A Survey on Variational Image Inpainting, Texture Synthesis and Image Completion

Işık Barış Fidaner,
Boğaziçi University

***Abstract*— In this survey, techniques developed in three distinct but related fields of study, variational image inpainting, texture synthesis and image completion, are investigated.**

Variational image inpainting involves filling narrow gaps in images. Though there are challenging alternative methods, best results are obtained by PDE-based algorithms.

Texture synthesis is reproduction of a texture from a sample. Firstly, statistical model based methods were proposed for texture synthesis. Then pixel and patch-based sampling techniques were developed, preserving texture structures better than statistical methods.

Image completion algorithms deal with the problem of filling larger gaps that involve both texture and image structure. This is a more general field of study that emerged by the combination of variational image inpainting and texture synthesis. State-of-the-art image completion techniques are exemplar-based methods that are inspired by greedy image-based texture growing algorithms, and the global image completion approach that was recently proposed to solve quality problems in exemplar-based image completion.

***Index Terms*— variational image inpainting, texture synthesis, image completion, survey.**

I. Introduction

INPAINTING is a very old practice in art. In Renaissance, artists updated medieval artwork by filling the gaps. This was called inpainting, retouching. The purpose was to filling-in the missing or damaged parts of the artistic work, and restore its unity.

This practice was eventually extended from paintings to photography and moving pictures. This time, the scratches in photos and dust spots in films were to be corrected. It was also possible to add/remove objects and elements.

In time, movies, photographs and other type of visual works were more and more digitized, and digital inpainting applications emerged. Attempts were made for automatically detecting and removing scratches from films. Some applications allowed more detailed manipulation, but were controlled manually. Some image inpainting applications are scaling-up images by superresolution, reconstructing old photographs, and removal of overlaid text or graphics.

In parallel to image inpainting there was also another field of study, texture synthesis. Synthesis of textures first emerged from methods that analyzed and extracted statistical features from textures. Then it developed as a distinct field. There are several applications of texture synthesis, including filling large image holes with textures, 3D surface covering for graphics, and creating artistic effects by using textures.

Image completion is a recent field that began with studies that combined image inpainting and texture synthesis methods. It involves filling larger regions by preserving both image structure and texture. In addition to the applications of variational image inpainting, image completion is applied for filling-in the image blocks that are lost in transmission and removing unwanted large objects from images.

In this survey, different approaches and state-of-the-art techniques in these fields of study are investigated.

II. Basic Concepts

In image inpainting, parts of the image are unknown. These parts are sets of pixels that may or may not be connected. These sets of pixels are called artifacts, gaps, scratches, holes, occluding objects or simply the unknown regions, depending on the application area. The unknown area is commonly signified by Ω. If whole image is I, then known image pixels are I - Ω.

In all inpainting methods, the known information on the image is used to fill in the gaps. However, how this information is going to be used differs extensively. Based on the nature of the information, two main approaches to image inpainting problem has been developed, but there are also methods that combine these two.

Variational Image Inpainting is focused on the continuity of the geometrical structure of an image. Image structure is conceptualized in terms of isophotes. Most variational inpainting methods involve solving partial differential equations. Variational methods are only good at filling-in small-narrow gaps in piecewise smooth images (also called cartoon images). Textured images cannot be filled by these methods.

Texture Synthesis is an area of study independent from inpainting. It is the problem of producing new instances of a texture from a smaller sample. Very different methods were developed for synthesizing textures, including statistical and image-based methods. Application of these methods for filling gaps is usually called constrained texture synthesis. It can be used for inpainting images, especially images that contain several textured areas.

Real images contain both cartoon and texture properties. Thus, neither variational methods, nor texture synthesis methods can offer the ultimate solution. There are also combined methods in image inpainting that use features from both main approaches in different ways. “Image completion” is a term that refers to general problem of filling large textured holes in images.

III. Variational Image Inpainting

Variational image inpainting considers the image as the collection of structures, shapes, objects that are separated from one another by sharp edges, but each one being smooth in itself. This kind of interpretation is natural, in the sense that human vision system has also evolved to detect objects with incomplete information. One can distinguish an object, even-though only parts of it are visible. This is called disocclusion, or "amodal completion", the unconscious process of extension of visible edges "behind" occluding object. The psychophysicists express that the continuation process depends on the smoothness and straightness of the edges and convexness of disoccluded shapes.

A. Level lines

One of the pioneering work in image inpainting was by Masnou and Morel in [1]. They considered the problem as a disocclusion problem. They defined T-junctions, the points where visible edges intersect the occluding objects. The continuation of object boundaries are performed between these T-junctions. Pairs of T-junctions must be connected, which belong to the same edge hidden "behind" the occluding object. To find corresponding pairs, they used level lines of the image.

The level lines are better than edge-detection methods, since they are more precise and reliable than edges. They are reliable, in the sense that the ordering of gray levels remains identical, independent from contrast change. The level lines representation allows contrast invariance, unlike the edge representation. Another advantage is that level lines never cross each other, and this introduces a spatial causality which makes it possible to make use of dynamic programming.

There are many limitations in this method. As T-junctions are connected by straight lines, the visible edges are broken at the occlusion boundary in the process, in other words, edge curvature is not preserved. Also, the algorithm requires a simple topology for the occluding object, it cannot have a hole in it. As a result, successful application of this method is only limited to filling very narrow gaps. However, it was the first conceptualization of inpainting as a disocclusion problem. And it was also one of the first techniques that was developed for the special problem of filling in gaps by preserving image structure.

B. PDE-based inpainting

The term "digital image inpainting" was introduced by Bertalmio *et al.* in [2]. In this work, they successfully implemented a image inpainting technique that is based on partial differential equations (PDEs). The user only marks the areas to be inpainted, and the application automatically fills them in.

In their paper, they defined image inpainting as a distinct area of study, being different from image denoising. This is because, noisy parts of the images contain information about both real data and the noise (for example, they are added in additive noise), but in image inpainting, the gaps contain no information at all.

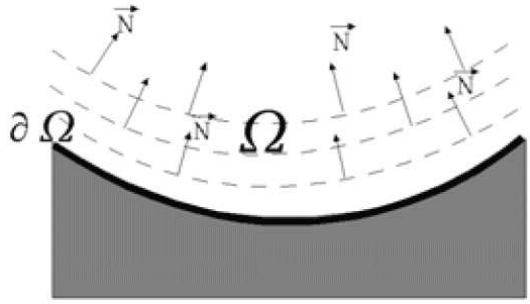

Fig. 1. Inpainting method of [2]. Propagating level lines from the region boundary inward.

After image inpainting problem was defined, Bertalmio *et al.* studied how professional restorators inpaint artwork. The basic principles are that:

1) Global picture is important. After the inpainting, the unity of the work should be restored.
2) The structure of the area surrounding the unknown region is continued into the gap, contour lines are drawn via the prolongation of those arriving at the boundary.
3) Every structure part inside the region is filled with colour that matches its colours at the region boundary.
4) Texture is added to the inpainted area.

Their algorithm is based on the second and third principles of the restorators. Therefore, it does not consider global information and the textures in the image. The algorithm is limited to inpainting in structured regions that cross through boundaries, not applicable for large textured areas.

Being a variational technique, the algorithm is based on prolonging the isophotes (level lines) arriving at the boundary, maintaining the angle of "arrival", which was not the case in the work of Masnou *et al.*. The isophotes are proceeded to be drawn inward, while curving the prolongation lines progressively to prevent them from crossing each other.

The inpainting is in the direction of level lines, which is determined as being orthogonal to image gradient and inward from region boundary. The Laplacian of the image is then propagated iteratively in this direction. In this way, both the *geometric structures* and *gray values* are propagated inward. Every few steps of iteration, the image is diffused by anisotropic diffusion. This is necessary for curving the lines with no crossing in between.

