Review and Preview: Disocclusion by Inpainting for Image-based Rendering

Zinovi Tauber, Ze-Nian Li, Mark S.Drew

***Abstract*—*Image-based rendering* takes as input multiple images of an object and generates *photorealistic* images from novel viewpoints. This approach avoids explicitly modeling scenes by replacing the modeling phase with an object reconstruction phase. Reconstruction is achieved in two possible ways: recovering 3D point locations using multiview stereo techniques, or reasoning about consistency of each voxel in a discretized object volume space.**

The most challenging problem for image-based reconstruction is the presence of occlusions. Occlusions make reconstruction ambiguous for object parts not visible in any input image. These parts must be reconstructed in a visually acceptable way. This review presents our insights for image inpainting to provide both attractive reconstruction and a framework increasing the accuracy of depth recovery. *Digital image inpainting* refers to any methods that fill-in holes of arbitrary topology in images so that they seem to be part of the original image. Available methods are broadly classified as structural inpainting or textural inpainting. Structural inpainting reconstructs using prior assumptions and boundary conditions, while textural inpainting only considers available data from texture exemplars or other templates. Of particular interest is research of structural inpainting applied to 3D models, impressing its effectiveness for disocclusion.

Index Terms—**Disocclusion, Image Inpainting, Image-based Rendering, Depth from Stereo, Volumetric Reconstruction, View-Dependent Rendering.**

I. Introduction

A. Motivation

One of the main objectives of computer graphics is to produce realistic imagery. When scene realism is the main concern, the easiest way to achieve it is by photographing or filming the desired scene, using it on a computer. Objects in recorded scenes, special effects and computer imagery can be composited together easily using chroma-keying/blue-matting techniques. It is often desired to incorporate objects that would be difficult to acquire on film, either due to physical limitations or practical limitations such as positioning of cameras. These objects can be modeled on a computer by an artist, but this process is very laborious and rarely achieves realistic imagery. Illumination, complex shapes, materials and interaction dynamics are all very hard to model in a realistic way. The term *photorealistic* rendering refers to a computer process that generates images indistinguishable from photographs. It is difficult to re-image acquired scene into a new virtual camera in space. The two main problems are that the depths of the visible scene points are unknown, and worse, that nothing is known of the invisible points. These two problems are interrelated with object reconstruction in 3D, as we shall show. *Disocclusion* refers to the process of recovering scene information obstructed by visible points.

Realistic computer models can instead be obtained using 3D acquisition methods on existing objects or models (maquettes). Acquisition using images is versatile and results in more detail than other 3D acquisition methods. Unfortunately, image correspondences for *multiview stereo matching* are hard to accomplish reliably. Most stereo reconstruction approaches initially recover at least camera pose parameters, so that *epipolar geometry* can be established between the reference view and any additional views. The epipolar constraint for a pixel in the reference view indicates that its line of sight (LOS) ray projects to the epipolar line in another view image. Reviews of stereo correspondence and rectification formulation can be found in the books [66][67]. For dense matching, a disparity map is calculated for all pixels in the reference view by matching them to the pixels on the corresponding epipolar lines of the other views. There are many issues that complicate calculating a good match, the worst of which are occlusions. In the presence of occlusions some pixels will not have other corresponding pixels at all, and pixels on depth discontinuity boundary have mixed colors [1].

Image-based rendering techniques combine both vision and graphics processes in various interesting ways to reconstruct an object from multiple images, and reproject it to a novel view. The ability of these methods to handle occlusions, despite many innovations, is insufficient and could benefit greatly from integration with principles from digital inpainting.

For any number of cameras, reconstruction algorithms might face a family of shapes to choose from [2], all projecting identically to all camera images due to occlusions. In order to estimate the occluded object sections we need prior knowledge or assumptions about the model. Disocclusion algorithms have been studied in computer vision for purposes such as segmentation or stereo matching [3][4]. Recently, such techniques have taken a new role, that of restoration of images with occluded and damaged regions, called holes, where the location of these regions is known. Bertalmio et al. [5] have formulated the problem in terms of a diffusion of image structure elements from the hole boundary into the hole. This process was called *digital image inpainting*, a term that is borrowed from the arts used to describe a restoration process for damaged paintings. Research in the image inpainting field focuses on improving assumptions for connectivity of object structures in the hole, as well as perform inpainting of texture, and even inpainting based on statistical/template data. Image inpainting has also been performed in a rudimentary fash-

Z. Tauber is with the Department of Computing Science in Simon Fraser University, British Columbia, Canada.

ion on surfaces in 3D that had holes due to occlusions. We argue that inpainting methodology, extended to 3D in a framework where an object surface is the structure component on which applies a 3D displacement map as the texture component, will enable rendering object sections that otherwise cannot be reconstructed. Here, 3D Texture is defined as a 3D displacement map. Moreover, this framework can be used to remove noise and improve disparity matching in 3D reconstruction.

B. Overview of Surveyed Literature

Reconstruction tasks such as image restoration, object disocclusion, texture segmentation and synthesis, and surface extraction share some similar underlying principles. These problems admit a probabilistic model in which each possible state of each element is assigned some probability drawn from a random field, most commonly a *Markov Random Field (MRF)* [6][7]. Then, reconstructing an image is accomplished by finding the *Maximum A-Posteriori (MAP)* hypothesis. *Gibbs* fields can calculate the equivalent MRF probability and explicitly depend on an energy functional, which can be more easily minimized than the probability itself in order to find the MAP hypothesis. New energy functionals can be constructed that are driven by some envisioned process, rather than by explicitly modeling the likelihood of states.

Some early work on disocclusion has been done by Nitzberg, Mumford and Shiota [4]. In their work they attempted to generate outlines of objects for image segmentation and depth extraction, so T-junctions were detected in an edge map, and corresponding T-junctions were connected using an energy functional minimizing edge length and curvature. Masnou and Morel [8] had extended the idea to level-sets of images. In this way, all the gray-levels of the occluded object can be overpainted. Bertalmio et al. [5] in an inspiring paper have proposed *digital image inpainting*. Having a manually selected inpainting region Ω in an image, the objective of inpainting is to complete the image inside the hole in a reasonable way. Their initial technique was to propagate the gray levels at the hole boundary $\partial\Omega$ into the hole along *isophotes* (level lines), by formulating the inpainting as a variational problem and solving by diffusion. This diffusion coupled with anisotropic filtering was shown to have an interpretation as fluid *transportation* using Navier-Stokes equations for fluid dynamics [9], which helps with stability and speed of convergence.

Inpainting methods that involve more complicated energy functionals assume the *Bounded Variation (BV)* image model [8][10][11][12]. This model states that image level-lines cannot oscillate infinitely, and immediately suggests a simple *Total Variation (TV)* inpainting technique, which tries to minimize the curve length. However, it is commonly accepted that for reasonable reconstruction the inpainting must consider the curvature as well. Ballester and Bertalmio et al. [10] have re-cast the inpainting process to consider curvature in a form similar to the Euler elastica, whereas Chan and Shen [12] have defined a new inpainting called Curvature Driven Diffusion (CDD), and in a later paper remarkably showed how the Euler elastica encapsulates both the CCD inpainting and transportation inpainting [11].

There are many additional types of inpaintings that were proposed subsequently, including textural inpainting [13][14][15][16] which rely on texture matching and replication, or global image statistics [17], or templates matching functionals [18]. Finally, there is also research done on inpainting in 3D, by explicitly reconstructing surfaces [19] or by applying the inpainting suggested in [5] to generate a surface in a volume grid [20].

The earliest research on image-based rendering used image warping techniques to generate new views of a realistic or a computer generated scene to speed up rendering time, between two existing close views. Chen and Williams [21] have extended the idea to 3D by calculating a linear warp field between corresponding 3D points of two scenes, and interpolate for views in between. Their research tries to deal with both holes and visibility ordering. The Lightfield [22] and Lumigraph [23] provide a more accurate and complete capability of viewing from any point in the sampled space. The space is sampled regularly, and a 4D lattice of images is created. Any single view direction corresponds to a 2D slice in the space, and interpolated from nearest neighbors as necessary. Acquisition, data storage, and access are main concerns here. In Plenoptic Modelling, McMillan and Bishop [24] described a complete framework for generating new views from arbitrary viewpoints. First, cylindrical image samples are generated, and then a form of stereo matching is performed on the cylinders for dense correspondence. A new visibility ordering approach was introduced called McMillan's Depth Ordering. This approach is widely used for reprojecting 3D points to a new view without depth sorting.

Image-based rendering methods can be divided into two types of approaches: those that are based on multiview stereo correspondences and generate a single depth map [24][25][26][1][27][28][29], and those that store multiple depth per pixel, such as volumetric reconstructions, and reason about voxel visibilities using consistency checks with input images [30][2][31][32]. Inpainting techniques share are more beneficial for global correspondence techniques of multiview stereo, but are valuable for all image-based rendering approaches.

Some methods make improvements in quality and realism by using view dependent optimizations. View dependent texture mapping is performed by most methods that texture map image information [27][25][33]. These methods emphasize the effectiveness of 2D texture when some 3D shape is known, and lead us to suggest inpainting of 3D structure and texture as a more complete and photorealistic solution.

