map, we simply define the value of the sketch at a pixel x as the level associated with the shortest level line that encloses x.

Note that extracting the most significant level lines does not mean extracting the objects in the image but constructing a piecewise constant image whose dynamics is very close to the original. However, as observed in [63], boundaries of objects usually locally coincide with pieces of level lines, so that the most salient objects will appear in the sketch representation.

The above procedure for computing a sketch is valid for gray level images only. It must be emphasized, though, that most of the geometric information of real world color images is carried by the luminance channel [20] (in classical chrominance/luminance representations, such as Lab or YUV). For the exemplar-based method to be presented in section 5, a geometric guide will be computed by interpolating a gray level sketch. This guide will be obtained from the luminance channel of the original image to be inpainted (the YUV color model has been used in all experiments presented below).

An example of a sketch is displayed in Figure 1. For the purpose of visualization, each

piecewise constant region of the sketch has been given the average color value computed over the corresponding region in the original image. One may observe in this experiment that microtextures and many small details do not appear in the sketch, while the geometry and macrotextures remain, which is desirable from the perspective of the "geometric" inpainting methods to be introduced in the next sections. Nevertheless, one cannot guarantee that the sketch image will always be purely geometric: Figure 2 illustrates a situation where some useless details still remain. In this very case, it does not alter the quality of the final inpainting results that can be seen in Figure 18.

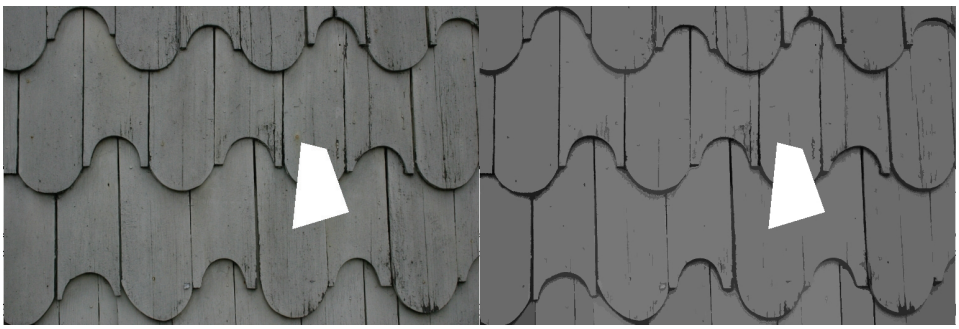

Figure 2. *The sketch image (right) associated with this picture of a wood wall is not perfectly geometric. It still contains a few useless details. Nevertheless, the final inpainting results are of good quality, as can be seen in Figure 18, where cropped subimages are shown.*

The situation in Figure 3 is more critical: branches and leaves form an in-between micro-/macrostructure that is partially captured in the sketch image. The resulting image is too complex to be properly inpainted by a geometric method, as shown in Figure 15.

At this point, it is worth noticing that any segmentation method enabling the creation of a piecewise constant image could be used to produce the sketch, possibly yielding better results, depending on the image. We chose the above method based on meaningful level lines for three reasons: first, it gives fair results for the various experiments that we have done. Second, this method relies on level lines, like both geometric inpainting methods to be introduced in the following sections. Finally, the *a contrario* approach enables us to perform all experiments using the same setting, namely $\epsilon = 1$, for producing the sketch.

The next section is devoted to the description of two geometry-oriented inpainting methods that will be used for the reconstruction of the sketch image. The first method relies on straight lines (section 4.2), while the second one relies on Euler spirals (section 4.3). Both are based on the interpolation of the broken level lines.

4. Level line completion, image inpainting, and sketch reconstruction. We describe in this section two *geometrical* inpainting methods, i.e., two methods essentially devoted to the reconstruction of geometry. The first one, introduced in [65], is based on the completion of level lines by straight curves (or polygonal lines when the inpainting domain is not convex), whereas the second approach is new and uses Euler spirals as completion curves. We will see at the end of the section and further in the paper that both methods can be used for the reconstruction of a sketch image aimed to be a geometric guide for the global inpainting algorithm to be presented in section 6.

Figure 3. *Branches and leaves form a micro-/macrostructure that is not properly captured in the sketch image. This is a typical situation where a pure patch-based approach is better than a geometry-guided approach, as illustrated in Figure 15.*

Before proceeding, let us explain how the methods, initially designed for gray level images, can be applied to color images. For these images, the notion of level line is not defined. Given a method that interpolates the broken level lines of a scalar image, the reconstruction of a color image will be performed by independently processing the luminance and both chrominance channels in a suitable luminance/chrominance representation (YUV, Lab, etc.). Note that using a luminance/chrominance representation instead of the usual RGB is crucial for limiting the creation of false colors. Even though the methods will be applied on a gray level sketch in the present paper, such an extension to color images may have an interest in other contexts.

We now show how an inpainting method can be derived from shape completion techniques.

4.1. From amodal completion to inpainting. Recall that, given a regular curve γ in the plane parameterized by arc-length, the *elastica* functional[1]

$$\int_0^{\mathcal{L}(\gamma)} |\kappa_\gamma(s)|^2 ds, \tag{4.1}$$

where κ_γ denotes the curvature and $\mathcal{L}(\gamma)$ the length of the curve, measures the elastic energy stored by the curve. It was first studied in 1744 by Euler in his work on the physics of thin rods. Much later, it was proposed by Ullman [85] and Horn [53] in the context of shape completion. Recall that our visual system has the ability to reconstruct missing edges when objects are partially occluded; see the important contribution of Kanizsa [56] on the so-called *amodal completion*. In terms of computer vision, the problem can be rephrased as follows: what is the most "pleasing" curve that joins two points with prescribed tangent boundary conditions?

[1]More general versions are $\int(\alpha + \beta|\kappa_\gamma(s)|^2)ds$.

The elasticae, i.e., the curves minimizing the elastica energy, are natural candidates since they are the smoothest curves in regard to the accumulated elastic energy. In addition, as shown by Ullman [85], they fulfill several reasonable axiomatic requirements (isotropy, locality, smoothness, etc.)

Elasticae are, however, hard to compute, for they do not admit any analytical expression [68]. There have been many attempts to derive efficient approximation methods, either for the elastica energy itself or for variants such as $\mathcal{L}(\gamma)^{-1}\int_0^{\mathcal{L}(\gamma)}|\kappa|^2 ds$, which is scale invariant; see [60, 69, 78, 82, 83, 91, 26, 15, 77, 35].

Although nice completions can be obtained with the elastica model, it has some drawbacks, such as the computational issues we have mentioned but also the incapacity to recover angles or circles or the nonexistence of a minimizer in some extreme boundary conditions [27, 62]. Other models have therefore been proposed, considering, however, that no model can *always* provide the perfect solution, essentially because the problem is ill-posed: the question of whether a curve is "pleasing" or not is somewhat subjective, and nothing totally reliable can be said in some extreme situations. We mention, however, two other models that yield interesting solutions in most common situations:

1. The minimal path/good continuation model of Fantoni and Gerbino computes an intermediate solution between the straight line and the polygonal line obtained by simply continuing the tangents from each extremity; see [41], where an exhaustive survey of completion models is provided.
2. The Euler spiral—also called Cornu's spiral or the clothoid loop—has been studied by Euler in his work on freely coiled up elastic springs and accurately plotted by Cornu in 1874. In the relaxed position of the spring—modeled as a curve γ—the scalar curvature at each point satisfies $\kappa(s) = C_1\, s + C_2$, where C_1, C_2 are constants and s is the arc-length. Euler spirals have already been used in the context of shape completion and for the inpainting of simple configurations; see, for instance, [58].

Splines are other somewhat natural candidates, and they have been used very frequently for curve interpolation; see, for example, [3]. Out of the lack of rotational invariance, they are actually less pleasing visually, in the context of curve completion, than Euler elasticae or spirals; see, for instance, the interesting [49] on three-dimensional curve completion.

Let us now come to the issue of computational complexity. Completing one broken curve using the Fantoni–Gerbino model or using Euler spirals is not difficult. A more difficult task is to complete many curves because there are $n!$ possible pairwise connections of $2n$ interrupted lines. Is there a particular family of connections that makes sense in the context of image inpainting? Can it be obtained without using an algorithm with exponential complexity? Let us start with a model that has been initially proposed in [65, 64] and that involves straight lines as very basic interpolating curves.

4.2. A simple inpainting algorithm using straight lines [65, 64]. The algorithm described in [65, 64] is shown in Table 1. It provides, for each channel in a chrominance/luminance representation of a color image (e.g., YUV), a *global* discrete minimizer of an energy that penalizes the integrated absolute curvature, not the squared curvature, along interpolating level lines. The algorithm involves the notion of a *causal* set of connections between *compatible T-junctions* of a gray level image. We introduce both notions in the following definition. For

Table 1

Algorithm from [65, 64] for the geometrical inpainting of images using straight lines without crossings.

INPAINTING WITH STRAIGHT LINES (WITHOUT CROSSINGS) [65, 64]

Input: A color image I known outside an inpainting domain A.

Output: An inpainted image $\tilde{I}$.

For each channel in a luminance/chrominance representation of I:

1. extract all T-junctions on ∂A, i.e., termination points of level lines (see Definition 4.1);
2. compute the associated tangents, either by a simple averaging approach like in [65, 64] or using a more robust method (see section 4.3 and [14]);
3. use dynamic programming to find a collection of **shortest** curves $\{\gamma_i\}_{i\in\{1,\dots,N\}}$ that realize a causal set of pairwise connections between compatible T-junctions γ_i^1, γ_i^2 (see Definition 4.1) and minimize a linear combination of the curves' total length and total curvature, including the angles at the extremities, i.e., the criterion

$$\sum_{i=1}^{N}\left(\int_{\gamma_i}(\alpha+\beta|\kappa|)ds+\beta(\text{angles at }\gamma_i^1\text{ and }\gamma_i^2)\right); \tag{4.2}$$

4. find the piecewise constant function $\tilde{I}$ canonically associated with these interpolating level lines.

simplicity, we assume that the gray level image I, defined outside the inpainting domain A, takes finitely many values, on a finite number of components, which is natural when dealing with images. Note, however, that the definition could be extended, with additional technical details, to the class of functions of bounded variation.

We will also assume that A is strictly contained in Ω. For the treatment of inpainting regions that touch the image borders, a prior naive one-pixel extension of the image domain can be performed that appears to be sufficient in practice: every valid pixel along the image border is replicated towards the exterior. Every other pixel to be set (those that are connected to a border pixel belonging to the inpainting region) is given the value of the closest valid border pixel.

Definition 4.1 (T-junctions, compatibility, and causality). *We call a* T-junction *any point $x \in \partial A$ on the inpainting domain's boundary that is a termination point for the level line $\partial\{y \notin A,\ I(y) \geq I(x)\}$. This notion is the counterpart for the level line framework of the notion of T-junction introduced by Kanizsa [56] to name the points where the edge of a partially occluded object encounters the edge of an occluding object.*

Being ∂A oriented counterclockwise, two T-junctions $x, y \in \partial A$ are compatible *if $I(x) = I(y)$ and $(\nabla I(x)\cdot T_{\partial A}(x))(\nabla I(y)\cdot T_{\partial A}(y)) < 0$, where $T_{\partial A}(x)$, $T_{\partial A}(y)$ denote the unit positively oriented tangents to ∂A at x and y, respectively. The compatibility property ensures that a curve joining x and y within A can be considered as the interpolation of a level line; see Figure 4.*

A system of pairwise associations between compatible T-junctions is said to be causal *whenever there exists a collection of curves realizing the connections without crossings (see*

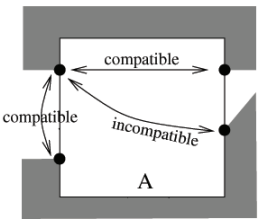

Figure 4. *T-junctions are marked with black dots. Two T-junctions are compatible if any curve that connect them within the inpainting domain can be considered as an interpolating level line; this is equivalent to assuming that image values at the T-junctions are the same and that image gradients have compatible orientation.*

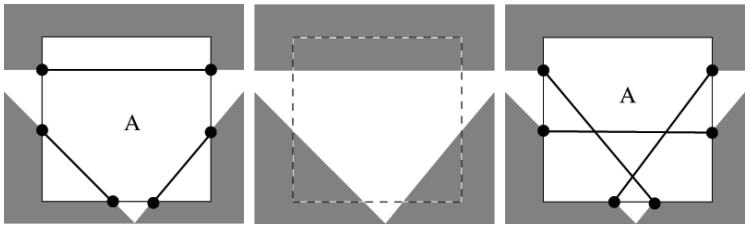

Figure 5. *Left: a causal system of pairwise connections between T-junctions: there exists a collection of curves that realize the connections without crossings, for instance the collection of shortest paths as in the figure. Middle: one can easily deduce a function from a causal system of connections between T-junctions and from a collection of noncrossing curves realizing the connections. Right: for a noncausal system of connections, there exists no system of curves that realize the connections without mutual crossings between some of them.*

Figure 5*).*

Causality, minimality, and existence of crossings are different notions: it is easy to build an example of a causal system of curves that realizes the connections without crossings but is not optimal. One can also build an example of a noncausal system of associations between compatible T-junctions and a related set of curves (necessarily with crossings) that minimizes criterion (4.2).