There is an example inpainting in Figure 2. Note that only narrow lines are inpainted, there are no large gaps that would

Fig. 2. An example colour image inpainted by filling missing parts. Note that there are no large gaps

Fig. 3. The blurring problem of this algorithm, continuation is not well enough

require texture inpainting. The image seems well inpainted as a whole. However, if zoomed in, there are problems in details as in Figure 3. The edges are blurred out due to diffusion, and good continuation is not satisfied. In other words, the algorithm is not contrast invariant.

C. Fluid interpretation

In [3], Bertozzi *et al.* introduced ideas from computational fluid dynamics into the image inpainting problem. They showed correspondences between concepts of fluid dynamics and concepts of variational inpainting. Fluid dynamics are a well founded area of research. Navier-Stokes equations are the differential equations that govern the Newtonian incompressible fluid behaviour by coupling pressure and velocity fields of a fluid.

In two-dimensions, the fluid stream function satisfies the same equation as the steady state image intensity equation. Therefore, this parallel between 2D incompressible fluid stream and image intensity function is exploited for developing a different approach to the inpainting problem.

In this inpainting technique, the image intensity is taken as the "stream function" for a two-dimensional incompressible flow. In this case, Laplacian of the intensity (smoothness of the image) corresponds to the vorticity of the fluid. Isophote line direction is the direction of fluid flow streamlines, and the anisotropic diffusion corresponds to fluid viscosity.

In the inpainting process, isophotes are continued while matching gradient vectors at the boundary of the unknown region. This process can also be considered as determining flow directions of a fluid at a certain part of the volume.

Firstly, the vorticity of the region boundary is computed from the image. Then, the vorticity stream is evolved into the region by a simple forward Euler time stepping. After each time step, the image intensity is computed and vorticity is recomputed for next step. Every few steps, anisotropic diffusion is applied. Steady state is achieved after several steps of computation.

The application is similar to [2], the user only marks the unknown region, no topology is assumed for the region. The region is automatically filled. This technique inherits the mathematical theory that is developed for fluid equations. These equations are well-posed and there are efficient convergent numerical methods designed to solve them. The quality is not better than [2], but the calculation times are improved from few minutes to few seconds.

D. Euler's Elastica and Bounded Variation

In their work [4], Chan *et al.* studied the Euler's Elastica and other mathematical foundations and properties of variational image inpainting models based on elasticas and curvatures. Elastica was first introduced to computer vision as a curve prior model. Then Masnou and Morel proposed an elastica-based inpainting model dependent on topology. After them, computational schemes for inpainting were based on PDEs, allowing automatical handling of different topologies of unknown regions. These PDE-based models were different but not independent from curve and elastica models.

The mathematical model for generic nontexture images is Bounded Variation (BV). This type of model successfully represents cartoon images, but is not designed for texture features. BV is based on edges, being the most crucial low-level visual cue. Modeling an image as BV simplifies the problem, as BV functions are tractable. But still, inpainting is an ill-posed problem requiring additional assumptions.

The guessing the most probable filling of the unknown region is scientifically explained as a Bayesian inference. A Bayesian inference contains two ingredients:

1) Data model is the known parts of the image.
2) Prior model is the "a priori" knowledge of the occluded object's colour and shape.

The main assumptions for BV model is about prior model having certain generic regularities, like being smooth, being convex etc. This kind of information can be processed only by methods that consider high-order geometric information such as the curvature of level sets. Euler's elastica is a second-order plane curve model. But it can become an image model, once it is applied to the level sets of an image.

E. Detecting edges

In 2005, Rares *et al.* took a different course in [5] compared to level lines and PDE-based approaches. Their technique relies on explicit edge information. They aimed to develop an inpainting technique based on the analysis of the image to reconstruct the skeletal structure around the region to be inpainted.

They defined edges with following properties: The colours of the objects they separate; estimation of continuity between edge parts; spatial ordering between edges. The edges and the areas separated by them are explicitly detected and independently inpainted by interpolating pixels from region boundary.

First phase of the algorithm is edge detection and edge feature extraction. The technique uses watershed segmentation to find contours around the unknown region, and extracts edge features such as luminance values of separated objects, local gradient along the edge (hardness), and insignificant edges are eliminated in this step.

In the second phase, edges entering and leaving the artifact are matched based on a locally circular curve model. There are also assumptions and constraints such as colour matching, noncrossing of object edges to simplify the ill-posed reconstruction problem. Edges are matched based on continuity (edge groups that fit in a circle) and sequentiality (edge groups with minimum crossing among them).

In the last phase, every pixel is filled based on the structure information constructed in the previous phase. Colours are interpolated from the unknown region boundary weighted by distance.

The successful results of this technique is limited to piecewise smooth (cartoon) images without textures. It is assumed that the unknown region having a simple topology. The interpolation uses only one-pixel-wide layer of pixels around the region.

The novelty of this paper is the enforcing of the constraint by the sequential order of edges, determining which object is behind the other. There is also the pixel filling method that takes edge directions and proximity of structure into account.

F. Heat transfer and global approach

Evolutive heat transfer is presented in [6], as another alternative to PDE-based methods. In this article, Auclair-Fortier and Ziou propose to use the global heat principle and its basic laws. This is a global approach, in contrast to solutions by discretization and PDE-based solutions to physics-based problems.

For some basic heat laws that arise from conservative principles, there is an exact global version. Moreover, basic laws can be made more comprehensible by making assumptions based on the problem and domain knowledge. First the problem is modeled, then a numerical scheme is derived from the model, so that every equation can be physically explained. In this way, a modular solution is obtained, in which modular and minimum approximations are possible.

The usual approaches to a physics-based problem are by using PDE-based local methods such as finite differences (FD), finite-element, finite-volume, or spectral methods. In these approaches, the global principle is reduced to a continuous equation. Then this continuous equation is simulated in a discrete environment. As global principles are originally stated in discrete terms, it is not necessary to put a continuous model in the process.

The global approach proposes to derive a numerical scheme directly from the global principle. Therefore, global principle is directly applied to 3D pixels (considering image and time dimensions), instead of applying PDE on points. In order to achieve this, the principle is decomposed into basic laws for a modular, tractable solution.

In the work of Bertalmio *et al.* the parallel between the PDE-based inpainting and Navier-Stokes fluid equations were shown. This kind of connection with physical phenomenon is similar to heat transfer approach. However, they realized these equations by simulating a PDE on a discrete environment.

Global approach is better than FD in terms of both quality and iterations. It converges in nearly ten fold less number of iterations than the method described in [2], and the reconstruction quality is same or superior. The quality difference is more visible at the boundaries between contrasting and clear regions.

Direct application of principles lead to a better result. The contour line is more continuous by the conservation principles than the solution by finite differences. This is because numerical discretizations such as FD tend to make oscillations and multiply deviations, violating the physics laws.

G. Strong-continuation inpainting

In a recent article, [7], Bertalmio developed a considerably better PDE-based image inpainting technique, compared to previous PDE-based techniques. In this work, propagation of level lines is expressed in terms of local neighbourhoods and a third-order PDE is derived using Taylor expansion. This PDE is proved to have the optimal accuracy among all third-order PDEs. This PDE also ensures strong continuation, restoring thin structures as well as thick ones.

A variational formulation for recovery of missing parts was proposed in [1]. But it used straight lines and could not achieve good continuation. In [2], smooth matching of curvatures was achieved, and also a parallel was shown with fluid dynamics in [3]. However, this technique was not contrast invariant. In addition, these works lacked mathematical formality in general.