In Section II we present the evolution of image restoration and inpainting towards a more unified variational problem. Other forms of inpainting methods are presented as well. Section III shortly presents some representative techniques in image-based rendering and argues for the necessity of the inpainting methodology. Section IV summarizes the current state of research and suggest future direction of research. Additional detail for most of the research covered in this article is available in [34].

II. Disocclusion and Digital Inpainting

A. Disocclusion and Image Restoration

The problems of disocclusion and boundary continuation can be viewed as a particular area of image restoration. Most work in image restoration focuses on de-noising, enhancing colors, and recovering isolated missing pixels. Yet approaches that disocclude objects have also been attempted. For example, a hierarchical two-level *Gibbs Random Field (GRF)* model can be applied as follows: The higher-level Gibbs process restores regions' boundaries in the image using some smoothness prior, while the lower-level MRF distribution will fill each region with a synthesized texture estimated in the texture segmentation stage [6]. Still, traditional formulation for filling-in holes is limited in scope to regions where some information is known in a close proximity, and so could not handle most disocclusion problems.

Bilinear and bicubic interpolation can be used to interpolate holes, and more advanced interpolations are possible as well. Adaptive anisotropic filters can have an edge preserving property (e.g. [35]). Single pass filtering does not produce visibly convincing disocclusion, however most inpainting schemes are applied by repeated application of boundary preserving filters.

Modeling the image properly can help restoring its data reliably. There are many ways to model the image, three of which presented in [36] are: I. *Random Fields*, most prominently MRFs, capture the statistical properties of some function u, such as the image function over the discrete image domain $\mathcal{S}$. II. *Physical Processes* in a steady state, like fluid flow, can approximate image continuity. III. *Function Spaces* assume characteristics about image energy function $E(u)$. One of the most adapted function spaces in image inpainting is the *Bounded Variations (BV)* image space. A function u is in $BV(\mathcal{S})$ if its total variation Radon measure $E(u) = \int_{\mathcal{S}} |Du|$ is bounded, where the measure $Du = \nabla u$ when the gradient exists [37].

MRFs are a useful modeling tool for many applications in computer vision, from image restoration to stereo matching. According to the Hammersley-Clifford theorem an MRF of a set of random variables U is a Gibbs Random Field (GRF) on $\mathcal{S}$ when the GRF takes the form:

$$p(u) = \frac{1}{Z} e^{-\frac{1}{T}E(u)}, \;\; Z = \sum_{u \in U} e^{-\frac{1}{T}E(u)}. \tag{1}$$

where T is the temperature, and is responsible for the variability in the distribution. Z is called the *partition function* and normalizes the quantity $p(u)$ to lie in $[0,1]$. It is quite hard to compute Z since the number of computed terms is combinatorial in the size of $\mathcal{S}$. A number of approximations have been developed to solve the problem, although for most applications trying to find the MAP estimate (those having Z constant) it is enough to minimize the energy function $E(u)$. More information about modelling with MRFs/GRFs is given in [6][7]. We can combine MRFs with MAP estimations according to Bayes theorem to obtain the MAP-MRF hypothesis; the value u^* that maximizes the posterior probability $p(u|u_0)$ is:

$$u^* = arg\max_u p(u|u_0) = arg\min_u E(u_0|u) + E(u). \tag{2}$$

where u is the ideal function we wish to recover and u_0 is the observed function.

As an example of MRFs in image reconstruction, we will consider the problem of interpolating the function $u(x,y)$ on the image domain $\mathcal{S}$, generally a finite Lipschitz domain in $\mathbb{R}^2$, from sparse data. This problem is equivalent to restoration of a depth surface obtained from sparse noisy data in stereo matching. We will assume the observation model $E(u_0|u)$ is given by Eq. 3, which assumes a white (gaussian) noise $u_0 = K * u + n_G$, where u is the ideal noise free image function, u_0 is the observed noisy image function, K is a smoothing kernel, and n_G is a random value drawn from a Gaussian distribution with standard deviation σ ($*$ denotes convolution operation).

$$E(u_0, K, \sigma) = \frac{1}{2\sigma^2|\mathcal{S}|}\int_{\mathcal{S}} (u_0 - K * u)^2 dxdy. \tag{3}$$

Note that the Gaussian noise has a specific Gibbs distribution pattern, and so can be used here directly. We design the prior model $E(u)$ to maintain the continuity principle. We could choose to penalize larger first order derivatives of u, but a better smoothness assumption will require the second order derivative to be small as well. Additionally, we would like to model piecewise discontinuity. We can do so by defining a function g, and a penalty term γ for discontinuity such that

$$\lim_{\eta \to \infty} |g\prime(\eta)| = \gamma < \infty. \tag{4}$$

It is also desirable for the function g to be symmetric, nondecreasing and differentiable. A possible choice can be the truncated quadratic (used in line process model MRF for edge detection):

$$g(\eta) = \min\{\eta^2, \gamma\}. \tag{5}$$

Thus, we can write our prior model as

$$E(u(x,y)) = \int_{\mathcal{S}} [g(u_{xx}(x,y)) + 2g(u_{xy}(x,y)) + g(u_{yy}(x,y))] dxdy. \tag{6}$$

The subscripts denote partial derivatives with respect to the variables. This model is called the weak plate. Without g in the formulation above an image edge discontinuity would incur a very large penalty, and the minimization would prefer to connect adjacent patches.

The minimization of the posterior $E(u|u_0)$ can be done by a gradient decent search. However, such a function is not strictly convex, and a gradient decent method could get stuck in a local minimum. More advanced optimization techniques are possible, whether deterministic or stochastic. The interested reader is referred to variational calculus resources such as the book by Gelfand and Fomin [38].

B. Structural Inpainting

Digital image inpainting is a computer process inspired by the strokes artists use to cover up defects in paintings (e.g. [39][40]), guided by four observations reported by Bertalmio and Sapiro et al. [5]: (1) The global picture determines how to fill in the hole. (2) The contours arriving at the boundary of the hole are continued inside it. (3) The different regions inside the hole are assigned similar colors to their colors at the boundary. (4) The small details are painted in. In the inpainting problems the hole is assumed given.

Formally, we can define $\Omega \in \mathcal{S}$ to be an open bounded subset of $\mathcal{S}$ with Lipschitz continuous boundary, where $\mathcal{S}$ is a square in $\mathbb{R}^2$. Ω is called the *hole* in the image $\mathcal{S}$ that terminates at a border $\partial\Omega$. We also define $\bar{\Omega}$ as the closure of Ω and $\tilde{\Omega}$ as another open region in $\mathcal{S}$ such that $\bar{\Omega} \subset \tilde{\Omega}$. The boundary is defined as the band $B = \tilde{\Omega} \setminus \bar{\Omega}$, which includes the pixels surrounding the hole region from which we are going to diffuse information into the hole region.

Inpainting approaches use either isotropic diffusion or more complex anisotropic diffusion (few algorithmic approaches do not [41][42]). Isotropic diffusion blends all information in Ω and cannot preserve image structures, so it requires a much smaller inpainting domain. In [43], Oliveira et al. proposed a diffusion process involving repeated convolution of $\tilde{\Omega}$ with small isotropic smoothing kernels of size 3×3. The main advantage of such an approach over anisotropic diffusion is the high speed of calculations. Another common diffusion method based on Euler's energy functional is

$$E(u) = \int_{x\in\Omega} |\nabla u|^2 dx. \tag{7}$$

As we have seen in the discussion about MRFs, this is just a prior on surface smoothness, although discontinuities are not modeled here. The solution to minimizing such cost functions is given by the gradient descent search:

$$\begin{aligned} &u^0|_\Omega = \text{Some initial value; random or simple interpolation} \\ &u^{n+1} = u^n + \lambda \frac{\partial u^n}{\partial t} \end{aligned} \tag{8}$$

where λ is an acceleration (or velocity) parameter to speed up convergence of the algorithm. Superscripts indicate the time step. In the discrete case we can approximate the energy by the equation $E(u) = \sum_{x\in\Omega}(u(x+1) - u(x))^2$ which will have the following set of partial derivatives:

$$\frac{\partial}{\partial t}u(x) = \frac{\partial}{\partial u(x)}E(u) = -2(u(x-1) - 2u(x) + u(x+1)) \tag{9}$$

which is the discretized version of the Laplacian as expected [44]. Initialization to some reasonable values in the hole might speed up the convergence. While in general isotropic diffusion is noticeably imperfect for image intensities, it makes more sense for orientation diffusion, because the human visual system is sensitive to contrast and sharpness changes, but less so to curvature changes, which are also much less frequent. A more complex inpainting based on the Euler energy is found in [45], the energy defined on a curve is presented in [46], and in [47] it is adapted for highlights removal. We can similarly attempt to apply orientation diffusion, but we cannot apply it naively due to the modularity in measuring the angels. Perona has reported thoroughly on how to apply orientation diffusion in images [44]. He defines a new energy function motivated from various physical forces, which can deal with the ambiguity resulting from angle subtraction. Jiang et al. [48] proposed to extend the idea of orientation diffusion to inpainting, especially for missing DCT blocks during image transmission. Contrast changes inside the hole cannot be approximated in this way. It is also possible to learn diffusion filters as is done in [49] within an MRF framework.