However, it is easy to check that, given a causal set of associations between compatible T-junctions, the shortest paths that realize the connections *do not mutually cross*. Moreover, among all collections of curves realizing this same set of connections, *the shortest paths have minimal energy*. This is obvious regarding the length term. As for the curvature term (i.e., the integrated absolute curvature plus angular defects), it follows from the fact that along any curve joining two T-junctions and living in A, the angular total variation (which is exactly the curvature term plus the angles at extremities) is necessarily not smaller than along the shortest path. Note that this property is no more true if another power of curvature is used.

Eventually, given a causal system of pairwise associations between T-junctions and a family of curves, not necessarily straight lines, that do not mutually cross and that realize the connections, it is easy to deduce an interpolated function. Each curve can be seen as an interpolating level line. For each value $t \in \{t_1, \ldots, t_M\}$ taken by the function I on ∂A, one defines A_t as the set enclosed by all the interpolating level lines at level t. By causality, the sets A_t are nested; i.e., $A_t \supset A_s$ if $t < s$. Indeed, if it were not the case, it would imply that some interpolating curves have crossings, which contradicts the causality assumption. Then,

one defines the inpainting for each $x \in A$ as

$$I(x) = t_1 + \sum_{i=1}^{M-1} (t_{i+1} - t_i) \mathbb{1}_{A_{t_i}(x)}.$$

In the next section, we will present a new method that yields an inpainting solution in the case where crossings between interpolating curves are allowed, the set of connections being causal or not.

In practice, how can one build a causal system of connections between compatible T-junctions and how can one find a causal system of curves (actually of straight lines; see above) that minimizes (4.2)? It suffices to observe that, once two compatible T-junctions have been associated, they induce a partition of ∂A into two intervals. The only way to maintain the causality is to impose that only T-junctions of the same component can be associated and to iterate the construction. There is a direct link between the minimization under this causal principle and dynamic programming: the size of a partition interval can be identified with a dynamic variable, and the minimization within an interval of ∂A is independent of the minimization within the complementary interval [65, 64]. Therefore, despite the limitations of the model (optimal curves are straight lines if the inpainting domain is convex, polygonal lines otherwise), it produces interesting results in a reasonable time. Instead of an exhaustive search with exponential complexity, the *global minimization* of (4.2) can be performed by dynamic programming in polynomial time. More precisely, the algorithm proposed in [65, 64] runs in $\mathrm{O}(N^3)$ with respect to the number N of T-junctions, thus approximately in $\mathrm{O}(M^{\frac{3}{2}})$ with respect to the number M of pixels in the inpainting domain.

In the next section, we describe a new geometrical inpainting approach using Euler spirals for the completion of broken level lines. We already mentioned that Euler spirals have been used in [58] for the design of a simple inpainting model. In contrast, our method can be used in much more general situations and, because all partially occluded level lines are interpolated, the resulting inpainted image is richer than what may be produced with [58]. In particular, our approach can handle situations where the natural completion produces crossings between edges.

4.3. A new inpainting method using Euler spirals. Let us first discuss the computation of a Euler spiral joining two points with prescribed tangents at both ends. Without other constraints, in particular if loops are allowed, there are infinitely many candidates. We shall therefore select a spiral that minimizes the Euler elastica energy. Interestingly, there is actually a close connection between elasticae and Euler spirals as explained in [27]. Parameterizing a curve γ by arc-length and denoting as ψ the inclination at $\gamma(s)$, i.e., the oriented angle between the x-axis and the tangent $\gamma'(s)$, the elastica functional reads as

$$\int_0^{\mathcal{L}(\gamma)} |\psi'(s)|^2 ds, \tag{4.3}$$

from which the following Euler–Lagrange equation can be derived, taking into account the transversality condition due to the unknown length $\mathcal{L}(\gamma)$ [27]:

$$(\psi'(s))^2 = \lambda \cos \psi(s) + \mu \sin \psi(s). \tag{4.4}$$

After linearization, one gets $(\psi'(s))^2 = \lambda + \mu\psi(s)$ whose solutions satisfy

$$\psi(s) = \frac{\mu}{4}(s-\alpha)^2 - \frac{\lambda}{\mu}, \tag{4.5}$$

where α is a constant of integration. Since the scalar curvature coincides with $\psi'(s)$, we deduce from the derivation of (4.5) that its solutions are Euler spirals, i.e., curves whose curvature is an affine function of arc-length. Therefore, Euler spirals are the solutions of the linearization of the Euler–Lagrange equation associated with the elastica energy. Euler spirals have interesting properties: they include circles, they are rotationally invariant in contrast to splines, and they can serve as C^2 transitions between a segment and an arc of circle, which explains why they are frequently used in very concrete applications, e.g., architecture, road design, typography, and even for the design of roller coasters! Integrating (4.5), one obtains that the curve can be parameterized as

$$(x(s), y(s)) = (x_0, y_0) + \left(\int_0^s \cos(\psi(t))dt, \int_0^s \sin(\psi(t))dt\right). \tag{4.6}$$

This expression involves the Fresnel integrals

$$C(x) = \int_0^x \cos(t^2)\,dt \quad \text{and} \quad S(x) = \int_0^x \sin(t^2)\,dt$$

that can be easily approximated from their Taylor expansion. Different methods have been proposed to find a numerical approximation of the Euler spiral joining two points with tangent boundary conditions [66, 15]. Here we use the rather simple approximation algorithm proposed in [27] that involves an iterated Newton scheme. Additional constraints are necessary to avoid solutions with loops, but even in that case uniqueness is not always guaranteed, at least in a few extreme situations: one can easily find two spirals with the same energy that join the point $(0,0)$ to $(1,0)$ with initial direction $(1,0)$ and final direction $(-1,0)$. In this kind of extreme situation, one may force a particular solution by imposing to the tangent along the curve a suitable range of allowed directions.

Let us now describe the adaptation to Euler spirals of the algorithm presented in the previous section. A first important point—for it influences the shape of the spirals a lot—is a careful computation of the tangent at each T-junction of the "broken" level lines. We follow the method due to Bornemann and März [14] after the work of Weickert [88, 89] on the robust determination of coherence directions in an image. Denote as Ω the image domain, A the inpainting domain, and I an image on $\Omega \setminus A$. The coherent direction at $x \notin A$ is defined as the normalized eigenvector associated with the minimal eigenvalue of the structure tensor [14]:

$$J(x) = \frac{\Big(K_\rho * \Big(\mathbb{1}_{\Omega\setminus A} \nabla I_\sigma \otimes \nabla I_\sigma\Big)\Big)(x)}{\Big(K_\rho * \mathbb{1}_{\Omega\setminus A}\Big)(x)}, \tag{4.7}$$

where $\mathbb{1}_{\Omega\setminus A}$ is the characteristic function of $\Omega \setminus A$, K_ρ is a Gaussian smoothing kernel with standard deviation ρ, and I_σ is defined by

$$I_\sigma = \frac{K_\sigma * \Big(\mathbb{1}_{\Omega\setminus A} I\Big)}{K_\sigma * \mathbb{1}_{\Omega\setminus A}}. \tag{4.8}$$

It can be experimentally observed that this computation yields a reliable estimation of the level line directions at the T-junctions (a sensible choice for the convolution parameters is $\sigma = 1.5$, $\rho = 4$).

The second important point is how one can handle several possibly crossing Euler spirals. We mentioned in the previous section that such a problem does not occur with straight lines if the set of connections is causal. If we now use Euler spirals, it is easily checked that even the causality constraint no longer guarantees that any two curves will not mutually cross. In addition, there are cases where noncausal connections, and therefore crossing curves, *are* the right solution (see Figure 6). Since crossing situations can be somewhat puzzling, as in Figure 7, the definition of an enclosed set from a collection of interpolating lines must be done carefully. This is the purpose of the method presented in Table 2 and illustrated in Figure 7. This method is the core of our new inpainting algorithm for the restoration of gray level images based on Euler spirals that is described in Table 3. Again, the generalization to color can be done by independently processing the three channels in a luminance/chrominance color space.

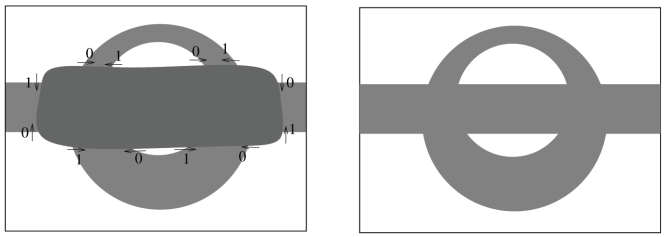

Figure 6. *A binary situation (left) where one can reasonably expect a completion with crossing curves (right). The left figure shows how T-junctions are oriented in the algorithm of Table 2.*

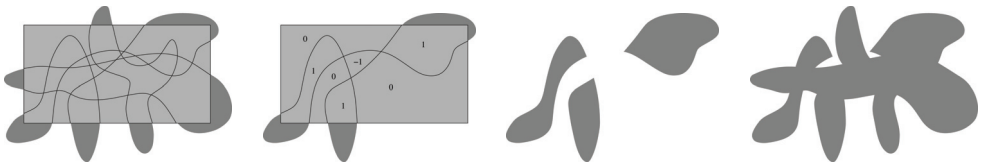

Figure 7. *An illustration of how the algorithm of Table 2 works in a puzzling situation where some interpolating curves mutually cross (left figure) and make the reconstruction difficult. The algorithm of Table 2 provides a systematic way to decide whether a point of the inpainting domain should be colored in gray or in white. The left figure shows several T-junctions connected by curves. In this situation, the algorithm of Table 2 yields three cycles, one of them being represented in the second figure, with the corresponding winding numbers. The third figure shows the set of points with a positive winding number. Doing the same for both other cycles and gluing the corresponding enclosed sets yields a reasonable result.*

The reason why we use formula (4.12) comes from a classical result: any measurable positive function $I : \Omega \to \mathbb{R}$ satisfies, at almost every point $x \in \Omega$,

$$I(x) = \int_0^{+\infty} \mathbb{1}_{\{y:\, I(y) \geq t\}}(x) dt. \tag{4.9}$$

Table 2

A METHOD FOR CONSTRUCTING AN ENCLOSED SET FROM A COLLECTION OF ORIENTED CURVES JOINING PAIRED COMPATIBLE T-JUNCTIONS, POSSIBLY WITH CROSSINGS

Inputs: A collection of T-junctions on ∂A, $\{j_1, j_2, \ldots, j_{2N}\}$, each of them endowed with an orientation, 0 or 1, depending on the orientation of the gradient on the associated level line. We take as a convention that, being ∂A oriented counterclockwise, the orientation at a junction j_i is 1 whenever the gradient at j_1 is oriented like ∂A, 0 otherwise (see Figure 6).
A collection of N curves, $\{\gamma_1, \ldots, \gamma_N\}$, joining pairs of *compatible* T-junctions, i.e., with opposite $0-1$ orientations (see Figure 6).

Output: A set $A_\gamma \subset A$ "canonically" enclosed by the curves and representing an interpolation in A of the outer level set associated with the level lines.

1. Step 1 : build a collection of closed curves $(\mathcal{C}_i)$ linking T-junctions with alternating labels 1, 0, 1, 0, etc., and obtained by gluing alternatively a curve γ_i and a piece of ∂A. The building process is as follows. Pick a T-junction with label 1, call it t_1, and let $n = 1$:
   (a) Find the next T-junction on ∂A (walking counterclockwise), which necessarily has label 0; call it t_{n+1}; $n \leftarrow n+1$.
   (b) Find the T-junction linked by a curve γ_i to t_n; call it t_{n+1} (it has label 1); $n \leftarrow n+1$.
   (c) If $t_n = t_1$, stop; else go to step (a).
   One can check that such a process will necessarily build a maximal cycle. Call it $\mathcal{C}_1$, and remove all visited T-junctions from the list. Start from a remaining T-junction with label 1 to build $\mathcal{C}_2$, etc.
2. Step 2 : for each $\mathcal{C}_i$ define its canonical interior A_i as all points in A with a positive winding number (index) with respect to $\mathcal{C}_i$. There are many ways to compute the winding number at a point with respect to a discrete oriented curve, not necessarily simple. We use the simple oriented crossings-check procedure of [44][a].
3. Step 3 : let $A_\gamma = \bigcup_i A_i$.

[a]See also http://softsurfer.com/Archive/algorithm_0103/algorithm_0103.htm for a pseudocode.

Obviously, if u takes values in $[m, M]$, then

$$I(x) = m + \int_m^M \mathbb{1}_{\{y:\, I(y) \geq t\}}(x)\, dt, \tag{4.10}$$

and (4.12) follows. Note, however, that, contrary to what happens with the level sets $\{y : I(y) \geq t\}_{t \geq 0}$ of the function I, our family of reconstructed sets A_t need not be nested; therefore (4.12) is not the exact counterpart of (4.9). Yet our tests show that it is a reasonable choice, much more sensible, for instance, than the formula $I(x) = \sup\{t : x \in A_t\}$, which is very sensitive to completion errors that might occur at the highest levels.