Bertalmio reformulated the inpainting problem as an interpolation by propagating level lines. This propagation is formulated in terms of local neighbours.

Firstly, the formulation based on colour change between neighbours is presented. This derivation gives a second-order PDE. This equation is close to total variation method explained in [4]. In the steady state of this equation, curvature is zero, leading to straight lines. This is expected, because it is proved that a third-order PDE is necessary for matching both level line curvatures and grey values.

Then, the formulation is based on the local change of image gradient along level line, instead of image intensity. In this case, the result is a third-order PDE. It is shown that this is the optimal third-order PDE. Any other third-order PDE will approximate the equation with a higher error than this one.

The implementation is based on an explicit, forward time, finite differences scheme. Also some anisotropic diffusion is added to prevent instabilities.

Quality of the results are superior than previous ones. Other algorithms tend to disconnect structures that are thinner than the width of the gap. An example inpainting result is shown in Figure 4. The blurring and continuation problems in previous algorithms are not present.

H. State-of-the-art in variational inpainting

In this section, some of many techniques that were developed for variational digital image inpainting were explored.

Fig. 4. An example inpainting with the third-order PDE

TABLE I

PROPERTIES OF VARIATIONAL IMAGE INPAINTING METHODS

MATHEMATICAL MODEL	METHODS
Curve model	Straight [1] / Elastica [4] / Circular [5]
PDE	2nd order [2] / 3rd order [7]
Physical based	Local [3] / Global [6]
TOPOLOGY	METHODS
Topology dependent	[1], [5]
Topology independent	[2], [3], [4], [6], [7]
QUALITY	METHODS
Broken lines	[1]
Blurring and some oscillations	[2], [3]
Edge conservation	[5]
No oscillations	[6]
Strong, contrast invariant	[7]
SPEED	METHODS
Few minutes	[2], [5]
Few seconds	[1], [3]

Masnou and Morel introduced the level lines (isophotes) to the problem in [1]. Bertalmio *et al.* distinctly defined the problem; then proposed a second-order PDE solution in [2] and showed its connection with another field of study, fluid dynamics [3]. After the introduction of these methods, mathematical foundations of the problem was investigated deeper, and connections with mathematical models such as Euler's elastica and bounded variation model were discovered as in [4].

The PDE-based approaches were always dominant in variational inpainting, but there are also alternatives such as explicit detection of edges around the unknown region [5], or direct application of a global physics principle to an image [6]. In the last paper, [7], the optimal third-order PDE was derived for image inpainting.

Since the introduction of the problem, state of the art in variational image inpainting has considerably advanced in terms of both quality and speed. For example, in [3], results are obtained in few seconds of time, and the results presented in [7] have a superior quality.

IV. TEXTURE SYNTHESIS

Texture synthesis is a field of study independent from, but related to inpainting. In the general definition of this problem, an input sample of a texture is given, and the goal is to produce more of that texture.

The simplest solution is to tile the texture sample on a rectangular grid of desired size. However, even if the sample can be tiled seamlessly, the resulting larger grid structure is easily noticeable and it distorts the perception of the actual texture. More sophisticated techniques are required for reproducing the actual texture with all its features and nothing more.

The most eminent property of a texture is its regularness. A regular (also called deterministic, structured, periodic) texture is characterized by a primitive element (texton or texel) that is regularly placed on a grid or a lattice. For example, floor tiles, brick walls are regular textures, sand, smoke are non-regular. Contrarily, in non-regular (stochastic, random) textures, there is no apparent repeating pattern or local structure, but global statistical properties. In practice, every texture contains some deterministic and some stochastic properties. Quality of texture synthesis algorithms are usually tested against a validation set that includes a variety of textures including textures from most periodic to most random types.

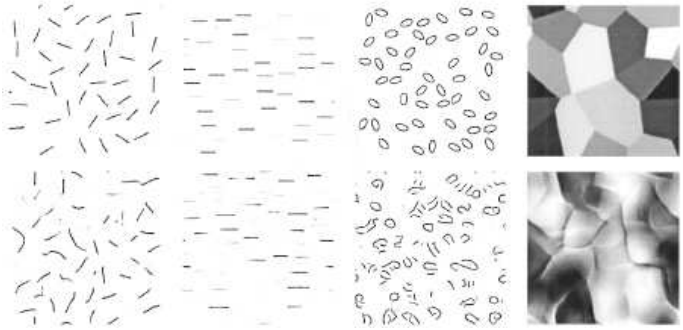

Fig. 5. Limitations of statistical texture synthesis [10].

A. Statistical model based texture synthesis

A texture is statistically described to be the result of a certain stochastic process. Result of a synthesis should be a different texture that is visually similar and it should seem to be the result of same process that could create the sample texture.

First texture synthesis techniques stemmed from this idea, and they attempted to model and match the statistical features of the sample texture. In [8], Heeger and Bergen proposed pyramid-based analysis and synthesis of textures. Image pyramid is a a Laplacian and steerable pyramid that refers to a subband transform to get images that correspond to multiple scales and orientations on the sample texture. The sample texture is statistically analysed in terms of histograms from the linear filter responses on the pyramid.

In pyramid-based synthesis, an initial noise image is modified until it matches sample statistics. As the features capture only marginal statistics; joint properties between different scales and orientations are not considered. Consequently, pyramid synthesis is less successful at producing more regular, structured textures.

A similar approach was adopted by De Bonet, in [9], by using multiresolution image pyramids to capture texture statistics, and matching histogram of filter responses. An opposite sampling approach was applied. Instead of beginning with a noise image, the algorithm started from the input sample itself, and scrambled it from coarser to finer properties to get a new texture, preserving conditional distribution of filter responses at the same time. It was better in quality, but problems with texture structure continued.

In 2000, Portilla and Simoncelli [10] took a bigger leap and developed a universal statistical model of texture. Their parametric model is based on an overcomplete complex wavelet transform. This texture model successfully captures global statistics, because it takes into account computations between pairs of coefficients across adjacent locations, orientations and scales. It involves more complicated calculations, and a significantly better quality is achieved than other statistical methods.

However, the features fail in some examples, especially in preserving non-periodic local structures as seen in Figure 5.

A more recent paper is [11]. In this study, Kwatra *et al.* formulated texture synthesis as an energy minimization problem, and developed a global optimization algorithm similar to expectation-maximization. The energy of an output was defined to be differences of every neighbourhood from their most similar matches in the input sample. This approach is not based on statistical filters like previous ones, but similar to statistical methods in the formulation of the problem as a global optimization. The approach yields good quality results for stochastic and structured textures in few minutes of computation, but bears the problem of sticking to a local minima depending on the initialization values.

TABLE II
PROPERTIES OF STATISTICAL SYNTHESIS METHODS

CRITERIA	METHODS
Marginal stats	[8], [9]
Joint stats	[10]
Global optimization	[11]
PROCESS	METHODS
Statistical feature mapping by steerable filters	[8], [9], [10]
EM-like algorithm	[11]
APPROACH	METHODS
Alter noise to match stats	[8], [10]
Scramble input sample preserving stats	[9]
QUALITY	METHODS
Limited to stochastic textures	[8]
Preserves global structures	[9], [10]
Preserves all structures	[11]
SPEED	METHODS
Several minutes	[10]
Few minutes	[11]

B. Image-based texture synthesis

In 1999, Efros and Leung [12] pioneered a different approach to texture synthesis. Instead of applying filters to the sample texture and trying to create an output that matches them, they adopted a nonparametric approach of "growing texture" from an initial seed. This was the first image-based texture synthesis technique, since they copied pixels of the sample image itself in synthesis.