Masnou and Morel [8] presented disocclusion using level lines of larger objects in images using variational formulations and level sets, yet still solved the problem using traditional vision techniques. The original variational continuation formulation was done earlier by Nitzberg, Mumford and Shiota[4], however it was done for image segmentation, and was based on T-junctions in the edge map which are generally few and unreliable in a natural image. Masnou and Morel had extended the idea to level lines which bound upper level sets defined at each gray level λ by

$$X_\lambda u = \{x \mid x \in \mathcal{S}, u(x) \geq \lambda\} \tag{10}$$

Thus the insight here is to continue each level curve into the hole left by the occluding object, rejoining the same level curve outside the hole.

Let us define more precisely the BV image model assumption that is commonly adapted. Once again define $\mathcal{S}$ as the image domain, and $u \in L^1(\mathcal{S})$ as a function over the domain. *If u's partial derivatives at each site are measured with finite total variation in $\mathcal{S}$, then u is a function of bounded variations.* The class of bounded variation functions on the domain $\mathcal{S}$ is denoted by $BV(\mathcal{S})$ [37][50][10]. We can define the total variation (TV) Radon seminorm as the energy equation

$$\begin{aligned} E_{TV}(u) &= \int_{\mathcal{S}} |Du| dx \\ &= \sup\{\int_{\mathcal{S}} u\nabla \cdot \varphi dx \mid \varphi \in C_0^\infty(\mathcal{S}), |\varphi| \leq 1\} \end{aligned} \tag{11}$$

The energy E_{TV} is minimized in the TV inpainting scheme [51][52]. A set $X \subset \mathcal{S}$ has a finite perimeter if its characteristic function $\chi_X \in BV(\mathcal{S})$ ($\chi_X = 1$ when $x \in X$ and $\chi_X = 0$ otherwise). Furthermore, the boundary length of the set X when it is Lipschitz continuous is given by $Perim(X) = E_{TV}(\chi_X)$. The TV inpainting is restated using level sets in the Coarea formula [36][50]

$$E_{TV}(u) = \int_{-\infty}^{\infty} Perim(X_\lambda u) d\lambda, \tag{12}$$

which shows that the total variation inpainting is equivalent to minimizing the length of the level lines, and results in piecewise straight level lines in the inpainting domain.

Masnou and Morel propose to minimize level lines using the functional

$$E(u) = \int_{\mathcal{S}} |Du| \left(1 + \nabla \cdot \frac{Du}{|Du|}\right) dx, \tag{13}$$

where as usual $u|_{\mathcal{S}\setminus\Omega} = u_0|_{\mathcal{S}\setminus\Omega}$ the observed function outside the hole. Note that the divergence $\nabla \cdot \frac{Du}{|Du|}$ is the curvature κ of u. This energy can be interpreted as driving the minimization of the length of the level lines (Du), and the angle total variation. Adding 1 is necessary not to discount the length despite little change in the orientation of the line. This is a particular form of the Euler elastica discussed later on. In order to minimize this equation, the T-junctions of each level line with the boundary set Ω are first detected. A dynamic programming approach can pair up all T-junctions in a way that minimizes the cost function in Eq. 13.

Bertalmio and Sapiro et al. have proposed inpainting using the mechanism of PDEs and diffusion [5]. The inpainting smoothly propagates the image information along the level lines direction (isophotes) from outside to inside the hole. The isophote direction is normal to the gradient ∇u, and is denoted by $\nabla^{\perp} u$. It is the direction of least change in gray values. The image information propagated is an image smoothness measure given by the Laplacian

$$L(u) = u_{xx} + u_{yy}. \tag{14}$$

While the energy equation is not explicitly given in this work, the propagation of the change in image smoothness along level lines is given by the PDE

$$\frac{\partial u}{\partial t} = \nabla L(u) \cdot \nabla^{\perp} u. \tag{15}$$

This evolution only applies to $u(x)$ when $x \in \Omega$, with the boundary conditions given by B. We can easily see that at a steady state when $\frac{\partial u}{\partial t} = 0$ the direction of largest information change is perpendicular to the isophotes as required. The implementation requires numerical stability considerations as discussed in [53].

It is important to notice from Eq.15 that the image smoothness will continue along isophote directions in a straight line until a conflict occurs. Therefore, it is proposed that after every few iterations of inpainting, there are a few iterations of an anisotropic diffusion that is supposed to preserve sharpness. Figure 1 shows an artificial example of two objects and an inpainting domain, and it is shown that this method can inpaint preserving shape, and (nearly so) sharpness. However, it is possible to see loss of sharpness at the inpainting regions. Additionally, these structure diffusion techniques have a general fault with inpainting textured objects, which have been addressed with limited success by textural inpainting. It follows that 3D textural inpainting is required for 3D objects as well, although both structure and texture definitions in 3D involve recovering point locations which is not necessary in 2D.

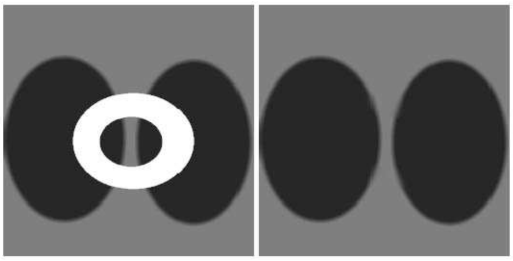

Fig. 1. Synthetic image, where Ω is the white region, and the inpainting result.

In their follow-up paper [9], Bertalmio, Bertozzi and Sapiro had restated the inpainting proposed in Eq. 15 using an analogy to fluid dynamics. The inpainting diffusion can be considered as a transport equation that convects the image intensity along level curves of the smoothness. If we define the velocity field $\boldsymbol{v} = \nabla^{\perp} u$, then u is convected by the velocity field $\boldsymbol{v}$. The image model is now formulated as Navier-Stokes governed incompressible fluid flow, where we can find a stream function Ψ such that $\nabla^{\perp}\Psi = \boldsymbol{v}$. Additionally, the vorticity is given by $\omega = \nabla \times \boldsymbol{v}$ which in 2D is a scalar equal to the Laplacian of the stream $L(\Psi)$. At the steady state we can describe the vorticity in the absence of viscosity as

$$\nabla^{\perp}\Psi \cdot \nabla L(\Psi) = 0. \tag{16}$$

The equivalence to the image diffusion process is immediately seen. The stream function Ψ is equivalent to the image function u, the fluid velocity $\boldsymbol{v} = \nabla^{\perp}\Psi$ is the isophote direction $\nabla^{\perp} u$ and the vorticity $\omega = L(\Psi)$ is the same as the image smoothness $L(u)$. The viscosity ν is the anisotropic diffusion factor. We obtain a vorticity transport equation for the image

$$\omega_t + \boldsymbol{v} \cdot \nabla \omega = \nu \nabla \cdot (g(|\nabla \omega|)\nabla \omega), \tag{17}$$

The inpainting proceeds in a similar manner to the previous method. The results are also similar to ones previously reported, yet the running time is significantly reduced due to faster convergence and better optimization using established methods in fluid dynamics. In [54][55] a similar style inpainting is suggested, generalized to a common vector-valued diffusion functional based on the trace of product of image Hessian and anisotropic tensor, which could take the form of an oriented Laplacian.

More extensive formulations have been developed using higher order PDEs and the Bounded Variations image model. Let the boundary data $u_0 \in L^{\infty}(\partial\Omega)$ and if $\boldsymbol{\theta}$ is the gradient direction vector field. A variational problem can be formulated requiring the gradient field to be in the direction of $\boldsymbol{\theta}$ after separate diffusions of both u and $\boldsymbol{\theta}$ [10]. A relaxation of this condition can be expressed as

$$\boldsymbol{\theta} \cdot \nabla u = |\nabla u|, \ |\boldsymbol{\theta}| \leq 1. \tag{18}$$

Ideally $\boldsymbol{\theta} = \frac{\nabla u}{|\nabla u|}$, yet at points where the gradient is zero $\boldsymbol{\theta}$ is null as well. We can define a functional that will help

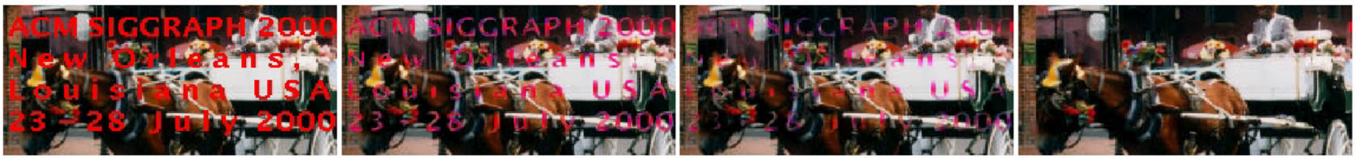

Fig. 2. Progress frames in the process of inpainting red text.

enforce such a constraint as

$$F(u) = \int_\Omega |\nabla u| - \int_\Omega \boldsymbol{\theta} \cdot \nabla u, \tag{19}$$

After some manipulation we get the equivalent minimization of the energy function

$$E(u) = \int_\Omega |\nabla u| + \int_\Omega (\nabla \cdot \boldsymbol{\theta}) \cdot u \tag{20}$$

with the admissible class of functions $\mathcal{A}(u) = \{u \in BV(\Omega) \mid |u(x)| \leq \|u_0\|_\infty, u|_B = u_0|_B\}$. The existence of a function u for such a minimization is provable.