4.4. Results and application to sketch reconstruction. We present in Figures 8–10 three examples of geometrical inpainting using either straight lines or Euler spirals. We illustrate

Table 3

Algorithm for the inpainting of gray level images with Euler spirals. The inpainting of color images can be done by independently processing each channel in a luminance/chrominance representation, e.g., YUV.

INPAINTING WITH EULER SPIRALS (AND POSSIBLY CROSSINGS)

Input: An image $I : \Omega \setminus A \to \mathbb{R}$ with $A \subset \Omega$ an (open) inpainting domain.

Output: An inpainted image $\tilde{I} : \Omega \to \mathbb{R}$.

1. Assume that I takes values $\{N_1, \dots, N_{n+1}\}$ on ∂A. Extract all T-junctions on ∂A, i.e., points of $\partial\{y \in \partial A,\, I(y) \geq t\}$ for $t \in \{N_2, \dots, N_{n+1}\}$.
2. Compute the tangents of the associated level lines by the Bornemann–März–Weickert method described above.
3. For each level $t \in \{N_2, \dots, N_{n+1}\}$ represented on ∂A:
   (a) Extract the set of all T-junctions at level t, and let $2n_t$ denote its (necessarily even) cardinality.
   (b) Find a set of Euler spirals $(\gamma_j^t)_{j \in \{1,\dots,n_t\}}$ that pairwise connect compatible T-junctions with tangency constraints and minimize
   $$\sum_{j=1}^{n_t} \int_{\gamma_j^t} (\alpha + \beta |\kappa|^2) ds. \tag{4.11}$$
   The minimization is performed over all configurations only if n_t is low, due to exponential complexity (in practice $n_t \leq 6$ is a reasonable choice). Otherwise the minimization is performed over *causal* configurations only (see section 4.2 for a brief account of how it can be done with dynamic programming). In contrast with the algorithm of section 4.2, curves may cross even for causal configurations since we are no longer limited to straight lines.
   (c) Use the algorithm in Table 2 to build a set A_t from the collection of interpolating curves.
4. Define the inpainted function $\tilde{I}$ as
   $$\tilde{I}(x) = \begin{cases} I(x) & \text{if } x \notin A, \\ N_1 + \sum_{i=2}^{n+1} (N_i - N_{i-1}) \mathbb{1}_{A_{N_i}}(x) & \text{otherwise.} \end{cases} \tag{4.12}$$

in particular in Figure 10 how the technique naturally applies for reconstructing a geometric sketch obtained with the algorithm described in section 3.

It now remains, in order to design a geometry-guided inpainting method, to combine the sketch completion with an exemplar-based inpainting algorithm. We describe in the next section the particular instance of exemplar-based method that we will use.

5. A closer look at exemplar-based inpainting. In this section we detail a robust and efficient exemplar-based inpainting algorithm, the "patchwork" algorithm introduced in [72], that we will use later on. The inpainting is performed by copying entire patches taken in the valid part of the image, which allows a much better reconstruction of simple geometric structures in comparison with the methods that involve only one pixel at a time. In the appendix we provide a result that helps understanding, at least partially, how a generic patch-

Figure 8. *Top: original color image with an occluded zone. The transparency of the occlusion is for information only. Only the outer information is used for the reconstruction. Bottom: result after inpainting with straight lines* [65] *of each channel in a YUV representation.*

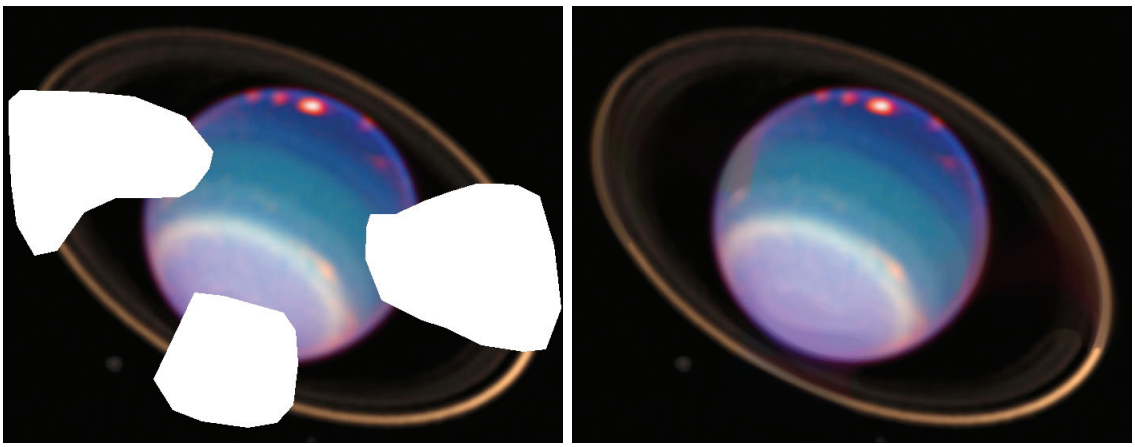

Figure 9. *Left: Uranus in infrared light (courtesy STScI and NASA) with artificial occlusions. Right: after reconstruction with Euler spirals of each channel in a YUV representation.*

based inpainting method handles the geometric information.

We consider a color RGB image $I = (I_1, I_2, I_3)$ defined on a discrete domain $\Omega \subset \mathbb{Z}^2$, and we denote as $A \subset \Omega$ the inpainting domain. The image norm at every pixel x is $\|I(x)\| = \sqrt{\sum_{i=1}^{3} I_i^2(x)}$. Given $p \in \mathbb{N}$, the *patch* $\Gamma_p(\mathbf{x})$ of a point $\mathbf{x} \in A$ is defined as the discrete $(2p+1)^2$-square neighborhood centered at $\mathbf{x}$. With $q > p \in \mathbb{N}$ we associate the ring $B_{p,q}(\mathbf{x}) = \Gamma_q(\mathbf{x}) \setminus \Gamma_p(\mathbf{x})$. If $G \subset \mathbb{Z}^2$, ∂G denotes its *inner boundary*, that is, the set of points that are in G and have at least one neighbor in its complementary set G^c. We also consider the set of all centers of neighborhoods that are included in A^c,

$$E_q = \{\mathbf{x} \in \Omega : \Gamma_q(\mathbf{x}) \subset A^c\}. \tag{5.1}$$

Eventually, we consider the set of patches intersecting A and lying on a regular grid with a mesh of size $(2p+1)^2$. That is, we define

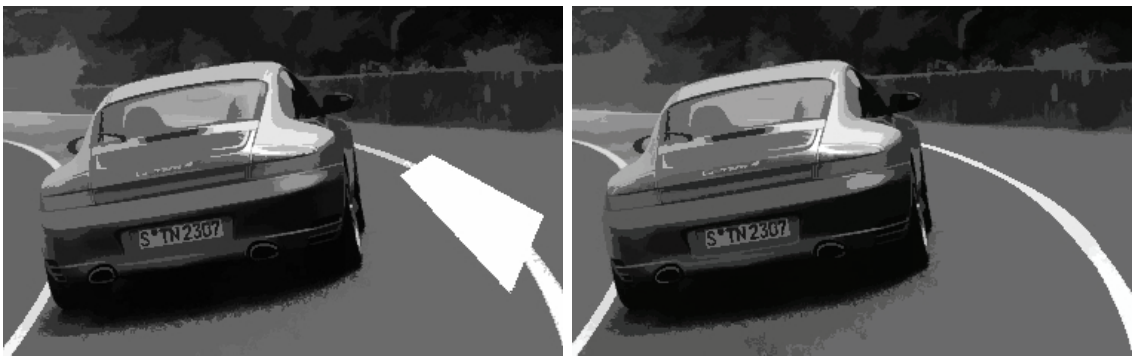

Figure 10. *A geometric sketch (see section 4.3) and its reconstruction using Euler spirals.*

$$\mathcal{F}_q = \{\Gamma_p(\mathbf{x}),\ \mathbf{x} = (k(2p+1)+p, l(2p+1)+p),\ k,l \in \mathbb{Z},\ \Gamma_p(\mathbf{x}) \cap A \neq \emptyset\}. \tag{5.2}$$

The basic idea of the algorithm is that the unknown parts of patches from $\mathcal{F}_q$ will be sequentially updated using a dictionary made up of patches centered at points of E_q.

Table 4
Exemplar-based inpainting algorithm [72] for a color RGB image (I_1, I_2, I_3).

EXEMPLAR-BASED INPAINTING ALGORITHM [72]

1. Let $\Gamma_p(\mathbf{x}_0)$ be a patch in $\mathcal{F}_q$ such that the intersection between $B_{p,q}(\mathbf{x}_0)$ and A^c has maximal area (a selection criterion such as the one proposed in [28] could also be used).
2. Let $\mathbf{y}_0$ be defined as

$$\mathbf{y}_0 = \operatorname*{argmin}_{\mathbf{y}\in E_q} \sum_{\substack{\mathbf{y}_i \in B_{p,q}(\mathbf{y}) \\ \mathbf{y}_i - \mathbf{y} + \mathbf{x}_0 \in A^c}} \left\| I(\mathbf{y}_i) - I(\mathbf{x}_0 - \mathbf{y} + \mathbf{y}_i) \right\|^2; \tag{5.3}$$

   i.e., $\mathbf{y}_0$ is the center of the patch in A^c whose associated ring is the most similar to the known part of the ring associated with the patch centered at $\mathbf{x}_0$.
3. For each $\mathbf{x} \in \Gamma_p(\mathbf{x}_0) \cap A$, let $I(\mathbf{x}) = I(\mathbf{y}_0 + \mathbf{x} - \mathbf{x}_0)$, and remove $\mathbf{x}$ from A.
4. Replace $\mathcal{F}_q$ with $\mathcal{F}_q \setminus \Gamma_p(\mathbf{x}_0)$, and go to 1 if $F_p \neq \emptyset$.

Given two integers $p < q$, the algorithm consists in replacing the unknown part of a patch $\Gamma_p(\mathbf{x})$ intersecting A with the values of the patch $\Gamma_p(\mathbf{y})$, where $\mathbf{y}$ is such that the distance between the rings $B_{p,q}(\mathbf{x})$ and $B_{p,q}(\mathbf{y})$ is minimal. In other words, the comparison and the copy involve different pixels and the fact that $B_{p,q}(\mathbf{x})$ encloses $\Gamma_p(x)$ avoids blocky effects. Indeed, the patches progressively inpainted by the algorithm do not overlap, but the associated rings do overlap. This is an important issue, for blocky effects are a noticeable drawback of many exemplar-based methods. The algorithm is summarized in Table 4.

This method gives impressive results on natural images, as may be seen in Figure 11. It essentially depends on two parameters: the size $(2p+1)^2$ of the patches to be copied and the size $(2q+1)^2$ of the neighborhood used to quantify the similarity of two patches. Usually, small values of p yield good results for weakly structured texture images. The case $p = 0$

corresponds to the original algorithm of Efros and Leung [33]. Larger values of p give better results in the presence of geometric structures. Indeed, large patches enable the propagation of straight edges, whereas small values of p may lead to the creation of erratically oscillating structures. However, even though edges are somehow preserved when using large patches, two problems remain for achieving a satisfying reproduction of geometry. First, exemplar-based methods are local or semilocal, and therefore long edges crossing the missing part of the image are often not satisfactorily restored (see the appendix for partial theoretical results on the connections between patch-based inpainting and geometry reconstruction, and [4] for a more detailed study from a related perspective). Second, curved edges are not well prolongated if a suitable curvy piece of edge cannot be found elsewhere in the image. A solution would be to enlarge the dictionary of patches by simply taking rotated versions. This sometimes yields nice results, but, in general, erratic results often occur when the dictionary becomes too rich.

Figure 11. *Top: original image with missing parts (holes) in white. Bottom: reconstructed image using the exemplar-based method from* [72] *(with $p = 4$ and $q = 6$).*

6. Using the reconstructed sketch to guide the exemplar-based algorithm: A Lagrangian formulation. Recall that the main motivation for this paper is to adapt an exemplar-based algorithm in order to improve the quality of the restored geometry, and in particular to properly handle long-range geometric features, while maintaining a good ability to reproduce the texture. We saw in section 3 how to compute a sketch of the image outside the inpainting domain and in section 4 how to interpolate it, using either straight lines or Euler spirals. Let us now examine how the reconstructed sketch can be used for guiding the exemplar-based algorithm of the previous section. Again, we denote by A the inpainting domain and by $I = (I_1, I_2, I_3)$ the RGB image known only on $\Omega \setminus A$. We write I_S^g for the interpolated sketch, a gray level image known on the *whole* image domain Ω. More precisely, I_S^g denotes the interpolation of the gray level sketch obtained from the luminance channel in a YUV representation of the original image I. As already mentioned (see also [20]), the luminance channel essentially contains all the geometric information of almost every real world image; therefore using color sketches is unnecessary or even potentially harmful because of false colors.