They modeled the texture as a Markov Random Field (MRF). In this model, the texture is considered to be the result of a local and stationary random MRF process. Locality assumes that every pixel only depends on a local neighbourhood, a window of pixels around it, by a set of conditional probabilities. Stationarity assumes that this set of probabilities are the same for any pixel on the texture.

Nonparametric synthesis procedure is shown in Figure 6. Window size is specified by the user, determining the local neighbourhood size to be considered. Algorithm starts by putting a small block from the sample texture, then grows it pixel by pixel. For each pixel to fill, an estimate is calculated by comparing the window around it to every similar window in the sample texture. Best fitting window is determined, and its center pixel is copied to the output. Pixels are traversed in raster order, and comparison is based on L2 norm weighted by a gaussian kernel to provide better preservation of local structures.

The deterministic searching of a window in the whole sample texture considerably increases time complexity, compared to statistical methods, but quality of the results are better in a wide variety of textures (Figure 7). Thus, the new image-based approach using the local MRF model was proved to be successful in most textures. However, there are a few problems, such as slipping in the wrong part of the search space and growing garbage, or get locked at one place on sample and create verbatim copies of a part.

In [13], Wei and Levoy implemented a fast algorithm for the search step in nonparametric synthesis, by applying multiresolution concepts similar to that of statistical methods. Their technique is based on tree-structured vector quantization (TSVQ). The algorithm creates two image pyramids, one for the sample texture, and one for the output image. Searching in multiresolution pyramids decrease computation time, since spatial relations in larger scale can be modeled by smaller windows in coarser scale. Example results in Figure 8. Compared to full searching in nonparametric synthesis, TSVQ approach dramatically accelerates the synthesis process, from days of computation to a few seconds. However, quality is also noticeably reduced in preservation of local structures in some examples.

Ashikhmin observed that the quality problems in TSVQ accelerated nonparametric synthesis can be localized in a certain class of textures which are called "natural textures". These textures consist of quasi-repeating patterns of small objects with irregular size. Examples are flower fields, pebbles, bushes, tree branches.

In [14], Ashikhmin proposed a special-purpose algorithm for synthesizing natural textures. In raster order, every pixel has four previously synthesized neighbour pixels. The algorithm diminishes the search space of each pixel to only four candidates based on these neighbours. A candidate of a neighbour pixel is simply its source in sample texture, shifted into the new pixel. For example, if the left neighbour is synthesized from (x,y) on the sample, its candidate pixel is (x+1,y).

This greedy approach favours the coherency of synthesis, and reduces the problem of searching all the sample texture

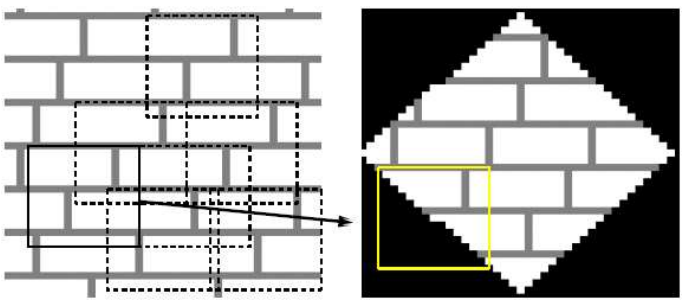

Fig. 6. Nonparametric synthesis procedure. At an iteration, some of the texture has already been synthesized. A window is considered around the new pixel to be synthesized. This window is compared to every similar window in the sample texture, by L2 norm on RGB filtered by a gaussian kernel. Center of the best fitting window is copied to the output.

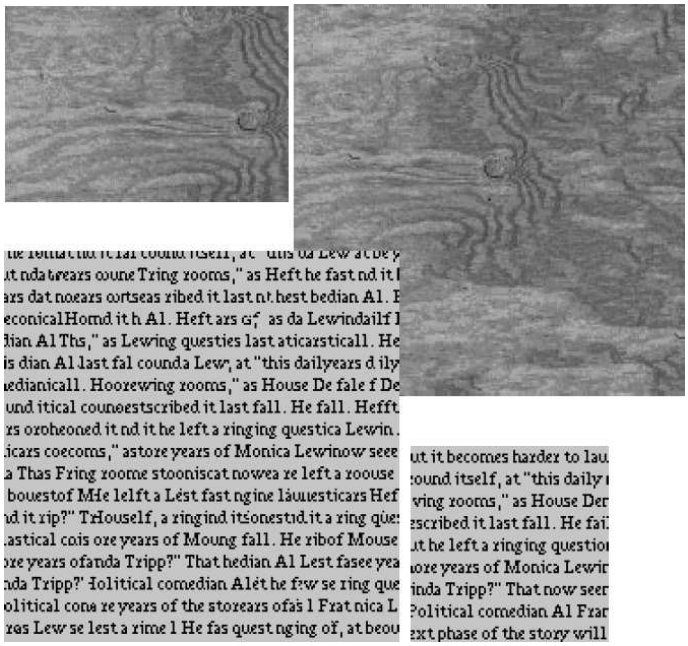

Fig. 7. Sample texture and output of image-based synthesis by nonparametric sampling [12].

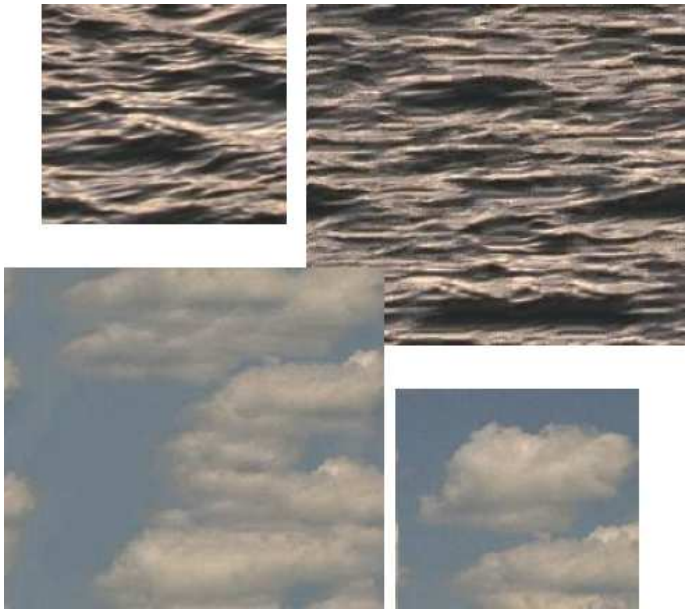

Fig. 8. Sample texture and output of fast nonparametric synthesis by TSVQ acceleration [13].

down to the question "which neighbour to continue?". In the results, many neighbour pixels are copied directly to the output, visually preserving local structures better. However, visual artifacts appear when the synthesis shifts out of the sample texture, and some kind of boundary handling is required.

Ashikhmin also implemented a user controlled synthesis process, in which the user paints a coloured target image, and the algorithm tries make the output similar to it (Figure 9).

Coherent synthesis is still a pixel-based approach in the sense that it copies one pixel at a time. Yet, it is sometimes considered as the intermediate step to patch-based synthesis

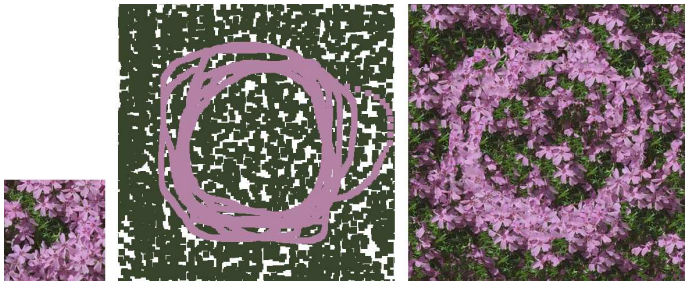

Fig. 9. Sample texture, target image, and output of controlled coherent synthesis [14].

algorithms discussed in the next part.