The minimization of the energy in Eq.20 can be done in a gradient descent search by solving the Euler-Lagrange equation for u, giving rise to the PDE

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) - \nabla \cdot \boldsymbol{\theta} \tag{21}$$

This solution can reconstruct the structure inside the hole. However, discontinuities in gray levels are not modeled, and worse, the vector field $\boldsymbol{\theta}$ is unknown. Therefore, we have to additionally impose a prior on the propagation of gray levels inside Ω.

The Euler elastica curve is the equilibrium curve of the elasticity energy given by

$$\varepsilon(C) = \int_C (a + b\kappa^2) ds \tag{22}$$

where C is the curve of integration, ds is the arc length element, and $\kappa(s)$ is the Eucledean curvature [11]. The Euler elastica model imposes regularity on the smoothness of the curves allowed. The energy minimization of such a curve is the ML estimation of a geometrical walk where the rotation each step is dependent on the step size, which are exponential i.i.d. random variables.

Instead of connecting level curves individually the elastica is reformulated on the domain Ω similarly to the Coarea formula. For a level curve $u = \lambda$ we have $\frac{d\lambda}{d\theta} = |\nabla u|$. Then the integral

$$\int_0^1 \int_{C(u=\lambda)} (a + b\kappa^2) ds d\lambda = \tag{23}$$

$$\int_0^1 \int_{C(u=\lambda)} (a + b\kappa^2) ds |\nabla u| d\theta = \int_S (a + b\kappa^2) |\nabla u| dx,$$

since ds is orthogonal to $d\theta$. This translation can inpaint the entire domain. Relaxing the definition of the elastica as formulated above we can write a functional of the following form

$$E(u) = \int_{\tilde{\Omega}} |\nabla u| (a + b|\nabla \cdot \boldsymbol{\theta}|^p), \tag{24}$$

where $p \geq 1$ gives weight to the curvature, and describes a more general *p-elastica* model. As in the Masnou and Morel approach, this model attempts to minimize the length plus angular variation of level sets. If we wanted to connect curves farther away using the curvature, then b/a would be large. If, however, $a = 0$ then the reconstruction might be ambiguous when the curvature is zero. When $b = 0$ this is the TV inpainting. Chan and Shen [11] provide further analysis on Euler elastica. They suggest that for p-elastica optimization if $p \geq 3$ then the model blows up (i.e. $E(u) \to \infty$) if the function u has stationary points in Ω. They also encourage $p > 1$ to restrict sharp curvatures.

Chan and Shen further allow a relaxation of the boundary constraint by including it in the functional as follows[1]

$$E(u, c) = \int_{\tilde{\Omega}} (a + b\kappa^2) |\nabla u| dx + \frac{c}{2} \int_S (u - u_0)^2 dx. \tag{25}$$

The first term is the prior model in the Bayesian view, and the second is the the data model based on the observed image values. From the Euler-Lagrange derivation it is seen that two orthogonal processes form this inpainting. One encapsulates the transport diffusion of Eq. 15, while the other is an application of *Curvature Driven Diffusion (CDD)* [12]. Hence the Euler elastica model remarkably contains both transport and diffusion inpaintings suggested, and can be thought of as a more general model superceding previous inpaintings.

Results using this type of inpainting are shown here for a portion of the Lena image in Figure 3. Fig. 3b. shows the inpainting done on the gray levels, while 3c., 3d. and 3e., show a particular level set with inpainting of $p = 1$ in 3d. and $p = 2$ in 3e. It can be seen that for $p = 2$ the curvature is not penalized highly enough so that it seems smoother when $p = 1$.

C. Textural Inpainting

The inpainting models presented thus far have only used boundary data and prior assumptions about the smoothness of structures in the hole. For textured images, smoothness priors alone may not reconstruct the object faithfully, and a statistical or template knowledge of the

[1] In the following the curvature is replaced by a diffused curvature κ, called the weak-curvature, that the authors proved some equivalence conditions on.

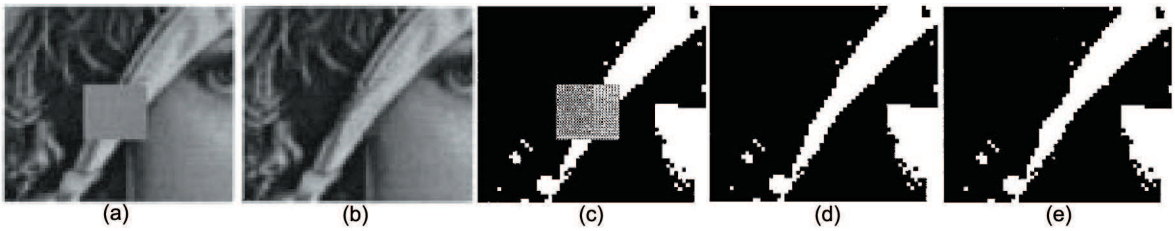

Fig. 3. Inpainting of a hole. (a) Original Lena image with hole. (b) Inpainting of the hole with $p = 1$. (c) The hole on one level set. (d) Inpainting with $p = 1$. (e) Inpainting with $p = 2$.

pattern inside the hole is needed as well. We know from MRFs that modelling the statistics of the energy function is a common practice. Likewise, there are methods that utilize global information about the image to further assist in inpainting the hole.

Levin et al. [17] suggest that global statistics about the image be estimated from outside the hole area, and the hole be inpainted with probability maximizing strokes. The approach taken by the paper is to extract some relevant statistics about the image and form a table for their probabilities and then combine them in an MRF framework. A max-product belief propagation algorithm, which is a message passing algorithm, is used for the optimization. Since there are loops in the MRF formulation, the belief propagation gives only a local maximum in a large neighborhood [56].

A substantially different approach assumes that we have the object to be inpainted in another image. Kang and Chan et al. describe such a template based inpainting approach [18]. The method has three stages: (1) finding and matching landmark points. (2) calculating the warp between the template and the current object. (3) inpainting the object.

Landmarks are any feature points p that can be extracted from the images. Given the set of feature points of the original image p_i and the template image q_i, each p_i is assigned a point q_i that minimizes some distance measure. Since the inpainting region Ω covers the object of interest, some feature points cannot be detected. The matching needs most features to be detected, so it is preferable to extract less features in the first place.

After finding the corresponding landmarks, the interpolation proceeds according to a thin-plate spline which minimizes the bending energy of u: $E = \int_S [u_{xx}^2 + u_{xy}^2 + u_{yy}^2] dxdy$. Solving the minimization is equivalent to interpolating with bi-harmonic radial basis functions such as $K(r) = r^2 \log |r * r|$. Then, the warp $U(x)$ from the given image to the template is given by

$$U(x) = Ax + t + \sum_{i=1}^{n} w_i K(|x - p_i|), \qquad (26)$$

where n is the number of corresponding landmarks.

Finally, to minimize intensity discontinuities, an inpainting approach is used to derive an optimal image u from the image given with the inpainting domain u_0 and the template u_1. The energy functional for the TV inpainting is formulated as follows

$$a_1 \int_{\mathcal{S}} |\nabla u| + a_2 \int_{\mathcal{S}\setminus\Omega} |u - u_0| + a_3 \int_{\Omega} |u - u_1(-U(x))|. \qquad (27)$$

The first term is the TV inpainting on the entire image domain, the second is the data term which tries to enforce boundary conditions, and the third is trying to make a smooth thin-plate spline interpolation. A unique insight is to try and optimize the copy of the template for the image instead of naively copying it.

One result of running this algorithm is shown in Figure 4, which shows the recovered landmarks for the girl in the middle in two different photographs, and a correction of the first photograph to make the girl smile using the whole face as an inpainting domain.

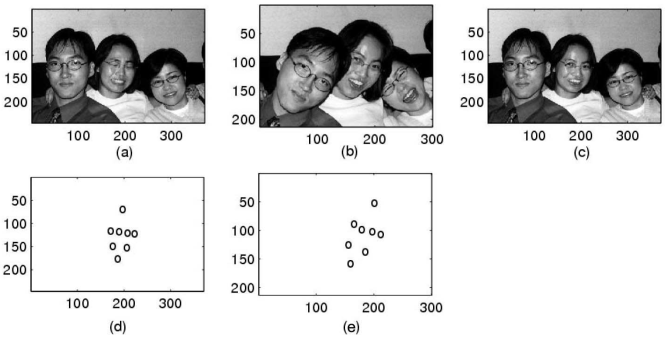

Fig. 4. Two pictures with same people in (a) and (b). (c) The face of the middle girl in (a) was inpainted with the template image (b). (d) Landmarks for the middle girl in image (a). (e) Landmarks for the middle girl in image (b).