Based on the interpolated sketch I_S^g, the core algorithm of our approach is given in Table 5. Roughly speaking, given a point $\mathbf{x}$ on ∂A and using the notations of section 5, the algorithm searches for $\mathbf{y} \in A^c$ such that, simultaneously, the following hold:

- the patches $\Gamma_q(\mathbf{x})$ and $\Gamma_q(\mathbf{y})$ in the luminance channel of the (previously reconstructed) sketch are similar;
- the (known parts of the) rings $B_{p,q}(\mathbf{x})$ and $B_{p,q}(\mathbf{y})$ in the current image $I = (I_1, I_2, I_3)$ are similar.

Let us comment on the use of the term "Lagrangian": we do not impose as a hard constraint that two related points $\mathbf{x}_0$ and $\mathbf{y}_0$ must have the same neighborhood in the sketch image. Instead, we rather use it as a soft constraint by introducing an additional term, of Lagrangian type, that penalizes the discrepancy between the neighborhoods.

Table 5

The Lagrangian-type geometry/texture inpainting algorithm proposed in this paper. I_S^g denotes the sketch that has been previously reconstructed on the whole image domain, and (I_1, I_2, I_3) is the RGB image that is progressively inpainted.

GEOMETRICALLY GUIDED EXEMPLAR-BASED INPAINTING ALGORITHM

1. Let $\Gamma_p(\mathbf{x}_0)$ be a patch in $\mathcal{F}_q$ such that the intersection between $B_{p,q}(\mathbf{x}_0)$ and A^c has maximal area (a selection criterion such as the one proposed in [28] could also be used).
2. Let $\mathbf{y}_0$ be defined as

$$\mathbf{y}_0 = \underset{\mathbf{y}\in E_q}{\operatorname{argmin}} \Big\{ \lambda \Big(\sum_{\mathbf{y}_i \in \Gamma_q(\mathbf{y})} \left| I_S^g(\mathbf{y}_i) - I_S^g(\mathbf{x}_0 + \mathbf{y}_i - \mathbf{y}) \right|^2 \Big) + (1-\lambda) \Big(\sum_{\substack{\mathbf{y}_i \in B_{p,q}(\mathbf{y}) \\ \mathbf{x}_0 + \mathbf{y}_i - \mathbf{y} \in A^c}} \left\| I(\mathbf{y}_i) - I(\mathbf{x}_0 + \mathbf{y}_i - \mathbf{y}) \right\|^2 \Big) \Big\}. \tag{6.1}$$

3. For each $\mathbf{x} \in \Gamma_p(\mathbf{x}_0) \cap A$, let $I(\mathbf{x}) = I(\mathbf{y}_0 + \mathbf{x} - \mathbf{x}_0)$, and remove $\mathbf{x}$ from A.
4. Replace $\mathcal{F}_q$ with $\mathcal{F}_q \setminus \Gamma_p(\mathbf{x}_0)$, and go to 1 if $A \neq \emptyset$.

The parameter λ weights the respective contributions of the sketch and the original image. Taking $\lambda = 0.5$ was a reasonable choice for all the experiments presented hereafter, which explains why we will not discuss the correct tuning of this parameter. We nevertheless believe that understanding correctly how λ must be chosen according to the very nature of the image is an interesting challenge.

7. Experimental results.

Parameter setting. The first parameter involved in the complete inpainting procedure is ϵ, driving the segmentation step. As is common for *a contrario* methods, it has been set to $\epsilon = 1$ for all experiments. As said before, we have no rigorous way to set the parameter λ, balancing the geometry and texture contributions. The value $\lambda = 0.5$ appears reasonable and has been kept for all experiments. As for the "patchwork" step, parameters have also been fixed to $p = 4$ and $q = 6$, respectively.

Experiments. We display in Figures 12–21 several examples of image reconstruction where we systematically compare the approach proposed in this paper with the original "patchwork" algorithm from [72]. In a few experiments, we also compare with the methods proposed in [90] and [57], as well as with the recent Photoshop's "content aware fill" that is a nice extension of [90] combined with [7]. We have systematically indicated the size of the inpainting domain,

or more precisely the size of a square with the same number of pixels. As can be observed, the experiments are made for fairly large inpainting domains, which proves the nonlocal nature of the geometric reconstruction. In Figures 12 and 13, we display examples where the use of a geometrical guide (obtained using straight lines) improves the reconstruction of the geometry. In Figure 13(e)–(g), we also display a case of failure in the presence of a T-junction. Indeed, T-junctions as well as corners are not accounted for in both sketch reconstruction algorithms presented in section 4.

Figure 14 illustrates that, in some situations, the geometric guiding may yield a result not better than with the plain "patchwork" algorithm. Note that, for this rectilinear example, it is preferable to use straight lines rather than Euler spirals for the geometric inpainting. In Figure 15, we display an example where the geometrically guided method actually yields worse results than the plain patchwork method. In this example of a dense forest, branches produce a kind of macrotexture that is kept in the sketch but whose structure is way too complex to be well interpolated by the geometric step of the inpainting. As a result, the interpolated sketch is wrong and, instead of helping the reconstruction, deteriorates the result obtained with only the "patchwork" algorithm.

In Figure 16, we present a toy example made of a textured disk, which illustrates the possibility of correctly reconstructing such a curvy geometry while reproducing well the texture part of the image.

In Figures 17 and 18, we show two examples where the use of a geometrical guide that has been interpolated by means of Euler spirals enables correct inpainting results in difficult cases. In these relatively high resolution examples, the geometry is restored over large distances. In both experiments, we also provide the results obtained with Photoshop's "content aware fill." Clearly, for such high-resolution images with a large inpainting domain and a clear long-range geometric information, our method performs better.

We have reported in Figure 18 the computational time required by our method on a PC Intel dual core 2 2.66GHz: less than 6 seconds are needed to get the result.

There are other situations where our method does not perform well, or at least worse than some state-of-the-art inpainting algorithms. This is illustrated in Figure 19, where we compare the "patchwork" algorithm (second column), the algorithm from [90] implemented by Yokoya's group (third column), the method proposed by Kawai, Sato, and Yokoya in [57] (fourth column), and our method (last column). A few comments on these experiments can be found in the figure's caption. Clearly, in these low resolution images, our method is not competitive if the geometry cannot be correctly computed. Multiscale strategies like in [90, 57] are much more efficient in this case. However, as illustrated by the performances of Photoshop's "content aware fill" in Figures 17 and 18, pure multiscale strategies might be not sufficient for high resolution images.

For the experiment shown in Figure 20(e), a larger dictionary of patches has been used in formula (6.1) by simply considering several rotations of each patch. This larger dictionary permits one to closely follow the reconstructed curves of the sketch even though none of the patches from the original image has the correct orientation. It is interesting to note that using geometrically guided rotated patches yields significantly better results than using rotated patches without guide since, as already mentioned, a richer dictionary of patches without additional constraints often leads to erratic results. Last, we display in Figure 21 an

example of a successful completion of an electron microscope image illustrating the ability of our method to restore properly both the texture and the geometry.

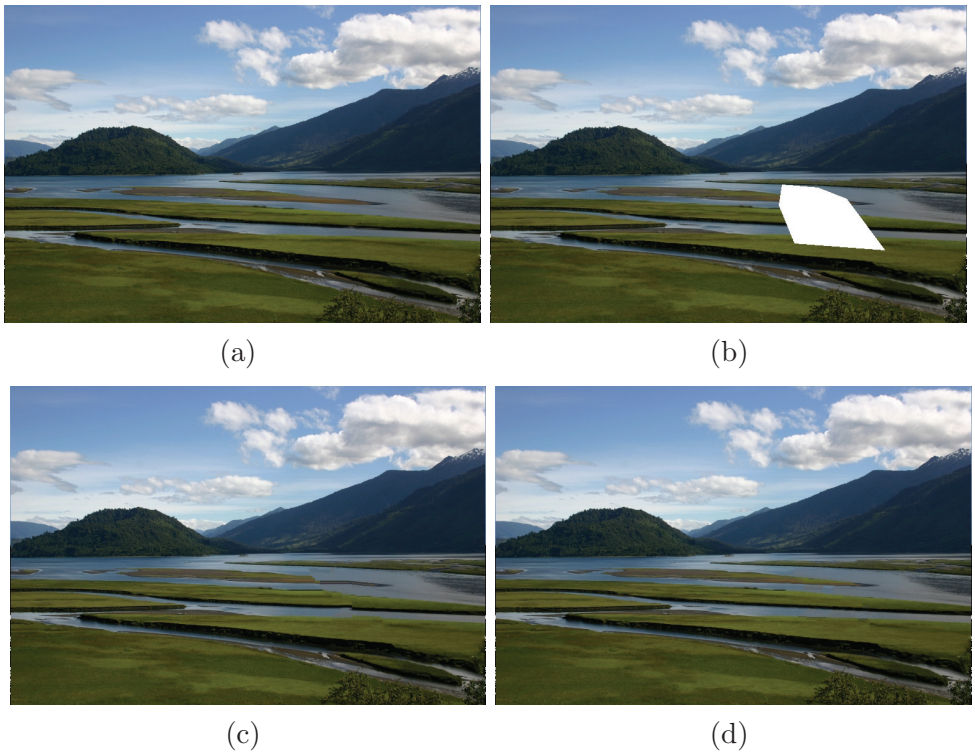

Figure 12. (a) *original* 512×768 *image;* (b) *occluded image (occlusion size* $\sim 100 \times 100$*);* (c) *image restored with the exemplar-based method from* [72]*;* (d) *image restored using the method introduced in this paper, detailed in section* 6 *(using* $p = 4$*,* $q = 6$*,* $\lambda = 0.5$*). The geometrical interpolation of the sketch (to obtain* (d)*) uses straight lines. A careful examination of the result reveals a few block artifacts that are inherent to the "patchwork" strategy if no postprocessing is applied in order to remove intensity jumps between blocks. This could be done using, for instance, the efficient blending method from* [71].

8. Conclusion.

We have presented a new method for the joint restoration of texture and geometry. In many situations, and in particular when the missing domain is large, our method has the ability to properly restore long-range geometric features like edges and, simultaneously, to sample and paste the texture information correctly. We showed that, in a few high resolution cases, it clearly outperforms state-of-the-art algorithms. For low resolution images together with complex geometries, multiscale strategies may be indicated more.

In brief, our approach has the following characteristics:

- It uses a simple and stable sketch representation of the image.
- Broken level lines are completed with Euler spirals that are natural curves for shape completion. In addition, sharp and possibly crossing discontinuities can be correctly handled.
- The Lagrangian-type variational formulation is flexible enough for the geometric guiding to combine softly with the texture reconstruction.

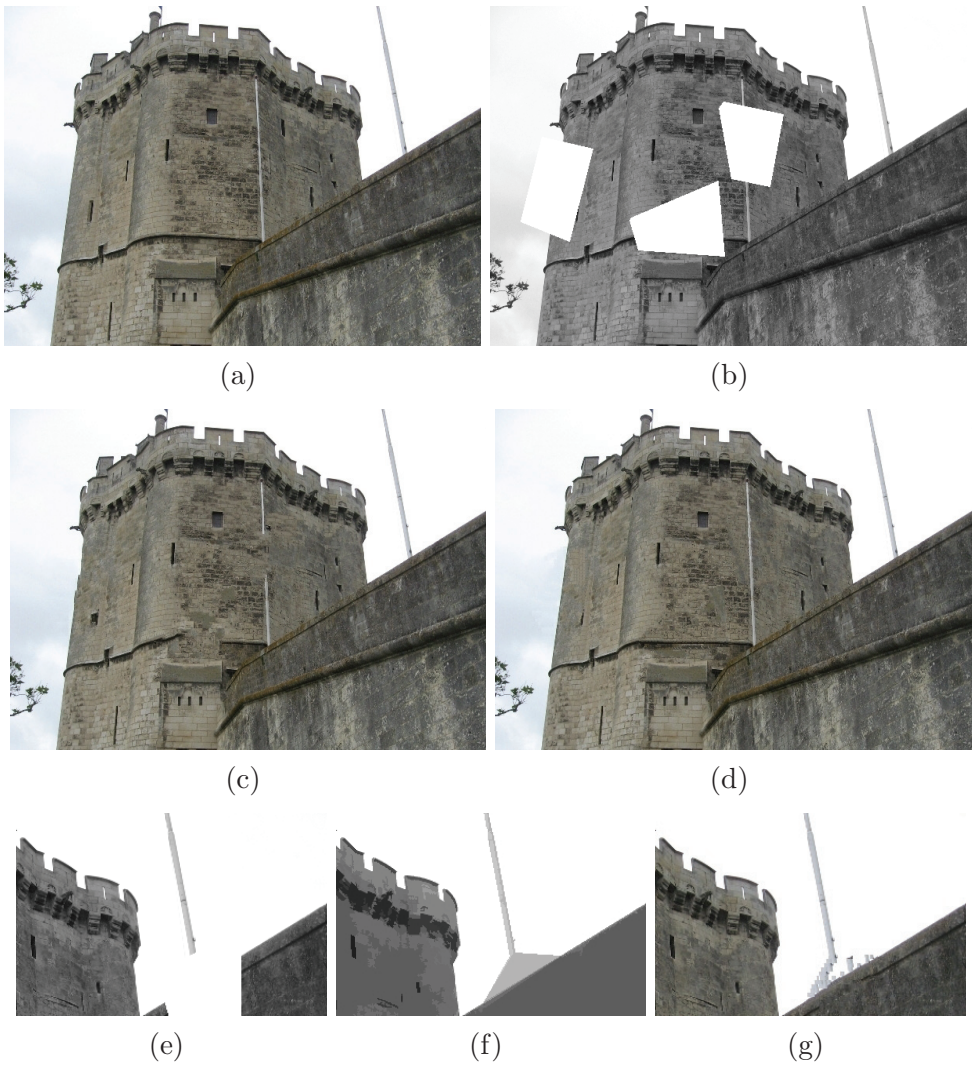

Figure 13. (a) *original* 434×609 *image;* (b) *occluded image (average occlusion size* $\sim 90 \times 90$*);* (c) *image restored using the exemplar-based method from* [72]*;* (d) *image restored using the method introduced in this paper, detailed in section* 6 *(using* $p = 4$*,* $q = 6$*,* $\lambda = 0.5$*, and straight lines for sketch interpolation). Last line: example of failure;* (e) *occluded detail from image* (a)*;* (f) *reconstructed sketch (using straight lines);* (g) *image restored using the geometrical guide. The method is not able to reconstruct the T-junction and therefore yields artifacts.*

- Very few parameters are used in comparison with other inpainting models involving a geometric guiding.