Capturing the analogy between images is a more general conceptualization that was proposed in [15]. It involves three input images A, A', and B. The goal is to create an "analogous" image B' that relates to B in "the same way" A' relates to A. This relation may be a standard or artistic image filter, as well as a texture formation.

To synthesize a pixel on B', the algorithm simply takes the corresponding pixel in B, and searches for a match in A, then copies the corresponding pixel in A' to the location in B'. To accelerate the process, it builds multiresolution structures as in TSVQ approach, and applies an approximate search that preserves local structures, based on Ashikhmin's coherent synthesis.

If A and B are taken constant and A' to be the desired texture, the algorithm performs texture synthesis. Despite its simple logic, image analogies have several applications including but not limited to standard texture synthesis, and another method, "texture-by-numbers". This is a controlled synthesis procedure similar to Ashikhmin's, in which pixels of A and B are not constant, but contain numbers that label regions in A' and B'. These labels let the user control which parts of A' is going to be used for synthesizing which parts

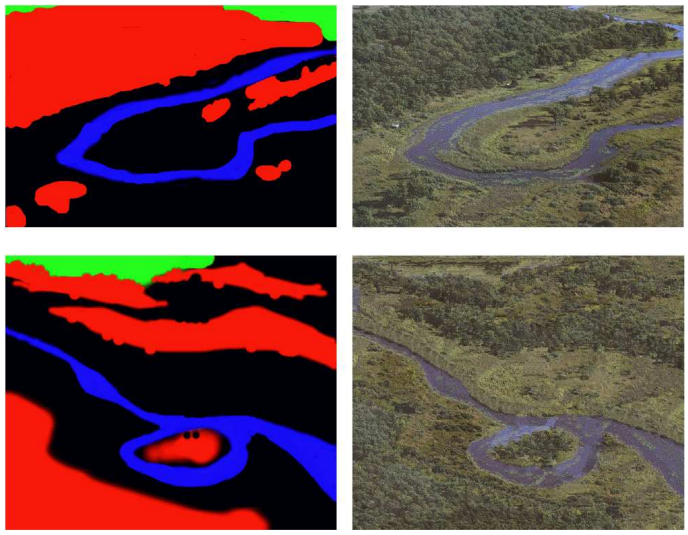

Fig. 10. Images A, A', B and B' from an example texture-by-numbers application by image analogies [15]. A' is used for sample textures and B' is synthesized.

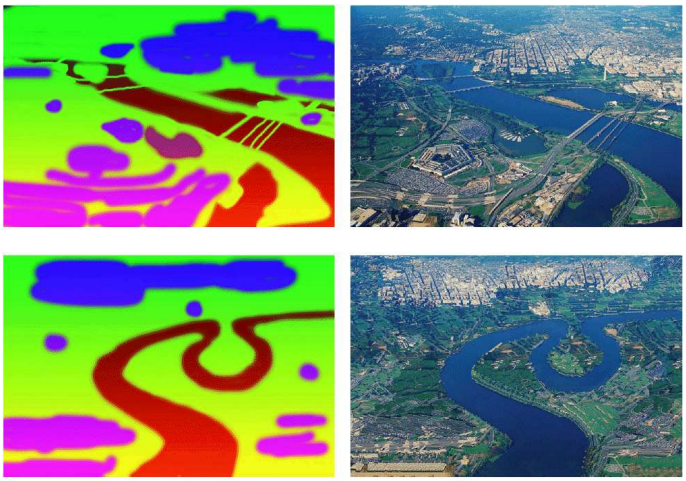

Fig. 11. Another example texture-by-numbers application [15]. Note that colour gradient is used for perspective information.

of B'. Examples are in Figures 10 and 11.

Another pixel-based approach was developed in 2006 by Lefebvre and Hoppe in [16]. They replaced the pointwise colours in the sample texture with appearance vectors that include nonlocal information including features and radiance-transfer data in 5x5 neighbourhood. These vectors are originally 100-dimensional, but reduced by principal component analysis (PCA) to only 8-dimensional vectors that cover most of the variation. Then, the calculations are moved from pixel colour space to this low dimensional Euclidean appearance-space.

Quality is improved greatly for smaller runtime neighbourhoods, because appearance vectors are richer in information than pixel colours. They also provide techniques for other applications such as anisometric synthesis, surface covering, and texture advection. The algorithm is implemented using the parallel synthesis approach involving a sequence of rasterization passes in graphics processing unit (GPU): upsampling, jitter, and correction. As a result, they achieve good quality results in real-time, 3 orders of magnitude faster than previous work.

TABLE III

PROPERTIES OF PIXEL BASED SYNTHESIS METHODS

APPROACH	METHODS
Nonparametric sampling	[12], [13], [14]
Image analogies	[15]
Appearance-space transform	[16]
QUALITY	METHODS
Preserves small local structures	[12], [13]
Larger local structures	[14]
Both small and larger local structures	[16]
SPEED	METHODS
Very slow	[12]
Order of seconds	[13], [14], [15]
Order of milliseconds	[16]

C. Patch-based synthesis methods

Pixel-based synthesis yielded good quality results in reasonable time. However, especially computer graphics applications require better preservation of local structures, and faster, real-time algorithms. This led the way to studies that consider copying entire patches instead of pixels. This approach takes the texture synthesis as a jigsaw puzzle: Putting together patches from sample texture seamlessly to produce more of the texture.

One of the first patch-based texture synthesis algorithms is image quilting [17] developed by Efros and Freeman. This is a very simple algorithm. It treats the output image as a grid of blocks, typically 32x32. Adjacent blocks overlap each other by 4 to 8 pixels. The blocks are picked from the sample texture to fit one another, in raster scan order. Overlapping regions on the left and top of the new block is matched in the search process.

The core of image quilting is the minimum error boundary cut (MEBC) algorithm that finds the most seamless contour along an overlapping region between two adjacent blocks. Firstly, error surface is calculated on the overlapping region, then minimum cost path is determined by dynamic programming. In this way, every overlapping region is cut through the best ragged edge between two adjacent patches.

Image quilting can also be used to transfer texture for different image effects. In this case, the overall output is tried to be matched to another "target image" similar to controlled synthesis in [14]. The result is a textured version of the target image.

The result quality is good for both structured and stochastic textures, and computation time is from 15 seconds to several minutes for an image. Compared to nonparametric pixel-based algorithms in previous part, image quilting is more stable and fast.

Another patch-based algorithm was developed by Liang *et al.* in [18]. In this approach, new blocks need not to be placed in a grid structure. In contrast to MEBC algorithm of image quilting, they simply use feathering (colour blending) for handling the overlapping boundaries between patches.

The crucial part of the algorithm is approximate nearest neighbours (ANN) search for a new patch that fits the overlapping boundary regions of previously synthesized patches. They invented quadtree pyramid data structure for ANN search in images. PCA is used to reduce dimensionality of search space, covering 97% of the variation in original data. The technique also handles constrained texture synthesis, or hole filling.

The result quality is not perfect, as some structures are blurred by feathering. But the optimization techniques has accelerated the synthesis process to tens of milliseconds.

An improvement on patch-based synthesis of Liang *et al.* is proposed in [19]. They use wavelets to improve quality in the results, but as wavelet computation is more expensive and there are no optimizations, real-time synthesis is not realized in this study. Texture structures are better preserved, but computation time is a few seconds to one hour per image.