A larger number of global information-based techniques concern textural inpainting and many more texture replication. A texture can be generated copied from examples or procedurally from statistics, though procedural methods are hard to apply for inpainting applications. The most popular texture synthesis approach for inpainting is *exemplar-based synthesis*.

Criminisi et al. [14] have proposed a patch exemplar-based inpainting where the inpainting order depends on the structure in the image. Exemplar-based texture synthesis takes windows $\Psi(i)$ centered at a pixel i, and computes the Sum of Squared Differences (SSD) match d with all available sample texels, and the best matching patch is copied [57].

Their insight is that the "inpainting front" should propagate along linear isophotes first, and secondary for all the other pixels in the hole, thus making the inpainting reasonable regardless of the boundary shape of the inpainting region. In contrast, a concentric filling-in of texture exemplars is sensitive to hole boundary. Initially, pixels in B are assigned the confidence value $C = 1$ and pixels in Ω are assigned $C = 0$. The confidence $C(i)$ and data $D(i)$ terms are defined as follows

$$C(i) = \frac{\sum_{j \in \Psi(i) \cap \Omega} C(j)}{Area(\Psi)}, \ D(i) = \frac{|\nabla^{\perp} u(i) \cdot \nabla \partial\Omega(i)|}{\gamma}. \tag{28}$$

The area of Ψ is known from the patch size and γ is a normalizing factor for intensities. The confidence term states that the more pixels that are already inpainted in the patch, and the higher confidence they each have, the higher is the confidence in the matching of the patch. The data term models the strength of the isophotes hitting the front $\partial\Omega$, and the stronger this flow is, the higher priority should be given to this patch to be duplicated first. Finally, the priority is defined for all pixels i in $\partial\Omega$ as $P(i) = C(i)D(i)$. The boundary pixels are then sorted according to the priority, and the highest priority pixel is processed first. After a matching patch has been copied, the confidence value of the patch is applied to all copied pixels, and the front $\partial\Omega$ is recalculated along with priorities for affected pixels on it.

Figure 5 shows an image inpainted first along stronger edges, and then in smoother regions. This approach has very convincing visual concealment if discontinuities can be avoided. Similar formulation has been used for inpainting in videos as well [16].

D. Structural and Textural Inpainting

Criminisi et al. [14] tried to incorporate structure into textural inpainting using a very insightful principle, where texture is inpainted in isophote direction according to its strength, which however was limited to linear structures often resulting in discontinuities where textures meet. We notice structural inpainting tries to enforce a smoothness prior yet preserve edges or isophotes. Then we can define the texture image from the decomposition $u = u_s + u_t$, where u_s is the structure component and u_t is the texture component [58][13][15]. This is a redefinition of the texture in the previous subsection, which is actual pixel colors. See Figure 6 top row for sample results from such a decomposition. The top right image is referred to as a texture image.

Bertalmio et al. have extended their inpainting to texture and structure [13] according to the decomposition above. In order to get the texture outside the hole they try to minimize the following functional:

$$E(u_s) = \int_S |\nabla u_s| + \nu \|u - u_s\|, \tag{29}$$

The first term in the energy restricts the function u_s to be in $BV(S)$, and tries to get the TV ML function u_s, while the second term tries to minimize the residual that is the texture image u_t. The balance is delicate and depends on the constant ν. It is shown that if ν is small then the decoupling of texture and structure is presentable. The norm of the texture term can be defined in a new space of oscillatory functions, which in some sense can be considered as the dual of the BV space.

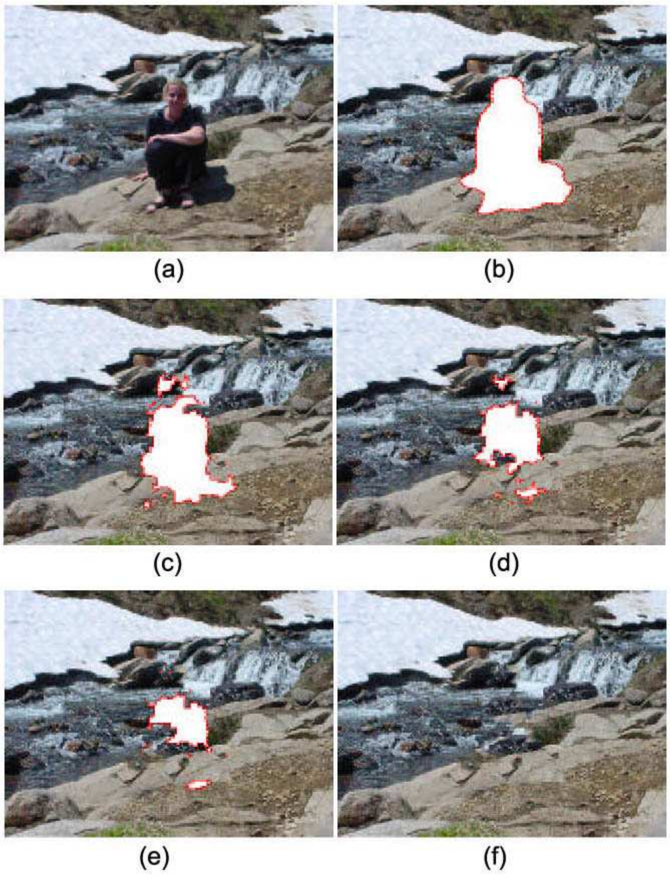

Fig. 5. Progressive stages in the texture inpainting of the image in (a).

The benefit of separating texture from structure is in the ability to inpaint each sub-image separately. For textural inpainting, an exemplar-based method is used for copying pixels rather than patches; Inpainting proceeds from the hole boundary to the center of the hole Ω with pixel values assigned only once [59]. For each hole pixel $i \in u_t$, compute the distance measure d from the neighborhood of i, to each possible boundary neighborhood. The distance measure is the SSD normalized to the number of available pixels in the neighborhood of i. The color of i is inpainted with any boundary pixel with d less than a threshold. Note that such a texture construction will not produce a unique solution, and textures may not reproduce the sample characteristics faithfully. One solution to avoid neighborhood dependency cycles as suggested by Wei et al [60] is to reference lower resolutions and previous iterations of the synthesis. Unfortunately, it is hard to see how it can be utilized for inpainting. Another approach that can alleviate the problem is to replicate patches instead of single pixels. This however, can introduce texture discontinuities as noted previously.

For the subimage u_s, inpainting proceeds as in [5] given in Eq. 15, although any other structural inpainting is applicable too.

The results given show promising improvements over non-textural inpainting. Shown here are two results: Figure 6 shows a section of a textured image and the decomposition into texture and structure, the inpaintings of both u_s and u_t, and the recomposition of u. The approach does a good job in continuing simple structures, yet over-smooths a bit at the elbow. The second result in Fig. 7 shows a comparison of the proposed method with the results of non-textural inpainting or texture synthesis alone. It is evident that this approach has a more realistic reconstruction. Also, the over-smoothing in this mixed inpainting is still easily detectable, and can suggest an inadequate separation of structure from texture. This seems to indicate that the problem of separating structure from texture is ill-posed, since arbitrary thresholds separate texture from structure edges.

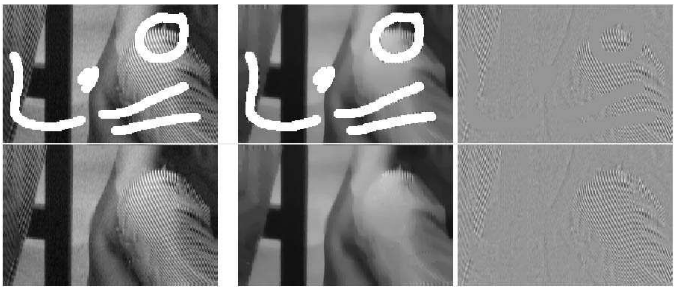

Fig. 6. Top left: original image. Top center and right: structure and texture decomposition. Bottom left: inpainting recomposition. Bottom center and right: inpainting of structure and texture images.

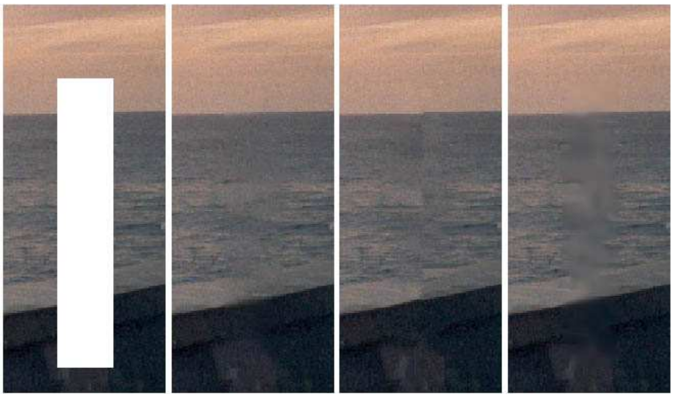

Fig. 7. From left to right: The original image with the hole, inpainting using texture and structure, using texture alone, and using non-texture inpainting.