What are the limitations of our approach? Although it allows us to recover long-range edges, its performances are somewhat limited when the sketch to be reconstructed is intricate and contains many edges that can hardly be interpolated without ambiguity. In such a situation, it appears that a manual intervention is necessary. A natural approach could be to adapt to the level line framework the contour editing strategy proposed by Elder and Goldberg

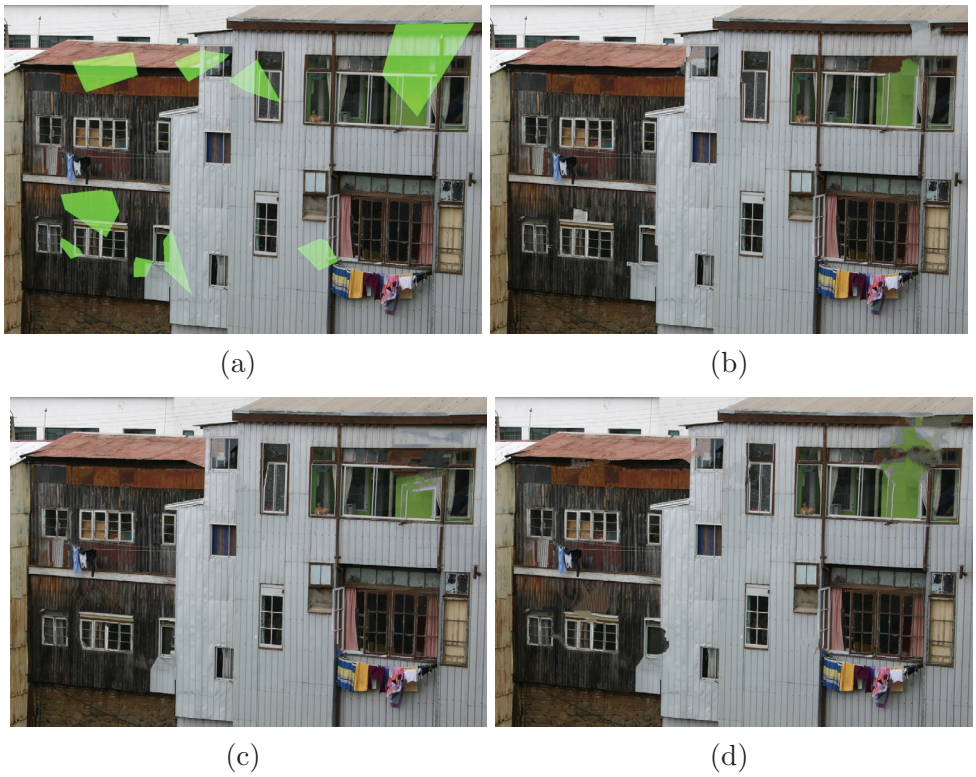

Figure 14. *A case where the geometrical guiding does not systematically improve the result given by the "patchwork" algorithm* [72]. (a) *occluded* 1160×800 *image (largest occlusion size* $\sim 170 \times 170$*, smallest occlusion size* $\sim 40 \times 40$*);* (b) *image restored using the exemplar-based method from* [72]*;* (c) *image restored using the method introduced in this paper, detailed in section* 6 *(using* $p = 4$*,* $q = 6$*,* $\lambda = 0.5$*, and straight lines for sketch interpolation);* (d) *same as* (c) *but using Euler spirals for the completion of the sketch.*

Figure 15. *A case where the geometrical guiding yields worse results than the "patchwork" algorithm* [72]*. In this case, the macrotexture structure is too complex to be correctly interpolated.* (a) *occluded* 676×1014 *image (occlusion size* $\sim 100 \times 300$*);* (b) *interpolated sketch using straight lines;* (c) *image restored using the exemplar-based method from* [72]*;* (d) *image restored using the method introduced in this paper, detailed in section* 6 *(using* $p = 4$*,* $q = 6$*,* $\lambda = 0.5$*).*

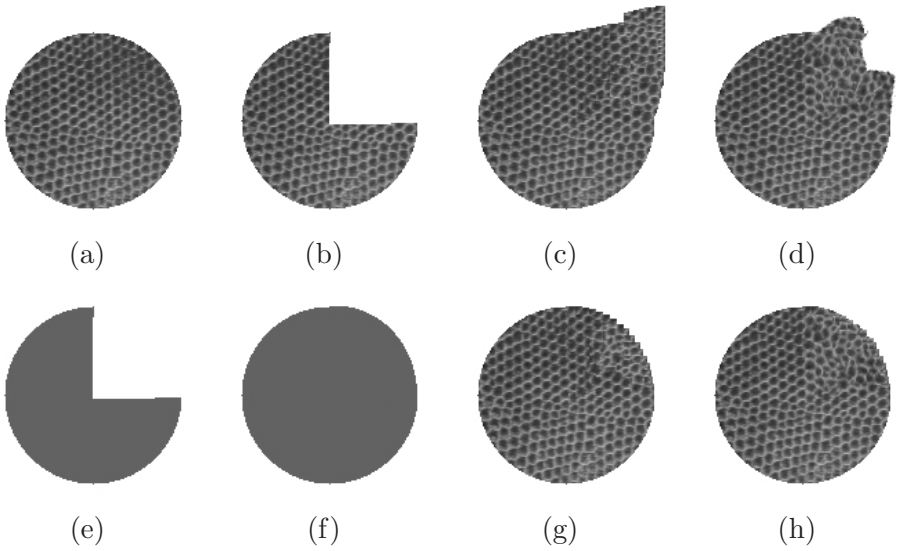

Figure 16. (a) *original image (disk diameter* ~ 260*);* (b) *occluded image;* (c) *image restored using the "patchwork" algorithm* [72]*;* (d) *same restoration process but using for each patch in the dictionary* 15 *additional patches obtained by rotation. The result is better but still unsatisfactory;* (e) *sketch of the image;* (f) *sketch reconstructed using Euler spirals;* (g) *image restored using the method detailed in section* 6 *(with* $p = 4$*,* $q = 6$*,* $\lambda = 0.5$ *and* (f) *as guiding image);* (h) *same method as in* (g) *but with the dictionary enriched by* 15 *rotations of each patch so that the disk boundary is smoother.*

in [37]. The user could simply mark the pairs of lines to be connected, and the program would compute the optimal sets of Euler spirals. This would give a more flexible—but no more automatic—algorithm for inpainting.

We conclude by mentioning that several variants of our model can be proposed: a natural one consists in pasting samples in priority along the interpolated curves of the sketch, in the spirit of [80], except that in our case the curves are automatically computed and not drawn by the user. A second variant consists in changing the energy in (6.1), for instance replacing the L^2 norm by another L^p norm (see [4] for a discussion based on axiomatic requirements). One can also replace the linear combination of the image and sketch energy terms in (6.1) by a new energy and define the center of a candidate patch as a solution to

$$\mathbf{y}_0 = \operatorname*{argmin}_{\mathbf{y}\in E_q}\Big\{\max\Big\{\frac{1}{Z_1}\Big(\sum_{\mathbf{y}_i\in\Gamma_q(\mathbf{y})}\Big|I_S^g(\mathbf{y}_i) - I_S^g(\mathbf{x}_0+\mathbf{y}_i-\mathbf{y})\Big|^p\Big), \frac{1}{Z_2}\Big(\sum_{\substack{\mathbf{y}_i\in B_{p,q}(\mathbf{y})\\ \mathbf{x}_0+\mathbf{y}_i-\mathbf{y}\in A^c}}\Big\|I(\mathbf{y}_i) - I(\mathbf{x}_0+\mathbf{y}_i-\mathbf{y})\Big\|^p\Big)\Big\}\Big\}, \tag{8.1}$$

where Z_1 and Z_2 are suitable normalization constants depending on the dynamic ranges of I_S^g and I, respectively, and on the respective sizes of $\Gamma_p(\cdot)$ and $B_{p,q}(\cdot)$. The choice of Z_1 and Z_2 is, however, rather delicate and strongly depends on the type of images, but there are obviously situations where this energy would be more adapted.

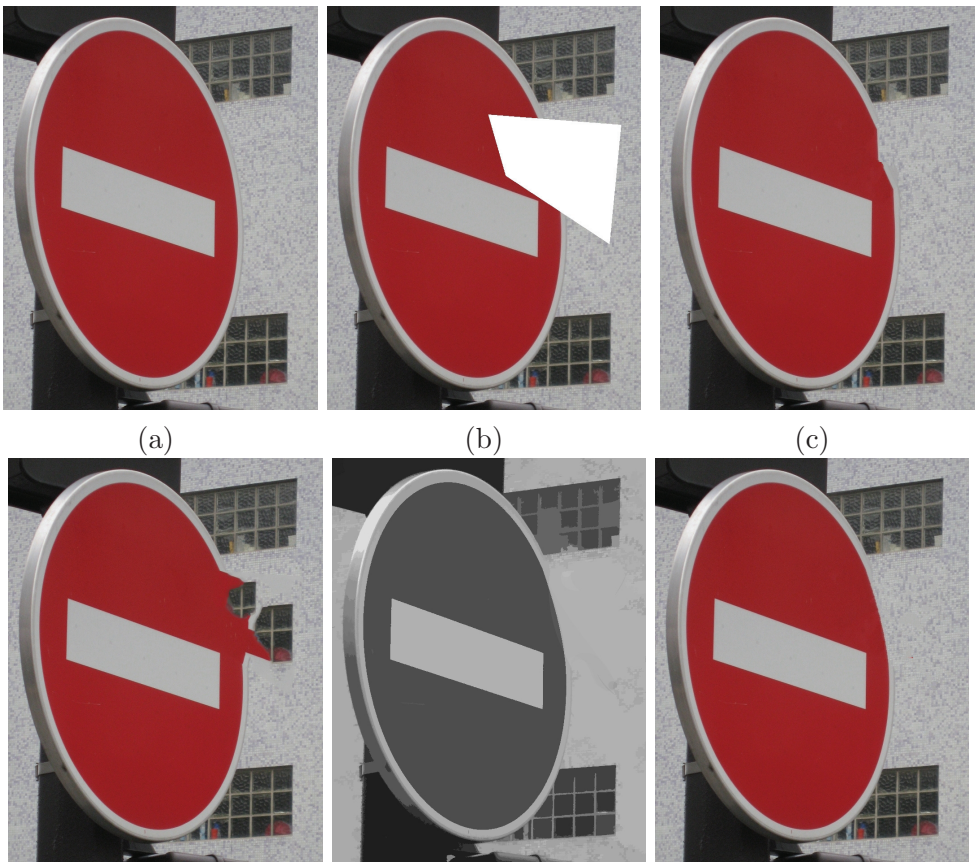

Figure 17. (a) *original* 900×700 *image;* (b) *occluded image (occlusion size* $\sim 230 \times 230$*);* (c) *image restored using the "patchwork" algorithm* [72]*;* (d) *image restored using Photoshop's "content aware fill";* (e) *after inpainting with Euler spirals of the sketch image computed from the luminance channel in a YUV representation of* (b)*;* (f) *image restored using the method detailed in section* 6 *(using* (e) *and* $p = 4$*,* $q = 6$*,* $\lambda = 0.5$*).*

Appendix. Patches metric and curvature. We briefly investigate a connection between the exemplar-based approach described in section 6 and the restoration of geometry. It is obvious from the experiments that exemplar-based inpainting methods restore not only texture but also the *local* geometric information. This is not surprising since copying a patch means copying both the texture and the geometry. However, we may wonder whether a theoretical link can be found between the replication of patches and the geometry. This is the purpose of the next proposition (see also [4] for a more general variational formulation), where we investigate a very simplified model of replication. More precisely, we prove that the difference between a smooth set and the concatenation of "straight" patches (i.e., intersections between balls and half-planes) can be evaluated using a criterion that involves the curvature. This is not surprising since the curvature measures the variation of the tangents to the set boundary. In addition, this result quantifies the fact (which is obvious from an experimental point of view) that the more oscillating a boundary is, the more difficult it is to recover it using only