Kwatra *et al.* proposed a different approach to the problem of finding a seamless patch boundary in graphcut texture synthesis [20]. Image quilting used MEBC, a memoryless dynamic programming method that only handled one dimensional contours between adjacent square blocks. In graphcut texture synthesis, patches need not be squares at all. In addi-

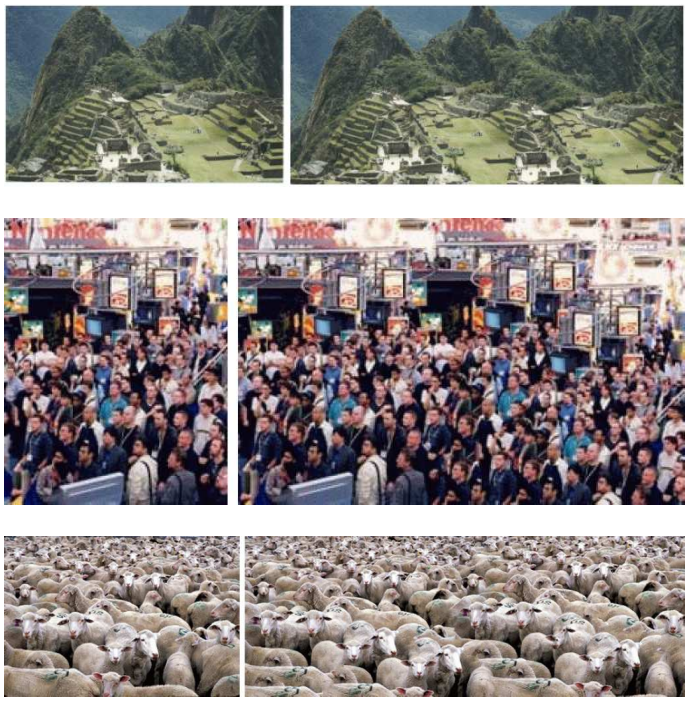

Fig. 12. Sample textures and graphcut synthesis results from [20].

tion, the patches can "remember" their cost values and adapt their boundaries for new patches. This also makes possible to put extra patches to cover seams in the output. The boundary finding method can also be used for compositing objects from different images.

The boundary region is modeled as a graph of nodes. Every node corresponds to a pixel, and arcs correspond to adjacent pixel pairs. In this formulation, the seamlessness problem becomes the min-cut or max-flow problem that is a well-known, studied problem in graph algorithms that has been efficiently solved.

Graphcut synthesis is not fast, but it gives very good quality results (Figure 12), and presents an elegant solution for the general problem of finding best boundaries between patches.

In 2004, Jump maps were proposed for texture synthesis in [21]. This study is mainly inspired by the work of Ashikmin, in which coherent synthesis is maintained by continuing parts of sample image from the previously synthesized neighbour pixels. Jump map is a representation for marking similar neighbourhoods in the sample texture. This information allows a very fast synthesis process.

In analysis phase, sample is analyzed to find all similar neighbourhood groups and mark them by a jump map. In this phase, ANN search data structure, gaussian image pyramids and PCA reduction is used for acceleration. However, analysis phase need not be as fast as synthesis, since it is made once for a sample texture.

With a known jump map, the synthesis is very simple. It starts from a random place in the sample, then copies successive pixels or patches one by one. It sometimes jumps to location in sample that is marked to be similar in the jump map. When and where to jump is probabilistically determined. Hilbert pixel ordering is used instead of raster scan, since it involves a more balanced ordering in terms of direction.

Jump map based synthesis result quality is roughly similar to that of [14], but it is very fast, in the order of tens of milliseconds. This acceleration is due to the complexity being carried from synthesis phase to analysis phase as a preprocessing in the sample texture.

Recently, Lin *et al.* selected four algorithms and quantitatively evaluated them on near-regular textures [22]. Near-regular textures are textures that have a general structure components and a stochasticity among components of this structure. These kinds of textures are most challenging for the evaluation of algorithms that can handle only structured or only stochastic textures.

Evaluation in this paper consisted of two tests. Regularity preservation test was an objective test that was based on well-defined, quantitative geometric measures to measure overall geometric regularity (G score) and appearance regularity (A score) of textures. Secondly, a subjective test followed involving 10 people giving scores to synthesis results in terms of colour quality, statistical variations and structure preservation.

The selected algorithms are graphcut synthesis, near-regular synthesis, patch-based synthesis of Liang *et al.*, and "regularized" patch-based synthesis. Near-regular textures have a general lattice structure that is like a geometrically distorted grid. Near-regular synthesis algorithm is designed specific for this type of textures. It automatically detects and exploits the underlying lattice structure in a texture. Regularized patch-based synthesis is a modified version of Liang's algorithm that additionally uses the lattice information.

For the objective test, near-regular synthesis and regularized patch-based synthesis gave the best results, showing that the lattice structure information is crucial for good quality synthesis of this type of textures. User evaluations also confirmed that near-regular synthesis gave best results, and also scores for these two algorithms were more consistent than graphcut and Liang's algorithm. This is a study that proves specific approaches give better results for specific types of textures, and yet there is no known single approach that can handle all types of textures.

D. State-of-the-art in texture synthesis

In this section, three approaches of texture synthesis were considered. These are statistical, pixel-based and patch-based texture synthesis methods.

Statistical model based synthesis methods were the first ones to be developed. They created hierarchical parametric models that covered marginal features of textures in [8], [9], and joint features in [10]. They made substantial contributions for the understanding of the underlying stochastic processes of textures. However, local structures of some textures that are encountered in practice could not be represented statistically, and this affected quality of the results.

Beginning by [12], an image-based family of algorithms were developed. These nonparametric sampling methods were pixel-based, since they copied pixels from sample texture. Firstly, brute force was used for searching, then faster algorithms could be created using different acceleration techniques

TABLE IV
PROPERTIES OF PATCH BASED SYNTHESIS METHODS

PATCH PLACEMENT	METHODS
Grid	[17], [21], [22]
Arbitrary	[18], [19], [20]
PATCH SHAPE	METHODS
Square	[17], [18], [19], [21]
Arbitrary	[20]
SYNTHESIS ORDER	METHODS
Raster scan	[17]
Hilbert	[21]
Arbitrary	[18], [19], [20]
SEAMLESS CONTOUR	METHODS
Minimum Error Boundary Cut	[17]
Graphcut	[20]
Feathering	[18], [19]
Jumpmap	[21]
QUALITY	METHODS
Very good	[17], [20], [19]
Not perfect	[18], [21]
SPEED	METHODS
Tens of milliseconds	[18], [21]
Several minutes	[17], [20], [19]

and data structures. Compared to statistical methods, quality was greatly improved. Texture structures were well preserved, except for bigger structures that could not be preserved by copying pixels.

Coherent synthesis [14] was a pixel-based method, but it favored copying neighbour pixels for preserving larger texture structures. It was the intermediate step just before patch-based synthesis methods were developed. Patch-based methods gave faster and better results in terms of structure. Some algorithms focused on accelerating the search procedure [18], [21] whereas others sought better quality by reducing errors on patch boundaries [17], [20]. A recent evaluation of patch-based synthesis algorithms on near-regular textures showed that special-purpose algorithms are necessary to handle special types of textures [22].

In addition, some of these methods allow controlled texture synthesis in which user can specify a "target image" that guides the procedure and affects general structure of the output [14], [15], [17].

Current texture synthesis algorithms are superior both in quality and speed. Especially patch-based techniques can handle a wide variety of textures in real-time. However, there are still types of textures that could not be covered by a general approach.

V. IMAGE COMPLETION

Variational image inpainting and texture synthesis are separate fields of study, though they intent to solve related problems. Inpainting aims to fill a hole by preserving image geometry, texture synthesis produces new texture from a sample.