Grossauer [15] had suggested a more extensive structural and textural inapainting framework, where first the image is filtered using a curvature minimizing diffusion to generate u_s. Then u_s is inpainted using a complex Ginzburg-Landau equation which has a diffuse-reaction form, where reaction is inversely proportional to edge width, and post-processed to clean the edges. Then the structure image is segmented using a scalar Ginzburg-Landau equation. The diffusion is constructed to have an effect like region growing that depends on image gradient, thus edges separate regions. The texture is then inpainted using the tree-based exemplar replication algorithm by Wei and Levoy as in [60]. The exemplars must be taken from the same region as the pixel to be inpainted. This way, the segmentation helps to avoid replicating mixed texture blocks. Also, segmentation of a structure image is easier than segmentation of a natural image. This algorithm improves upon structural and textural inpainting, but the same problems of separating texture and structure still prevail.

In another segmentation approach, Jia et at [61] have proposed texture inpainting where the image is first texture segmented, and then region boundary curves (considered here as structure) are continued into the hole using tensor voting and B-Splines. Then the texture is inpainted in each region using *N*D tensor voting, for a neighborhood of size N, with adaptive scaling. Texture tensor voting turns out to be a very attractive method for maintaining curvature, unfortunately the method does not perform well on complex structures. Segmentation is also quite hard to achieve correctly. However, the need for segmentation and continuation of regions generalizes into 3D as the need to continue surface separate from texture/3D detail, and thereafter the inpainting of the 3D detail and color.

E. Surface Inpainting

Inpainting is potentially a very important disocclusion step for image-based rendering if we can do so in 3D. It is important to realize that for photorealistic rendering we are not interested in volume visualization, only in rendering the visible surface. In 3D we can represent the data as a volume and use similar inpainting as in 2-space for structural inpainting. While it is possible to generalize the inpainting from 2D to 3D by inpainting on another dimension, that allows diffusion of the surface with empty space, and involves higher computational complexity than inpainting only the surface of an object.

There are numerous methods for shape reconstruction from 3D points (variational and otherwise), based on implicit surfaces (e.g. [62][63]) or explicit surfaces (e.g. [64]), which implicitly fill in holes as well. These methods do not attempt to reconstruct the real surface shape in the hole so instead the reconstruction is in the shape of the initial bounding surface. There are also methods which attempt to explicitly recover missing surfaces by fitting parameterized curves [19] to boundary date. Since explicit surfaces are fairly constrained, variational approaches are more attractive.

Verdera et al. [20] have extended their image inpainting model to 3D using the diffusion equation Eq. 15. They redefine the 3D inpainting domain for surface reconstruction as follows: Let $A \subset R^3$ be a set representative of the volume spanned by the surface. Then the observed function u_0 is defined outside the hole as $u_0 = \chi_A$. Let us define a box Q around the inpainting region. This box contains both the inpainting region and enough of u_0 to inpaint from. We can define the boundary $B = Q \setminus \bar{\Omega}$ (recall that $\bar{\Omega}$ is the closure of the hole Ω). If we construct the smallest ball possible containing the inpainting region (the gap), then we can add part of the ball's surface to cover the gap in the surface u_0 in Q. This is the initial value in the inpainting region. From here on the equivalence to 2D is established and the inpainting can proceed in a similar manner. Notice that the equivalence is between gray-scale in 2D and pixel locations in 3D. We argue that this equivalence should extend to texture as well, where 3D texture is defined properly as pixel displacements. The 2D problem is harder since texture shading and color are combined in a single pixel.

A few results of this method trying to inpaint parts of a laser scan of Michaelangelo's David which have been occluded are demonstrated in Figure 8. The reconstruction seems to be quite successful. However, it should be noted that the regions are fairly small, and as partially evident in the images here, in 3D the problem is confounded when the surfaces have much larger curvature. This result should demonstrate how effective it could be to have inpainting for depth recovery and novel view generation in image-based rendering, in particular when more advanced inpainting approaches geared for such a task are available.

Another interesting instance of inpainting in combination with surface reconstruction is given in [65]. In their research, they attempt to reconstruct the 3D surface of all objects in the scene, given a single 2D image. The reconstruction is based on a Bayesian framework that involves prior knowledge of the type of objects and other statistical principles like similarity of angles for polyhedra objects and mirror symmetry. The algorithm starts with a segmentation and primal sketch of the scene, after which it attempts to reconstruct the 3D structure of objects so that their projection to the image is preserved. The objects are reconstructed so that they maximize their probability given all the statistical principles guiding possible configurations. Thus, the reconstruction might need to add vertices, faces or paths to define the 3D shape that is most probable. This approach is limited in its application domain, and it cannot produce highly realistic images, yet it does provide many principles that inpainting could follow, and in fact seems to indicate what textural inpainting can do in 3D.

III. Inpainting in Image-based Rendering

Image-based rendering is a class of various rendering approaches in which the geometrical modeling phase of traditional rendering is replaced by an image acquisition process, and as a result the rendering phase might vary significantly from established rendering approaches. Initially, image-based rendering was primarily used to speed up rendering by tweening rendered frames. Today, a more important use is to acquire realistic objects that are very complex to model on a computer. This section presents a very short survey of image-based rendering, emphasizing the necessity of incorporating inpainting methodology.

A. View Interpolation

The earliest techniques in image-based rendering were based on warping and morphing principles. View interpolation techniques take images as input, and use them to create novel views directly by warping without reconstruction first [21][22][23]. In contrast, newer techniques involve 3D reasoning, with a depth ordered warping frequently used for reprojection.

One of the earliest 3D view interpolations was suggested by Chen and Williams [21]. In their approach it is assumed that all images have associated range data, and the camera location in scene coordinates is known. For each pair of adjacent viewpoints, an image morph map is established in scene coordinates by taking the displacement vectors of the points in the first view transformed into the second view coordinate system. Then to render a novel view from a viewpoint on the line between the two viewpoints, the first view pixels are linearly interpolated along the morph map. This warping produces the correct parallax only in two specific cases: where the camera moves in parallel with the image plane, or when the camera moves in a perpendicular direction to the image plane, in which case more work needs to be done. Special considerations for visibility ordering are needed. Image pixels are sorted according to their depth and projected in a back-to-front order to the novel view. The visibility problems that emerge are many pixels contracting into one, and pixels expanding leaving gaps. To solve for ambiguity in pixel contraction, the pixels may need to be resorted only when the view change is larger than $\pi/2$ degrees. When pixels expand however, this is a disocclusion problem. The filling-in algorithm suggested was to interpolate the four corners of the pixels and then interpolate their color in the rendered image. A different method is actually used which is faster but less accurate. It involves painting the image with a reserved color, and over-painting it with the warped image. Gaps are detected by checking for reserved colors in the image, and are interpolated using their neighboring pixels in the image. This disocclusion problem happens in many other image-based rendering techniques as well, including some that are based on volumetric reconstruction. These are normally solved in the way described above. Applying trivially an inpainting algorithm to the images generated with holes will already result in much improved rendering.

B. Multi-view Stereo Models

Stereo matching-based techniques attempt to discover the 3D structure of the scene using point correspondence from multiple views of the scene. These techniques are extended and adapted for the usage of image-based rendering. The basic approach of stereo matching is to calculate epipolar geometry using sparse point correspondences, and then use it to restrict the search space to a line for recovering a depth image by dense matching [66].

A similar disocclusion problem due to pixel expansion arises in Plenoptic modeling [24], which generates novel views using warp flow-fields. Using a form of multiview stereo-matching, Plenoptic samples are projected onto cylindrical manifolds. Then cylindrical epipolar geometry and dense correspondence are established between any two neighboring sample cylinders in space. Here, the McMil-

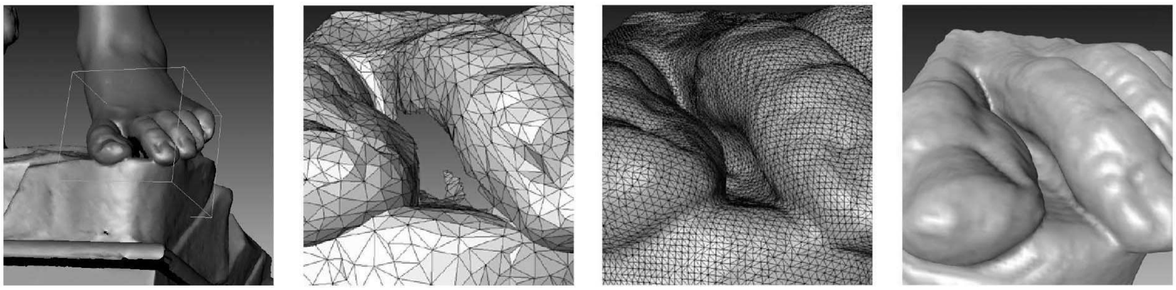

Fig. 8. The original mesh model is converted to a volume, inpainted, and then triangulated again and shaded. Shown are the mesh before and after inpainting.

lan depth ordering algorithm is first presented for warping a source cylinder to the novel viewpoint in occlusion compatible order without depth sorting. This technique can be proven to apply to any shape of projection manifold, and is commonly used for image-based renderings that involve reprojections. This warping produces the same occlusion tears in space, and inpainting is again necessary.