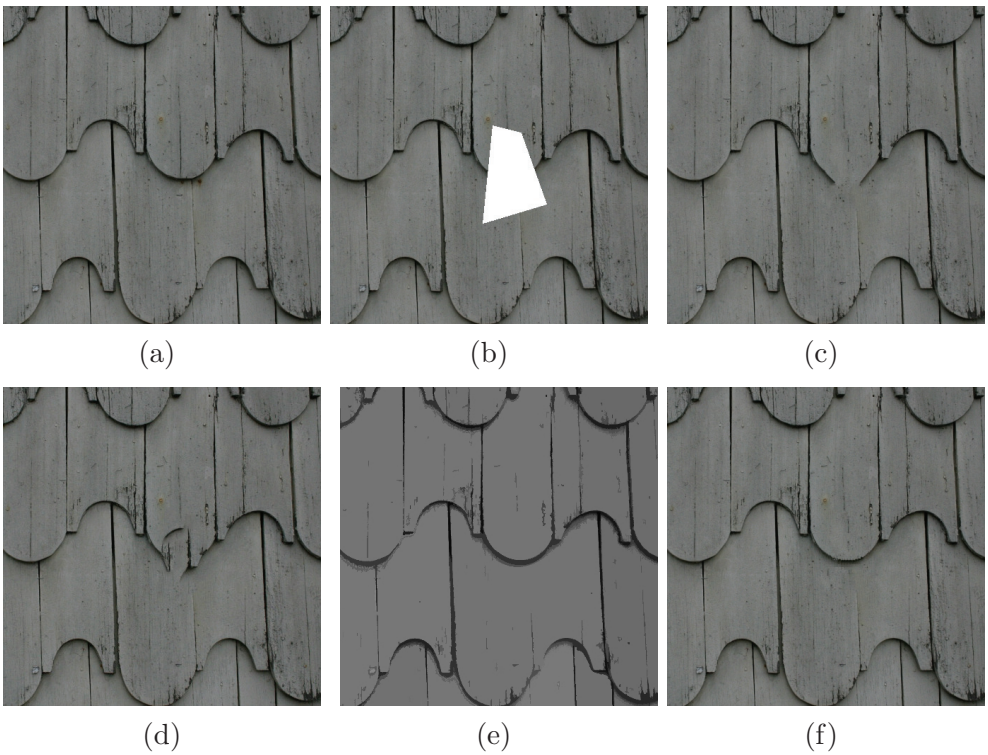

Figure 18. *Various experiments with the image of Figure 2. For the purpose of visualization, the results* after *processing have been cropped:* (a) 450×450 *cropped image;* (b) *occluded image (occlusion size* ∼ 90×90*);* (c) *image restored using the "patchwork" method from* [72]*;* (d) *image restored with Photoshop's "content aware fill";* (e) *inpainting with Euler spirals of the sketch image associated with the luminance channel of a YUV representation of* (b)*;* (f) *image restored using the method detailed in section 6 (using* (e) *and* $p = 4$, $q = 6$, $\lambda = 0.5$*). It takes less than* 6 *seconds on a PC with Intel dual core* 22.66*GHz to get this result. More precisely, the computational times for each step of the algorithm are the following: sketch computation* (3"45)*, geometric inpainting* (0"20)*, "patchwork" method applied to a region of extra-width* 16 *pixels around the inpainting region* (1"69).

"straight" patches.

Let E be an open, bounded subset of $\mathbb{R}^2$ with C^3 boundary. Given $x \in \mathbb{R}^2$, we call a *straight patch* of size r centered at x any set obtained as the intersection between the disk $D_r(x)$ of radius r centered at x and a half-plane. We let $\mathcal{P}$ denote the collection of all half-planes in $\mathbb{R}^2$. We want to quantify asymptotically (as r tends to 0) how much E differs at every point from a straight patch. A natural dissimilarity criterion is defined as

$$J_r(E) = \int_{\mathbb{R}^2} \left(\inf_{P \in \mathcal{P}} \left| (E \Delta P) \cap D_r(x) \right|^p \right) dx, \tag{A.1}$$

where Δ is the symmetric difference operator, $|\cdot|$ the Lebesgue measure, and $p > 0$ a parameter.

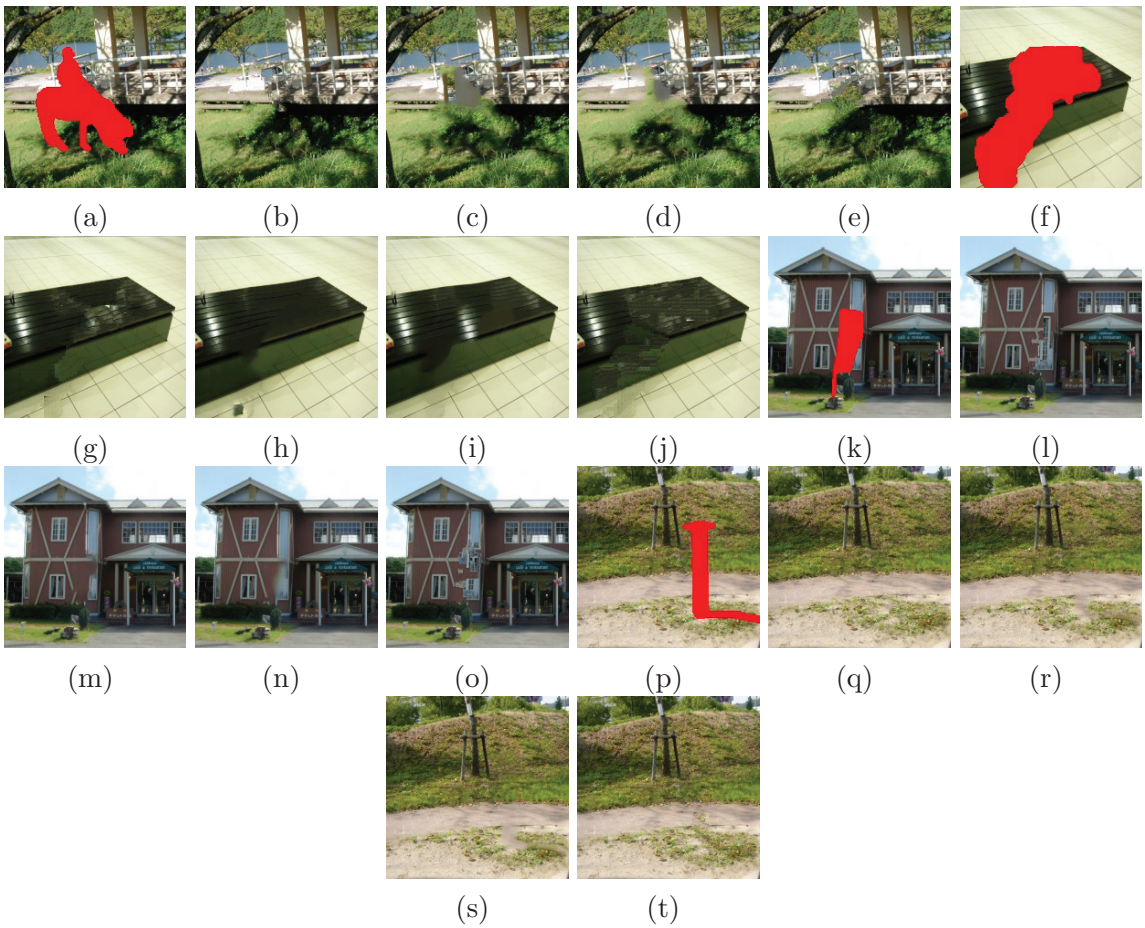

Figure 19. *An illustration that, for these low-resolution images taken from the database* http://yokoya.naist.jp/research/inpainting, *our method (last column) does not perform better—and may sometimes perform notably worse—than state-of-the-art algorithms as in* [90] *(third column) or* [57] *(fourth column). Results in the second column are obtained with the "patchwork" algorithm. The results in both the third and the fourth columns are provided by the authors of the database. A possible explanation for the failures of our method in the second and third lines is that, for such low-resolution images, a good multiscale approach together with a suitable choice of the block size actually works better than a geometric guiding, especially if the geometry is complex (third line) or cannot be easily recovered from the level lines due to strong changes of illumination (second line, where our result could in fact be improved by iterating a Poisson filter).*

Proposition A.1. *There exists $C_p > 0$ such that*

$$J_r(E) = C_p r^{3p+1} \int_{\partial E} |\kappa_{\partial E}|^p \, d\mathcal{H}^1 + o(r^{3p+1}). \tag{A.2}$$

Proof.

Step 1. Up to a translation, we may assume that $x = 0$ and suppose that, in a neighborhood of 0, E coincides, up to a rotation, with the set of points (x, y) such that $y \geq \lambda + \frac{\kappa_{\partial E}(0)}{2} x^2$ for some $\lambda \in \mathbb{R}$. If $\lambda < -r$ or $\lambda > r$ or if $\kappa_{\partial E}(0) = 0$, then $J_r(E, 0) = 0$. By symmetry, we can

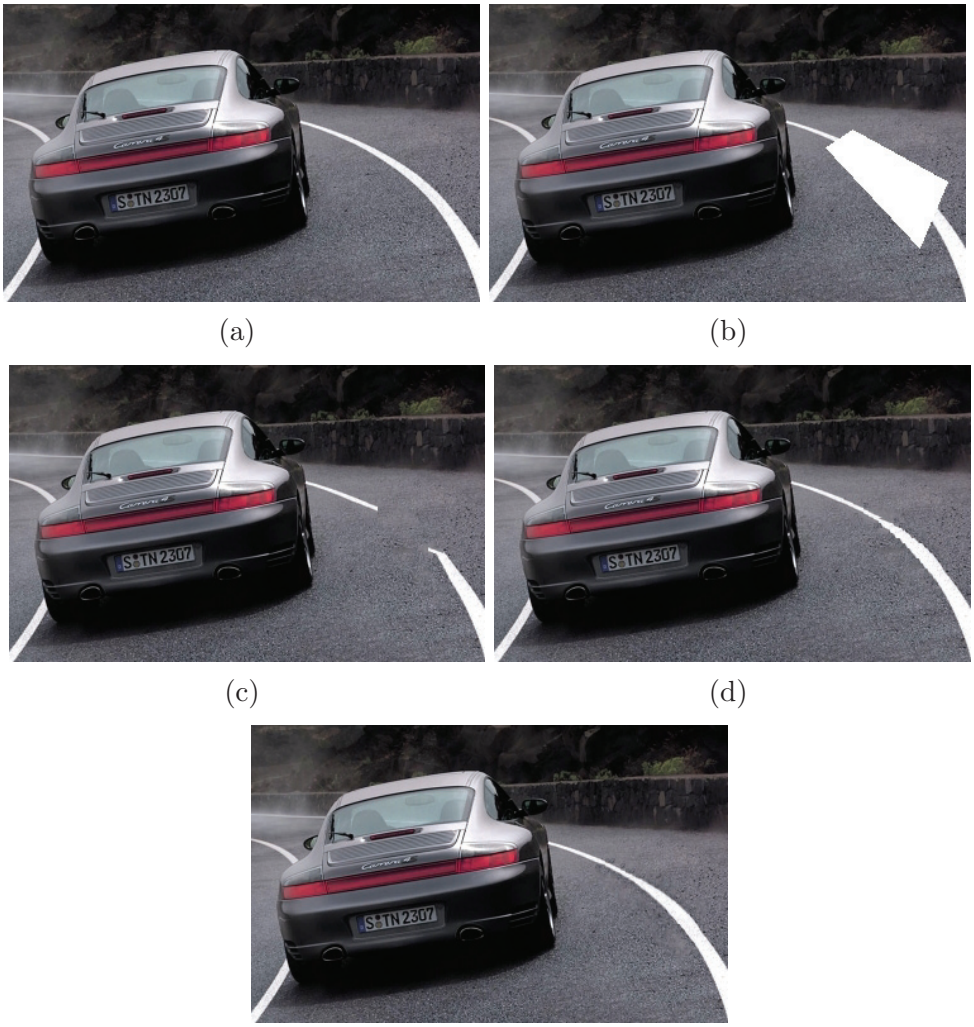

Figure 20. (a) *original* 235×377 *image;* (b) *occluded image (occlusion size* $\sim 40 \times 100$*);* (c) *image restored using the "patchwork" algorithm* [72]*;* (d) *image restored using our geometry-guided inpainting method based on Euler spirals (with* $p = 4$*,* $q = 6$*, and* $\lambda = 0.5$*);* (e) *same method as in* (d)*, but this time the dictionary used to performed the exemplar-based step contains patches from the original image as well as rotations of these patches.*

now assume that $\kappa := \kappa_{\partial E}(0) > 0$ and observe that

$$J_r(E,0) = \inf_{(m,p)\in\mathbb{R}^2} \left| \left(\{y \geq m + px\} \Delta \left\{ y \geq \lambda + \frac{\kappa}{2} x^2 \right\} \right) \cap D_r(0) \right|. \tag{A.3}$$

By symmetry arguments, we observe that for every $m \in \mathbb{R}$, $p \mapsto |(\{y \geq m + px\}\Delta\{y \geq \lambda + \frac{\kappa}{2}x^2\}) \cap D_r(0)|$ is minimal whenever $p = 0$. Therefore, the computation of $J_r(E,0)$ reduces to minimizing $m \mapsto \int_{D_r(0)} |\lambda - m + \frac{\kappa}{2}x^2|\, dx$; see Figure 22.