Variational image inpainting algorithms fill an unknown region by smoothly prolonging image geometry inward in the level lines (isophote) direction. The pioneering study [2] was based on professional restorators' guidelines, except for reproduction of details and texture. Though reduced in [7], PDE-based inpainting methods produce blurred results, because of the diffusion process in these algorithms. In general, variational inpainting techniques are limited to relatively smaller unknown region with smooth gradient and no texture.

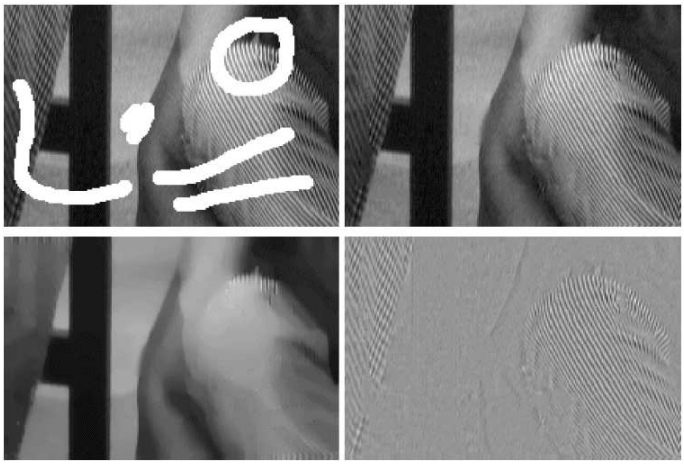

Fig. 13. Input image, output image, structure and texture inpainting results by image decomposition based technique [23].

Texture synthesis algorithms focus on synthesizing from a sample texture. The process is either based on statistical models [8], [10] or image-based nonparametric sampling [12], [13]. Some methods make possible controlled synthesis, in which the user provides structure constraints for synthesis by drawing a "target image" [14], [15], [17]. Although it is possible to fill a large unknown region by texture synthesis (texture inpainting), there should be a way to detect and force the process to fit the structure of the surrounding information.

Since few years ago, a different field of study emerged by combining these two fields. By generalizing the problem to both texture and structure inpainting, new techniques were proposed that can fill larger unknown regions with better quality and efficiency. "Image completion" is a new term that means completing large gaps in images by both structure and texture; used to discriminate the problem from "image inpainting", which usually refers to variational methods that only consider structure completion for small gaps.

A. First approaches

In [23], Bertalmio *et al.* pioneered by applying texture and structure inpainting simultaneously to fill image gaps. In their algorithm, image decomposition is used to divide the image into its structure and texture components. Then, inpainting and texture synthesis methods [2], [12] are applied to these images separately, and the result is obtained by adding them back together. The image decomposition method is based on total variation minimization for image denoising, and space of oscillating functions for modeling texture.

The results of the sum are better than both its parts, variational inpainting and texture synthesis. Advantages of two algorithms are combined in this method. The algorithm is slow, gives blurry outputs due to diffusion, and it is still limited to small gaps, but it is the first work to merge the strengths of variational inpainting and texture synthesis.

A different approach was taken by Jia *et al.* in [24]. They used adaptive ND tensor voting for image repairing. The question is to estimate normal direction at a curve endpoint given the normal at starting point. Many tensors on the tensor field that covers the visible image parts, communicate through voting to determine information in the unknown region.

The algorithm first applies image segmentation to extract curves on the image. Then, the curves are extrapolated in the image gap by tensor voting. Finally, pixel colours in the gap are filled by adaptive ND tensor voting.

Results are good, and large areas can be repaired successfully. The algorithm is notably efficient in connecting parts of curves around the region. But the technique requires image segmentation, which is a very difficult task.

B. Exemplar-based inpainting techniques

Latter algorithms are mainly inspired by nonparametric sampling as in [12]. Drori *et al.* proposed a fragment-based algorithm for image completion that could both preserve structure and texture [25].

This algorithm iteratively infers colour of unknown pixels from visible parts of the image. A confidence map is used to determine which pixels have more surrounding information available. The process starts from more confident pixels, and proceeds in a multiscale fashion from coarse to fine. In each step, a similar image fragment is found and copied to current unknown location, until image is completed. A fragment is a circular neighbourhood, and its radius is defined adaptive to its underlying structure.

Most results of fragment-based algorithm are good in quality. However, it is a complex and slow method. Completion of an image lasts a few hours.

One of the most influential works in this field is the exemplar-based image inpainting of Criminisi *et al.* [26]. It is a patch-based greedy sampling method like fragment-based completion, but simpler and faster. Inspired by [12], it simply selects locations at gap boundary (fill front), then searches for and copies matching image blocks from known regions to fill the gap.

The crucial component of the algorithm is the order of filling at the fill front. Previous approaches used concentric layer filling (onion peel) that fills layer by layer from outer contour to the center. Onion peel gives same priority to every point on the region boundary. Another fill order is presented for exemplar-based inpainting that assigns higher priority to points that lie on the continuation of image structures. This priority order helps to preserve image structure in the process as in variational inpainting.

Fig. 14. Input image, texture segmented image and inpainting result by tensor voting method [24].

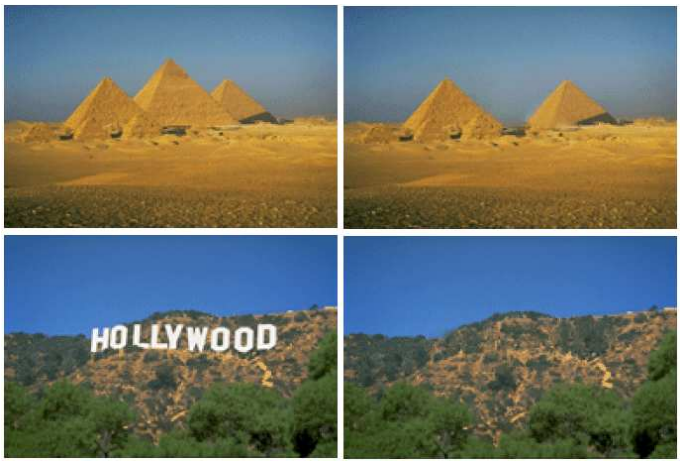

Fig. 15. Input and outputs from fragment-based inpainting [25].

The priority is determined by data term and confidence term. Confidence of a point is the known pixel proportion around a point, similar to fragment-based method. Data term is derived from measuring isophote strength orthogonal to the fill front. Two terms balance each other, and more confident points that lie on image structure are given more priority.

The results are same or superior in quality than previous approaches. No blurring is present as in [25], [23]. The algorithm also handles composite textures and "knitting" between two textures. No image segmentation required, and region topology is automatically handled. However, there are also many limitations. It cannot model and connect curves as in [24]. Boundaries of copied blocks are sometimes noticeable. Matching and copying patches sometimes creates artificial repetition by overuse of same patches. It cannot inpaint a colour gradient background.

Some improvements to this exemplar-based method were proposed recently. Nie *et al.* proposed to change fill order and a new matching cost function to solve some problems [27]. In the original algorithm, data and confidence terms are multiplied to calculate priority. In this case, if data term is zero, priority immediately becomes zero and cause visual inconsistencies. They propose addition of data and confidence, instead of multiplication. They also propose a matching cost function that encourages to sample uniformly from source, and

Fig. 16. Satellite image inpainting example from exemplar-based inpainting [26].

Fig. 17. Global image completion results [30] compared to exemplar-based inpainting [26] results that are in the rightmost column.

penalize repetition of same patches over and over again.