We can think of multiview stereo matching as a construction of a generalized disparity space in common coordinate frame that all other views are transformed into [26]. Finding the depth for a pixel consists of finding the best disparity value given the neighborhood and all images available (taking disparity to mean inverse depth). Generally, dense matching has the following steps: (1) For each disparity value a pixel can take, calculate its support from all available images using some matching cost like SSD or variance. (2) Aggregate support from neighbors in disparity space, e.g. by diffusion. (3) Find the best score disparity value at each pixel. Steps (1) and (2) are often combined for a window-based stereo matching approach. Window matching tends to smooth pixels at depth discontinuities. Steps (2) and (3) can be combined if we use a global optimization that tries to model the surface as well[2]. We have seen how to use an MRF of a robust prior model for piecewise continuous surfaces in Eq. 6 (recall the duality between reconstructing surfaces and holes). Additionally, we've seen that inpainting can reconstruct structures and holes better by using intuitive anisotropic functionals. Therefore, a straight forward application of inpainting to steps (2) and (3) above for surface extraction would already be a significant contribution to reconstructing 3D objects, the visible and occluded parts, which is not yet done. More over, steps (1) and (2) can be made better by inpainting if occluded pixels are aggregated as well over images and locations in space.

Occlusions are a two-fold problem for multiview stereo. First, they make feature detection problematic, since occlusions can look like structure features (e.g. lines and corners). Reliable features that have correspondence in multiple views are necessary for camera calibration, or projection matrix recovery, and without it the entire matching process would be inaccurate. Second, occluded pixels might not be matched even if they are visible in some images, or they might not be visible at all. Efforts to combat these effects are specialized and do not tackle the systemic problem of occlusions. Favaro et al. have suggested a way to detect different types of T-junctions [68], which should help make a more accurate detection of corner features (and incidently would also help inpainting by detecting occluded lines to be continued). Kang et al. have suggested a few techniques to help matching [69]: For area-based matching, they use spatially shiftable, temporally selectable windows; Normally, disparity support aggregation is computed over a window in image space centered at the pixel of interest p in a scheme such as SSSD (Sum of Sum of Squared Differences) where support is summed over all images. However, for pixels near a depth discontinuity the neighborhood changes in different images, causing a bad match measure. Spatially shiftable windows allow moving the window from being centered at p (near a depth discontinuity) to a location where it does not cross the depth discontinuity, such as a corner of the window. Likewise, for pixels that are occluded in some images but not in the others, temporal selection of windows attempts to heuristically select only windows in images where p is not occluded. Local aggregation techniques also exhibit more difficulties matching textureless regions, so they suggest regressively enlarging window sizes using a criterion of variance on the SSSD measure.

Kang et al. [69] also discussed a global energy minimization for surface reconstruction. Visibility reasoning is used in a similar way to volume reconstruction techniques. The error function is modified by multiplying it by a binary visibility factor that indicates whether a pixel is occluded at some depth:

$$E_{SSSD}(i) = \sum_{k \neq i} \sum_{(x,y) \in S} v(x,y,z,k) g\left(u_i(x,y) - H_i^k(z) u_k(x,y)\right), \tag{30}$$

where i is the reference image we are computing a depth map for, u is the observed image function, S is the domain of u, $z = z(x,y)$ is the depth associated with pixel (x,y), v is the visibility of a depth pixel in image k, and $H_i^k(z)$ is the homography from the plane of depth z in image k to the image plane of image i. This process proceeds iteratively by computing the pixel depths using graph cuts, and committing a percentage of the best matching depths in the next iterations to assure convergence. Graph cuts have

[2] Note that gradient descent optimization does not require knowing all the probabilities, which eliminates the need to calculate support for every disparity value for all pixels.

been shown to be usable for a class of energy functions up to three parameter clique potentials [70]. A less restrictive symmetric visibility constraint is assumed in [71].

If the surfaces in the image have parametric descriptions, and we try to model them, then our depth matching is made more accurate due to added constraints. Layered images are suggested in [31], where every image is composed of several layers with transparency. The subclass of planar layers is discussed since it allows easier view transformations via Homographies. Since scenes are hardly ever exactly planar, each pixel on a layer has also an associated offset depth from the layer. If we take this ideology further, every surface should have its own parametric description with 3D texture. Inpainting does not directly provide parametric surfaces, however it produces surfaces that are reasonable visually, and so likely provides an even stronger constraint.

Szeliski suggested a way to increase matching accuracy, in a similar way to temporal selection [26]. In order to represent pixels that are occluded in the reference view but are visible in some images, he suggested key-frame views, where each key-frame has it is own depth map consistent with the others. Finding the key-frames is problem-dependent and can be, for example, some collection of characteristic views. The *brightness compatibility* term is defined as the SSSD energy given in Eq. 30, with the summation going over all key-frames and over the support neighborhood for the key-frames (all the views that are required to match the specific key-frame view). Also, the equation is modified to include a weight w_{ik} that represents the influence of view k on the key-frame view i. Two additional terms used to enforce constraints among the multiple depth maps are: a depth compatibility constraint that requires mutual consistency between depths of neighboring key-frames, and a depth smoothness constraint that encourages depth maps to be piecewise smooth. Rendering is accomplished by warping key-frames to the new viewpoint. Combining multiple depth maps can be done by fusion techniques such as the Curless and Levoy volumetric integration technique. This fusion technique is mathematically sound, but might not reconstruct typical object curves as well as an inpainting approach would do, which would also naturally enforce the depth smoothness constraint. Inpainting though, can be applied to achieve equal results[3] without the costly multiple depth maps, by using an algorithm with visibility determination.

After getting the depth for all the pixels in the reference frames, we can render them by converting them to voxels and applying volume visualization techniques [72], or by tessellating them and rendering using raster conversion techniques [73][74]. The latter is predicated by the realization that only surfaces are extracted by image-based methods, and in-fact are all that is necessary [4]. Existing multi-view reconstruction techniques, including multi-baseline stereo described here, produce noisy and sometimes oversmooth results. By applying surface inpainting in the disparity space, we can perform better depth recovery with smooth surfaces and sharp edges, and recover occlusions, which in itself can help produce better matches. Naturally, not all objects are composed of smooth surfaces; 3D textural inpainting should be handled as well.

Increasing the number of depth maps to the number of source images, *Virtualized Reality* was proposed in [25]. A real event is virtualized in a *3D Dome* by recording in real-time the images from all cameras positioned in the Dome at regular intervals. A *Visible Surface Model* (VSM) is constructed for each camera viewpoint using a multi-baseline stereo algorithm, using at most six neighboring views. The depth data is tessellated and triangles with large depth differences are considered as a discontinuity and marked as hole triangles. Finally, the mesh is decimated not to exceed a number of error conditions. Most importantly, the mesh is required to stay more refined near depth discontinuities, since it is assumed novel views will not be far away and humans are more sensitive to depth errors near object boundaries. A VSM is rendered via standard graphics techniques such as Z-Buffering.

As the virtual camera moves in space, pixels that were occluded in any one VSM can now be visible. Thus, it is necessary to combine multiple VSMs to render an image without holes. A reference VSM and two support VSMs are chosen such that they define a bounding triangle for the virtual camera's ray. First the reference VSM is rendered, with its hole triangles rendered into a separate buffer. Then, the two supporting VSMs are rendered only for pixels that are painted in the hole buffer. Finally, holes that are still not covered are rudimentarily inpainted using interpolation.

Figure 9 shows the results of a virtual camera moving around a scene of a person swinging a baseball bat. The shading errors around the person are where a support VSM had to paint over the occlusion holes, and apparently there are shading and geometry differences between two neighboring VSMs. Here it does worse than the multiple depth map method suggested by Szeliski [26] described above, since there are no depth compatibility constraints between neighboring VSMs. A second visual inconsistency that occurs when animating the sequence is a jerky motion when the reference VSM changes. This is again due to inconsistencies in 3D geometry.

A *Complete Surface Model* (CSM) is suggested to resolve these inconsistencies and enable interaction by generating a consistent 3D surface model. The Curless and Levoy surface volume integration technique is used to join all VSMs with a marching cubes tessellation [73]. Aside from the volume accumulation method, there is no effort for detecting occlusions, which results in poorer models. Volume space projective reconstruction techniques address these problems. If inpainting methodology were used here it would have enforced surface and texture consistency, and perhaps even helped avoid the need for decimation.

Jin et al. [62] proposed a multiview stereo approach that can accommodate higher order illumination models. The

[3] Ignoring view direction dependent illumination effects.

[4] Transparency and mixed color pixels at discontinuities are generally not handled, since they are too complicated.