Let A be the intersection point between the right branch of the parabola and the circle $C_r(0)$, and let B be the intersection point with positive x-value between the line $y = m$ and

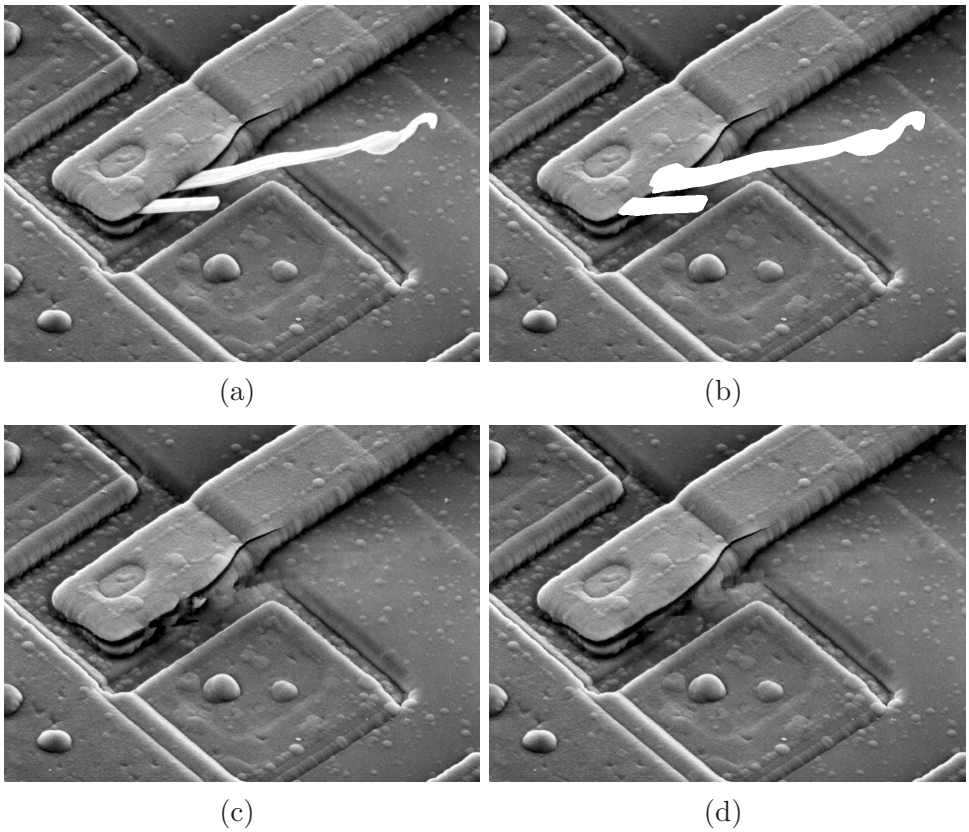

Figure 21. (a) *A low-resolution scanning electron microscope* 748×1000 *image showing an integrated circuit which was given an overload current and subsequently failed thermally. The fibers are oxides which formed and grew out of the device during thermal destruction (From: Digital Image Processing, 3rd ed. by R. C. Gonzalez and R. E. Woods, Prentice Hall,* 2008. *Used with permission).* (b) *a mask covering the fibers (global size* $\sim 40 \times 775$*);* (c) *image restored with the exemplar-based method from* [72] *(using* $p = 6$*,* $q = 8$*,* $\lambda = 0.5$*);* (d) *image restored using the method detailed in section* 6 *(using* $p = 6$*,* $q = 8$*,* $\lambda = 0.5$*, and Euler spirals for sketch interpolation).*

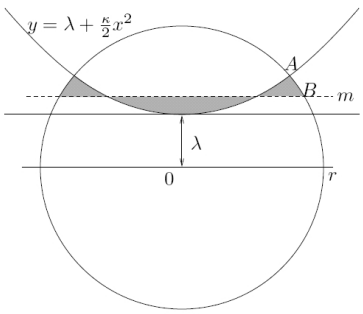

Figure 22. *Approximation in* $D_r(0)$*.* $J_r(E, 0)$ *is the gray area.*

the circle. Clearly, if m minimizes $\int_{D_r(0)} |\lambda - m + \frac{\kappa}{2}x^2|\,dx$, then $\lambda \le m \le y_A$. Let P be the patch $D_r(0) \cap \{y \ge m\}$. Then

$$\int_{-x_A}^{x_A} \left|\lambda - m + \frac{\kappa}{2}x^2\right| dx \le |E \Delta P| \le \int_{-x_B}^{x_B} \left|\lambda - m + \frac{\kappa}{2}x^2\right| dx, \tag{A.4}$$

and therefore

$$\inf_{m\in\mathbb{R}} \int_{-x_A}^{x_A} \left|\lambda - m + \frac{\kappa}{2}x^2\right| dx \le J_r(E,0) \le \inf_{m\in\mathbb{R}} \int_{-x_B}^{x_B} \left|\lambda - m + \frac{\kappa}{2}x^2\right| dx. \tag{A.5}$$

Now the minimum of $\int_{-x_A}^{x_A} |\lambda - m + \frac{\kappa}{2}x^2|\,dx$ is reached at $m_A =$ the median value of $\lambda + \frac{\kappa}{2}x^2$ on $[-x_A, x_A]$, i.e., $m_A = \lambda + \frac{\kappa}{2}(\frac{x_A^2}{4})$. Analogously, a minimizer of $\int_{-x_B}^{x_B} |\lambda - m + \frac{\kappa}{2}x^2|\,dx$ is $m_B = \lambda + \frac{\kappa}{2}(\frac{x_B^2}{4})$. Plugging into (A.5) yields

$$\frac{\kappa}{4}x_A^3 \le J_r(E,0) \le \frac{\kappa}{4}x_B^3. \tag{A.6}$$

Note that $x_B^2 = r^2 - \lambda^2$ and $x_A^2 = r^2 - \left(\lambda + \frac{\kappa}{2}x_A^2\right)^2$; thus $x_A^2 \ge r^2 - \lambda^2 - \kappa r^3$, which is positive for r small enough, and therefore

$$\frac{\kappa}{4}(r^2 - \lambda^2 - \kappa r^3)^{\frac{3}{2}} \le J_r(E,0) \le \frac{\kappa}{4}(r^2 - \lambda^2)^{\frac{3}{2}}. \tag{A.7}$$

Step 2. Assume now that E has C^3 boundary, and let $x \in \partial E \oplus D_r$. For r small enough and after a suitable change of coordinate frames, ∂E on $D_r(x)$ can be approximated as a parabola of the form $\lambda + \frac{\kappa}{2}x^2$ in $D_r(0)$, where κ is the curvature of ∂E at $(0,\lambda) \in \partial E$ in the new coordinate frame. The approximation error is an $o(r^2)$; therefore

$$\frac{\kappa}{4}(r^2 - \lambda^2 - \kappa r^3)^{\frac{3}{2}} + o(r^3) \le J_r(E,0) \le \frac{\kappa}{4}(r^2 - \lambda^2)^{\frac{3}{2}} + o(r^3). \tag{A.8}$$

Step 3. Let d_E denote the signed distance function to ∂E (positive outside E, negative inside). Since d_E is smooth in a band containing ∂E and $|\nabla d_E| = 1$ a.e., we deduce from the coarea formula that

$$J_R(E) = \int_{-r}^{r} \int_{\partial\{x:\, d_E(x) \ge \lambda\}} \Big(J_r(E,y)\Big)^p \, d\mathcal{H}^1(y)\, d\lambda. \tag{A.9}$$

By a well-known property of the distance function, d_E locally inherits the regularity of ∂E. If r is small enough, for any $x \in \partial E \oplus D_r$, there exists a unique $y \in \partial E$ such that $x = y + d_E(x) n_{\partial E}$, where $n_{\partial E}$ denotes the outer normal to E at y. Considering the arc-length s on ∂E, there exists a parameterization s' on $\partial\{x : d_E(x) \ge \lambda\}$ such that $ds' = (1 - \kappa\lambda)\,ds$. Therefore, by Fubini's theorem,

$$\begin{aligned} J_r(E) &= \int_{-r}^{r} \int_{\partial E} \Big(J_r(E, x + \lambda n_{\partial E})\Big)^p (1 - \lambda\kappa(x))\, d\mathcal{H}^1\, d\lambda \\ &= \int_{\partial E} \int_{-r}^{r} \Big(J_r(E, x + \lambda n_{\partial E})\Big)^p (1 - \lambda\kappa(x))\, d\lambda\, d\mathcal{H}^1. \end{aligned}$$

By (A.8) and the change of variables $\lambda \to \lambda/r$,

$$\text{(A.10)}\quad r^{3p+1}\int_{\partial E}\left|\frac{\kappa}{4}\right|^p\int_{-1}^{1}\Big((1-\lambda^2-\kappa(x)r)^{3/2}+\mathrm{o}(1)\Big)^p(1-\lambda\, r\,\kappa(x))\,d\lambda\,d\mathcal{H}^1$$
$$\leq J_r(E)\leq r^{3p+1}\int_{\partial E}\left|\frac{\kappa}{4}\right|^p\int_{-1}^{1}\Big((1-\lambda^2)^{3/2}+\mathrm{o}(1)\Big)^p(1-\lambda\, r\,\kappa(x))\,d\lambda\,d\mathcal{H}^1.$$

With the curvature along ∂E being uniformly bounded, the proposition ensues by taking $C=\frac{1}{4^p}\int_{-1}^{1}(1-\lambda^2)^{3p/2}\,d\lambda$. ∎

Acknowledgment. We are grateful to G. Scoutheeten, who implemented a preliminary version of the code.

REFERENCES

[1] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro, *A variational framework for exemplar-based image inpainting*, Int. J. Comput. Vis., 93 (2011), pp. 319–347.
[2] M. Ashikhmin, *Synthesizing natural textures*, in Proceedings of the ACM'01 Symposium on Interactive 3D Graphics, 2001, pp. 217–226.
[3] L. Atzori and F. D. Natale, *Error concealment in video transmission over packet networks by a sketch-based approach*, Signal Process. Image Comm., 15 (1999), pp. 57–76.
[4] J.-F. Aujol, S. Ladjal, and S. Masnou, *Exemplar-based inpainting from a variational point of view*, SIAM J. Math. Anal., 42 (2010), pp. 1246–1285.
[5] D. Auroux and M. Masmoudi, *A one-shot inpainting algorithm based on the topological asymptotic analysis*, Comput. Appl. Math., 25 (2006), pp. 1–17.
[6] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, *Filling-in by joint interpolation of vector fields and gray levels*, IEEE Trans. Image Process., 10 (2001), pp. 1200–1211.
[7] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, *PatchMatch: A randomized correspondence algorithm for structural image editing*, ACM Trans. Graph., 28 (2009).
[8] M. Bertalmio, A. Bertozzi, and G. Sapiro, *Navier-Stokes, fluid dynamics, and image and video inpainting*, in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Kauai, HI, 2001.
[9] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image inpainting*, in Proceedings of SIGGRAPH'00, New Orleans, LA, 2000, pp. 417–424.
[10] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, *Simultaneous structure and texture image inpainting*, IEEE Trans. Image Process., 12 (2003), pp. 882–889.
[11] J. Besag, *Spatial interaction and the statistical analysis of lattice systems*, J. Roy. Statist. Soc. Ser. B., 36 (1974), pp. 192–236.
[12] J. De Bonet, *Multiresolution sampling procedure for analysis and synthesis of texture images*, in Proceedings of SIGGRAPH'97, 1997, pp. 361–368.
[13] R. Bornard, E. Lecan, L. Laborelli, and J.-H. Chenot, *Missing data correction in still images and image sequences*, in Proceedings of the 10th ACM International Conference on Multimedia, 2002, pp. 355–361.
[14] F. Bornemann and T. März, *Fast image inpainting based on coherence transport*, J. Math. Imaging Vision, 28 (2007), pp. 259–278.
[15] A. Bruckstein, R. Holt, and A. Netravali, *Discrete elastica*, in Discrete Geometry for Computer Imagery, Springer, Berlin, 1996, pp. 59–72.
[16] A. Bugeau and M. Bertalmío, *Combining texture synthesis and diffusion for image inpainting*, in Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP'09), Lisboa, Portugal, 2009, pp. 26–33.
[17] J. Cai, R. Chan, and Z. Shen, *A framelet-based image inpainting algorithm*, Appl. Comput. Harmon. Anal., 24 (2008), pp. 131–149.