Wu *et al.* proposed in [28] a different data term calculation for exemplar-based inpainting. They call the original algorithm, "along isophotes" exemplar-based inpainting (AIEI), and propose to calculate "cross isophotes" strength for data term (CIEI). This calculation is based on total variation diffusion that depends on the contrast of isophotes. They present example results that seem more natural with their modification.

Using the Poisson equation Shao *et al.* proposed a decomposition-based texture and structure inpainting method [29] that divides the image and applies separate processes as in[23]. For the texture image, exemplar-based inpainting is directly incorporated. In structure image, a Laplacian is firstly applied to enhance it, then exemplar-based inpainting is applied on this Laplacian image. Finally, the structure image is reconstructed by Poisson equation and the components are added back together. The results are improved, boundaries of copied blocks are less noticeable due to distribution of error in the Poisson step.

C. Global MRF model

Despite these improvements, exemplar-based inpainting is a greedy algorithm that uses heuristics with ad hoc principles, and quality is not guaranteed as there are no global principles nor strong theoretical grounds. Komodakis and Tziritas pointed out this basic problem in exemplar-based inpainting and proposed a global optimization approach [30]. In their method, energy of the system is modeled by a lattice of patches with potentials of a discrete MRF model. Then, the optimal solution is found by Priority-BP algorithm, an accelerated belief propagation technique introduced in this paper.

Priority-BP includes "label pruning" that accelerates the process by allowing less number of source locations that can be copied to more confident points, and "priority-based message scheduling" that also speeds up by giving high priority to belief propagation from more confident points.

They presented better and more consistent results in comparison to exemplar-based inpainting, confirming their method's theoretical grounds. Moreover, the computation lasted only few seconds up to 2 minutes for an image.

TABLE V

PROPERTIES OF IMAGE COMPLETION METHODS

TEXTURE SYNTHESIS	METHODS
Pixel copying	[23]
Exemplar copying	[25], [26], [27], [28], [29]
ND tensor voting	[24]
Global optimization	[30]
STRUCTURE PRESERVATION	METHODS
Variational inpainting	[23]
Curve extrapolation	[24]
Multiscale filling	[25]
Along isophotes filling	[26], [27], [29]
Cross isophotes filling	[28]
Global optimization	[30]
QUALITY	METHODS
Best quality	[30]
Good quality	[24], [25], [26], [27], [28], [29]
Only small gaps	[23]
SPEED	METHODS
Few minutes	[23], [26], [27], [28], [29]
Several minutes	[24], [25]

D. State-of-the-art in image completion

As successful and efficient methods were developed for variational image inpainting and texture synthesis, it became possible to combine them in to solve a greater problem, image completion. The first to combine structure and texture was [23] by image decomposition. Then, a tensor voting method was proposed [24] using texture segmentation that explicitly modeled image curves.

Fragment-based image completion [25], and exemplar-based image inpainting [26] was inspired by nonparametric sampling as in [12]. Being simple and efficient, exemplar-based inpainting became influential and a number of improvements were proposed [27], [28], [29].

Recently, a global image completion approach was proposed [30], which is not a greedy algorithm like exemplar-based inpainting and does not bear several related quality problems.

VI. CONCLUSION

In this survey, several methods that were proposed in three related fields of study, variational image inpainting, texture synthesis and image completion, were presented.

Variational image inpainting approaches focus on filling narrow nontexture gaps in images by prolonging image structure. State-of-the-art inpainting methods that give best results are PDE-based algorithms [7].

Texture synthesis algorithms were developed in three approaches. Parametric models that synthesize by statistical feature matching, pixel-based methods that use nonparametric sampling and patch-based techniques, which are more enhanced sampling methods. Very good quality outputs were obtained especially by patch-based synthesis algorithms [20].

Image completion is a more general problem of filling large textured holes in images, which emerged through studies that combined image inpainting and texture synthesis techniques. State-of-the-art image completion techniques are exemplar-based algorithms [26] that are inspired by nonparametric texture synthesis, and a more recent method, global image completion [30].

REFERENCES

[1] S. Masnou and J. Morel, "Level lines based disocclusion," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*.

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[3] M. Bertalmio, A. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," 2001.

[4] T. Chan, S. Kang, and J. Shen, "Euler's elastica and curvature based inpaintings," *J. Appl. Math.*, 2002.

[5] A. Rares, M. J. T. Reinders, and J. Biemond, "Edge-based image restoration." *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1454–1468, 2005.

[6] D. Z. M.-F. Auclair-Fortier, "A global approach for solving evolutive heat transfer for image denoising and inpainting," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2558– 2574, 2006.

[7] M. Bertalmio, "Strong-continuation, contrast-invariant inpainting with a third-order optimal pde." *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1934–1938, 2006.

[8] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *SIGGRAPH*, 1995, pp. 229–238.

[9] J. S. D. Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," *Computer Graphics*, vol. 31, no. Annual Conference Series, pp. 361–368, 1997.

[10] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49–70, 2000.

[11] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*. New York, NY, USA: ACM Press, 2005, pp. 795–802.

[12] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *ICCV (2)*, 1999, pp. 1033–1038.

[13] L.-W. Wey and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Siggraph 2000, Computer Graphics Proceedings*, K. Akeley, Ed. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000, pp. 479–488.

[14] M. Ashikhmin, "Synthesizing natural textures," in *SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*. New York, NY, USA: ACM Press, 2001, pp. 217–226.

[15] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *SIGGRAPH 2001, Computer Graphics Proceedings*, E. Fiume, Ed. ACM Press / ACM SIGGRAPH, 2001, pp. 327–340.

[16] S. Lefebvre and H. Hoppe, "Appearance-space texture synthesis," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*. New York, NY, USA: ACM Press, 2006, pp. 541–548.

[17] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *SIGGRAPH 2001, Computer Graphics Proceedings*, E. Fiume, Ed. ACM Press / ACM SIGGRAPH, 2001, pp. 341–346.

[18] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, "Real-time texture synthesis by patch-based sampling," *ACM Trans. Graph.*, vol. 20, no. 3, pp. 127–150, 2001.

[19] L. Tonietto, M. Walter, and C. R. Jung, "Patch-based texture synthesis using wavelets," in *SIBGRAPI '05: Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing*. Washington, DC, USA: IEEE Computer Society, 2005, p. 383.

[20] V. Kwatra, A. Schodl, I. Essa, and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," 2003.

[21] S. Zelinka and M. Garland, "Jump map-based interactive texture synthesis," *ACM Trans. Graph.*, vol. 23, no. 4, pp. 930–962, 2004.

[22] W.-C. Lin, J. Hays, C. Wu, Y. Liu, and V. Kwatra, "Quantitative evaluation of near regular texture synthesis algorithms," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 427–434.

[23] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *cvpr*, vol. 02, p. 707, 2003.

[24] J. Jia and C.-K. Tang, "Image repairing: Robust image synthesis by adaptive nd tensor voting," *cvpr*, vol. 01, p. 643, 2003.

[25] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," in *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*. New York, NY, USA: ACM Press, 2003, pp. 303–312.

[26] P. P. A. Criminisi and K. Toyama, "Region filling and object removal by exemplar-based inpainting," *IEEE Trans. Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[27] S. X. D. Nie, L. Ma, "Similarity based image inpainting method," in *Multi-Media Modelling Conference Proceedings, 2006 12th International*, 2006, p. 4 pp.

[28] J. Wu and Q. Ruan, "Object removal by cross isophotes exemplar-based inpainting," in *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 810–813.

[29] X. Shao, Z. Liu, and H. Li, "An image inpainting approach based on the poisson equation," in *DIAL '06: Proceedings of the Second International Conference on Document Image Analysis for Libraries (DIAL'06)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 368–372.

[30] N. Komodakis, "Image completion using global optimization," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 442–452.