[18] F. Cao, P. Musé, and F. Sur, *Extracting meaningful curves from images*, J. Math. Imaging Vision, 22 (2005), pp. 159–181.
[19] V. Caselles, B. Coll, and J.-M. Morel, *Topographic maps and local contrast changes in natural images*, Int. J. Comput. Vis., 33 (1999), pp. 5–27.
[20] V. Caselles, B. Coll, and J.-M. Morel, *Geometry and color in natural images*, J. Math. Imaging Vision, 16 (2002), pp. 89–105.
[21] T. F. Chan, S. H. Kang, and J. Shen, *Euler's elastica and curvature-based inpainting*, SIAM J. Appl. Math., 63 (2002), pp. 564–592.
[22] T. F. Chan and J. Shen, *Mathematical models for local nontexture inpaintings*, SIAM J. Appl. Math., 62 (2002), pp. 1019–1043.
[23] T. Chan and J. Shen, *Non-texture inpainting by curvature-driven diffusion (CDD)*, J. Vis. Comm. Image Represent., 12 (2001), pp. 436–449.
[24] T. Chan, J. Shen, and H.-M. Zhou, *Total variation wavelet inpainting*, J. Math. Imaging Vision, 25 (2006), pp. 107–125.
[25] Y. Chen, Q. Luan, H. Li, and O. Au, *Sketch-guided texture-based image inpainting*, in Proceedings of the IEEE International Conference on Image Processing (ICIP), 2006, pp. 1997–2000.
[26] G. Citti and A. Sarti, *A cortical based model of perceptual completion in the roto-translation space*, J. Math. Imaging Vision, 24 (2006), pp. 307–326.
[27] I. Coope, *Curve interpolation with nonlinear spiral splines*, IMA J. Numer. Anal., 13 (1993), pp. 327–341.
[28] A. Criminisi, P. Pérez, and K. Toyama, *Object removal by exemplar-based inpainting*, in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Vol. 2, 2003, pp. 721–728.
[29] G. Cross and A. Jain, *Markov random field texture models*, IEEE Trans. Pattern Anal. Mach. Intell., 5 (1983), pp. 25–39.
[30] A. Desolneux, L. Moisan, and J. Morel, *Edge detection by Helmholtz principle*, Int. J. Comput. Vis., 14 (2001), pp. 271–284.
[31] V. Do, G. Lebrun, L. Malapert, C. Smet, and D. Tschumperlé, *Inpainting d'images couleurs par lissage anisotrope et synthèse de textures*, in Reconnaissance des Formes et Intelligence Artificielle (RFIA'06), Tours, France, 2006.
[32] I. Drori, D. Cohen-Or, and H. Yeshurun, *Fragment-based image completion*, ACM Trans. Graph., 22 (2003), pp. 303–312.
[33] A. Efros and T. Leung, *Texture synthesis by non-parametric sampling*, in Proceedings of the International Conference on Computer Vision, Vol. 2, Kerkyra, Greece, 1999, pp. 1033–1038.
[34] A. A. Efros and W. T. Freeman, *Image quilting for texture synthesis and transfer*, in Proceedings of SIGGRAPH'01, 2001, pp. 341–346.
[35] N. El-Zehiry and L. Grady, *Fast global optimization of curvature*, in Proceedings of CVPR'10, 2010.
[36] M. Elad, J.-L. Starck, D. Donoho, and P. Querre, *Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)*, Appl. Comput. Harmon. Anal., 19 (2005), pp. 340–358.
[37] J. Elder and R. Goldberg, *Image editing in the contour domain*, IEEE Trans. Pattern Anal. Mach. Intell., 23 (2001), pp. 291–296.
[38] S. Esedoglu, S. Ruuth, and R. Tsai, *Threshold dynamics for shape reconstruction and disocclusion*, in Proceedings of the IEEE International Conference on Image Processing, 2005.
[39] S. Esedoglu and J. Shen, *Digital image inpainting by the Mumford-Shah-Euler image model*, European J. Appl. Math., 13 (2002), pp. 353–370.
[40] G. Facciolo, P. Arias, V. Caselles, and G. Sapiro, *Exemplar-based interpolation of sparsely sampled images*, in Energy Minimization Methods in Computer Vision and Pattern Recognition, D. Cremers, Y. Boykov, A. Blake, and F. Schmidt, eds., Vol. 5681, Springer, Berlin, 2009, pp. 331–344.
[41] C. Fantoni and W. Gerbino, *Contour interpolation by vector field combination*, J. Vision, 3 (2003), pp. 281–303.
[42] J. Froment, *A compact and multiscale image model based on level sets*, in Scale-Space Theories in Computer Vision, Vol. 1682, Springer, Berlin, 1999, pp. 152–163.
[43] H. Grossauer and O. Scherzer, *Using the complex Ginzburg-Landau equation for digital inpainting in 2D and 3D*, in Scale Space Methods in Computer Vision, Lecture Notes in Comput. Sci. 2695,

Springer, Berlin, 2003, pp. 225–236.
[44] B. Grünbaum and G. Shephard, *Rotation and winding numbers for planar polygons and curves*, Trans. Amer. Math. Soc., 332 (1990), pp. 169–187.
[45] O. Guleryuz, *Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-Part* i: *Theory*, IEEE Trans. Image Process., 15 (2006), pp. 539–554.
[46] O. Guleryuz, *Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-Part* ii: *Adaptive algorithms*, IEEE Trans. Image Process., 15 (2006), pp. 555–571.
[47] B. Guo, H. Shum, and Y. Xu, *Chaos Mosaic: Fast and Memory Efficient Texture Synthesis*, Technical report, Microsoft Research, Redmond, WA, 2000.
[48] C. Guo, S. Zhu, and Y. Wu, *Towards a mathematical theory of primal sketch and sketchability*, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2003, pp. 1228–1235.
[49] G. Harary and A. Tal, 3*D Euler spirals for* 3*D curve completion*, in Computational Geometry, ACM, New York, 2010, pp. 393–402.
[50] P. Harrison, *A non-hierarchical procedure for re-synthesis of complex texture*, in Proceedings of WSCG'01, Plzen, Czech Republic, 2001.
[51] D. J. Heeger and J. R. Bergen, *Pyramid based texture analysis/synthesis*, in Proceedings of SIGGRAPH'95, 1995, pp. 229–238.
[52] A. Hirani and T. Totsuka, *Combining frequency and spatial domain information for fast interactive image noise removal*, in Proceedings of SIGGRAPH'96, 1996.
[53] B. Horn, *The curve of least energy*, ACM Trans. Math. Software, 9 (1983), pp. 441–460.
[54] H. Igehy and L. Pereira, *Image replacement through texture synthesis*, in Proceedings of ICIP'97, 1997.
[55] J. Jia and C.-K. Tang, *Inference of segmented color and texture description by tensor voting*, IEEE Trans. Pattern Anal. Mach. Intell., 26 (2004), pp. 771–786.
[56] G. Kanizsa, *Organization in Vision: Essays on Gestalt Perception*, Präger, Westport, CT, 1979.
[57] N. Kawai, T. Sato, and N. Yokoya, *Image inpainting considering brightness change and spatial locality of textures and its evaluation*, in PSIVT'09, 2009, pp. 271–282.
[58] B. Kimia, I. Frankel, and A.-M. Popescu, *Euler spiral for shape completion*, Int. J. Comput. Vis., 54 (2003), pp. 159–182.
[59] N. Komodakis and G. Tziritas, *Image completion using efficient belief propagation via priority scheduling and dynamic pruning*, IEEE Trans. Image Process., 16 (2007), pp. 2649–2661.
[60] T. Leung and J. Malik, *Contour continuity in region based image segmentation*, in Proceedings of the Fifth European Conference on Computer Vision, Freiburg, Germany, 1998.
[61] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H. Shum, *Real-time texture synthesis by patch-based sampling*, ACM Trans. Graph., 20 (2001), pp. 127–150.
[62] A. Linnér, *Existence of free nonclosed Euler-Bernoulli elastica*, Nonlinear Anal., 21 (1993), pp. 575–593.
[63] J. L. Lisani, L. Moisan, P. Monasse, and J. M. Morel, *On the theory of planar shape*, Multiscale Model. Simul., 1 (2003), pp. 1–24.
[64] S. Masnou, *Disocclusion: A variational approach using level lines*, IEEE Trans. Image Process., 11 (2002), pp. 68–76.
[65] S. Masnou and J.-M. Morel, *Level lines based disocclusion*, in Proceedings of the 5th IEEE International Conference on Image Processing, Chicago, IL, 1998.
[66] D. S. Meek and D. J. Walton, *A note on finding clothoids*, J. Comput. Appl. Math., 170 (2004), pp. 433–453.
[67] P. Monasse and F. Guichard, *Fast computation of a contrast-invariant image representation*, IEEE Trans. Image Process., 9 (2000), pp. 860–872.
[68] D. Mumford, *Elastica and computer vision*, in Algebraic Geometry and Its Applications, C. Bajaj, ed., Springer-Verlag, New York, 1994, pp. 491–506.
[69] M. Nitzberg, D. Mumford, and T. Shiota, *Filtering, Segmentation and Depth*, Lecture Notes in Comput. Sci. 662, Springer-Verlag, Berlin, 1993.
[70] J. M. Ogden, E. Adelson, J. R. Bergen, and P. Burt, *Pyramid-based computer graphics*, RCA Engineer, 30 (1985), pp. 4–15.
[71] P. Pérez, M. Gangnet, and A. Blake, *Poisson image editing*, in Proceedings of SIGGRAPH'03, 2003, pp. 313–318.
[72] P. Pérez, M. Gangnet, and A. Blake, *PatchWorks: Example-Based Region Tiling for Image Editing*,

Technical report, Microsoft Research, Cambridge, UK, 2004.
[73] G. Peyré, S. Bougleux, and L. Cohen, *Non-local regularization of inverse problems*, Inverse Probl. Imaging, 5 (2011), pp. 511–530.
[74] E. Pnevmatikakis and P. Maragos, *An inpainting system for automatic image structure - texture restoration with text removal*, in Proceedings of the IEEE International Conference on Image Processing (ICIP), 2008, pp. 2616–2619.
[75] A. Rares, M. Reinders, and J. Biemond, *Constrained texture restoration*, EURASIP J. Appl. Signal Process., 2005 (2005), pp. 2758–2771.
[76] A. Rares, M. Reinders, and J. Biemond, *Edge-based image restoration*, IEEE Trans. Image Process., 14 (2005), pp. 1454–1468.
[77] T. Schoenemann, F. Kahl, and D. Cremers, *Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation*, in Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 2009.
[78] E. Sharon, A. Brandt, and R. Basri, *Completion energies and scale*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, 1997.
[79] E. P. Simoncelli and J. Portilla, *Texture characterisation via joint statistics of wavelet coefficient magnitudes*, in Proceedings of the 5th IEEE International Conference on Image Processing, Chicago, IL, 1998.
[80] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, *Image completion with structure propagation*, ACM Trans. Graph., 24 (2005), pp. 861–868.
[81] A. Telea, *An image inpainting technique based on the fast marching method*, J. Graphics Tools, 9 (2004), pp. 23–34.
[82] K. K. Thornber and L. R. Williams, *Analytic solution of stochastic completion fields*, Biol. Cybernet., 75 (1996), pp. 141–151.
[83] K. K. Thornber and L. R. Williams, *Characterizing the distribution of completion shapes with corners using a mixture of random processes*, Pattern Recognition, 33 (2000), pp. 543–553.
[84] D. Tschumperlé and R. Deriche, *Vector-valued image regularization with PDE's: A common framework for different applications*, IEEE Trans. Pattern Anal. Mach. Intell., 27 (2005), pp. 506–517.
[85] S. Ullman, *Filling-in the gaps: The shape of subjective contours and a model for their generation*, Biol. Cybernet., 25 (1976), pp. 1–6.
[86] L. Vese and S. J. Osher, *Modeling textures with total variation minimization and oscillating patterns in image processing*, J. Sci. Comput., 19 (2003), pp. 553–572.
[87] L. Wei and M. Levoy, *Fast texture synthesis using tree-structured vector quantization*, in Proceedings of SIGGRAPH'00, New Orleans, LA, 2000, pp. 479–488.
[88] J. Weickert, *Anisotropic Diffusion in Image Processing*, Teubner, Stuttgart, 1998.
[89] J. Weickert, *Coherence-enhancing shock filters*, in Pattern Recognition, Lecture Notes in Comput. Sci. 2781, B. Michaelis and G. Krell, eds., Springer-Verlag, New York, 2003.
[90] Y. Wexler, E. Shechtman, and M. Irani, *Space-time completion of video*, IEEE Trans. Pattern Anal. Mach. Intell., 29 (2007), pp. 463–476.
[91] L. R. Williams and D. W. Jacobs, *Stochastic completion fields: A neural model of illusory contour shape and salience*, Neural Comput., 9 (1997), pp. 837–858.
[92] S. Zhu, Y. Wu, and D. Mumford, *Filters, random fields and maximum entropy (FRAME)*, Int. J. Comput. Vis., 27 (1998), pp. 1–20.