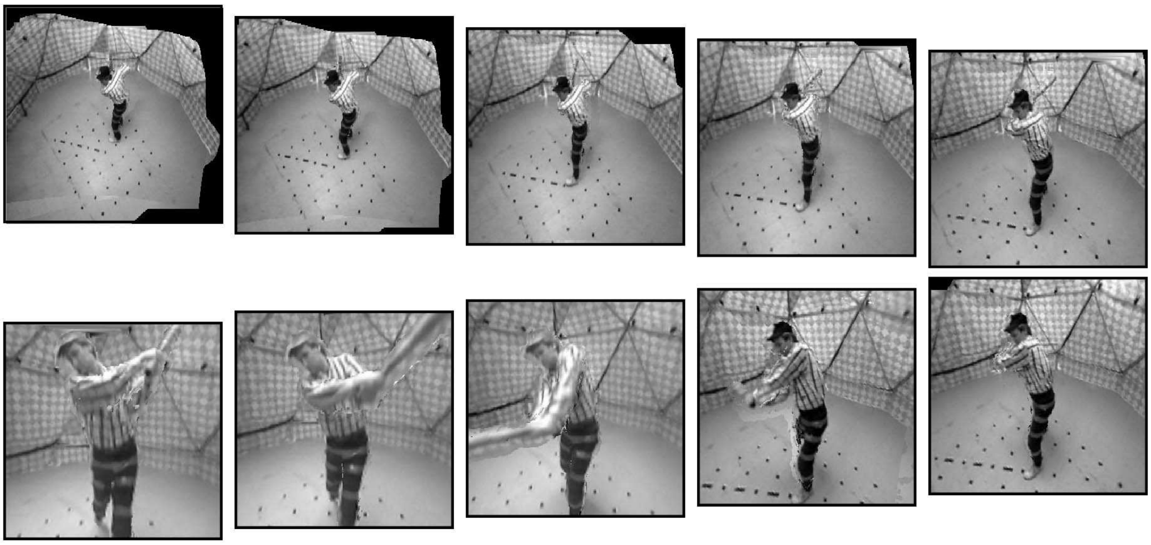

Fig. 9. The virtual camera drops down into a dynamic scene.

approach is similar to volumetric methods in that it requires projections of the object into the images to be faithful. A radiance matrix is established around each point on the current surface based on the values of the projection to the point's neighborhood too all cameras views. Then a PDE evolution is suggested for the particular case of diffuse and specular reflectance according to Ward's elliptical Gaussian model, such that the rank of the radiance matrix gets closer to 2. The evolution is supposed to converge when the current shape estimates the radiance of the true shape best. New views are created from projection of the shape and interpolation of the radiance function. This method is much less sensitive to differences in matching pixel values due to shading, and it does not have holes, but it's reconstruction precision is not very high, and the surface at hole points is in the shape of the original bounding box rather than a reasonably reconstructed shape (likewise, in [75] a surface evolution is performed such that it maintains visibility constraints). If the PDEs were adjusted to include some inpainting formulation it would be able to handle holes and reconstruct a more visually plausible surfaces.

C. Volumetric Reconstruction Methods

Volumetric reconstruction methods reconstruct the scene in discretized scene space. This, apparently, is a simpler approach for supporting visibility reasoning, and so can provide superior matching quality than the depth map methods.

The volumetric stereo reconstruction algorithm given by Szeliski [26] initially performs the same steps of matching, and aggregation as in stereo matching. To start with, voxels are filled at locations where the disparity match value is large enough. The volume is then reprojected to each camera's view plane. Reprojection is accomplished by warping volume depth layers using Homography in a back-to-front order into the desired view. After reprojection, the pixels in each image that are of the same color as the projected volume are marked as visible. This helps to determine occluded voxels in the next iteration of matching and voxel filling. He forms an optimization problem that tries to adjust voxel colors and transparencies. This work has an advantage over other volume reconstruction techniques in that it enforces a smoothness constraint instead of a specific regularization criterion that real objects do not possess.

Seitz and Dyer were among the first to suggested a volumetric reconstruction method. In their landmark paper talking about voxel coloring they suggested a provably consistent photorealistic reconstruction [30]. *Photo integrity* is defined as a reconstruction that when reprojected onto the reference images' view planes reproduces them accurately, preserving color, texture and resolution. The voxel coloring method tries to project the volume in a front-to-back order onto all the images and color only the voxels that are consistent across the images. They use a simplistic approach requiring the convex hull of all camera centers to be devoid of scene points in order to maintain proper visibility ordering. There can be many consistent scenes with a given set of images. A voxel V is considered *color invariant* with respect to a set of images if for every two consistent scenes S_1, S_2 it is in, its color in both scenes is the same. A voxel coloring of a set of m images is therefore the set of all color invariant voxels with respect to the images. The downside of it is that this reconstruction builds perturbations volume (cusps) towards the cameras.

The construction of the voxel coloring can be formulated inductively by adding one color consistent voxel to a consistent set at a time. For each projected voxel it is decided if the pixels in the images it projects onto match in color. If so, then all the pixels it projects to are marked as recovered, and they are not considered again for color matching.

Kutulakos and Seitz [2] have extended the voxel coloring idea to that of the popular *space carving* theory and approach. Space carving attempts to compute a photo-consistent shape from images of N perspective cameras located anywhere in space outside the surface itself, as long as the scene radiance is locally computable. Here they use a plane-sweeping approach to reason about the photo consistency (same as photo integrity) of each voxel, carving out each inconsistent voxel from the volume. The complete reconstruction, called the *photo hull*, subsumes all photo consistent scenes, not just visible voxels as in Voxel Coloring, but has the same protrusions towards the cameras. They claim that after getting such an "equivalence class" of all consistent shapes it is possible and even advantageous to apply other a-priori constraints such as smoothness on it. Inpainting can provide better constraints and will also be useful in modelling occlusion. In addition to recovering holes and hidden faces, inpainting provides continuation principles that are structure and texture preserving, and thus are significantly better for reconstruction than simple smoothness assumptions. However, finding an appropriate error function for retrieving the true shape from a photo-hall might be quite complicated.

Space carving utilizes a plane sweep approach as in voxel coloring by noticing that for cameras that lie in front of the plane P, any voxel on the plane P cannot be occluded by voxels behind the plane P, regardless of camera orientation. Hence, during the front-to-back plane sweep, surface voxels are projected to each camera view in front of the sweep plane, and their consistency is checked. In order to consider consistency among all images – not just those visible in front of the sweep plane – a multi-sweep space carving algorithm repeats the plane sweep carving on each principle axis x, y, and z, in a positive and negative direction. Voxels that had consistency check performed on them by at least two sweep planes also undergo a global consistency check for all cameras they are visible in.

For photorealism, the number of voxels required is very large, and the number of images required could be relatively large as well (from a dozen to over a hundred). The camera calibration has to be very exact too, here calibrated with sub-pixel precision. The Gargoyle statue rendered in Fig. 10 shows some results of this approach. In the Gargoyle renderings we can see that the surface is not accurate and there are holes. The inaccuracies are from the bulging effects and insufficient voxel resolution, and the holes were carved since there were some inaccuracies due to illumination effects or other noise. Applying continuation principles here as in stereo matching is necessary but not sufficient. We need an inpainting formulation to account for holes as well. It could additionally reduce the number of images required to acquire.

One interesting way to represent volumes so that image based rendering is facilitated fast is given by *Layered Depth Images* (LDI) [31]. A layered depth image is the projective space volume of a single virtual camera, with few points on each ray added as necessary. The LDI can be created using modified ray tracers or from real images using an adaptation of voxel coloring. Thus, the LDI takes slightly more space than an image with depth, but can handle occlusion. The rendering of the LDI from a novel viewpoint is decomposed into incremental warping computations (using a scan line approach). Proper visibility ordering is maintained using McMillan's depth ordering algorithm extended to the LDI. For each reprojected LDI ray, the depth information is rendered in a back-to-front manner, blending the previous output pixel values. The rendering itself is accomplished using splatting with quantized alpha values to facilitate Gaussian type kernels. The LDI provides a fast parallax preserving rendering, but is limited by construction to only close-by views of the scene. An interesting observation is that if we partition surface and 3D texture, then some of the multiple depths per ray are actually texture representation. If we change the projection manifold to the surface shape, the points along each ray will represent texture alone, and the entire process can be made more photorealistic, especially when coupled with inpainting.

Layered depth images were used for the purpose of representing the entire 3D structure of objects in *Image-Based Objects* (IBOs) [76]. IBO are represented using six LDIs from an object centric point, the IBO COP. Multiple IBOs can be placed in the same scene and rendered efficiently.

D. View Dependent Hybrid Methods

Some improvements in reprojection quality can be accomplished by using view dependent reconstruction for the novel view point. That is due to preservation of source image resolution and view dependent shading effects. We have seen it used in Virtualized Reality [25] in the selection of VSMs nearest to the new view.

A visual hull is another reconstruction result similar to space carving, but without regard to lighting. The foreground in the reference images is called the silhouette, and shooting rays from an image COP through all its foreground pixels creates a cone-like volume with the silhouette cross-section. All the volumes of all the images are combined using Constructive Solid Geometry (CSG) intersection operation. When the number of images from different view angles approaches infinity then the intersection is called the visual hull of the object. The visual hull cannot reconstruct concave surfaces. Typically the reconstruction is done in volume space, as with space carving, which has similar granularity limitations, although improvements have been suggested using Delaunay triangulations [77], they still do not provide image resolution. An *Image-Based Visual Hull* (IBVH) construction is suggested in [33] in order to avoid using volume space. For each pixel in the desired view image, its viewing ray is projected onto all reference images, then intersected with the silhouettes obtaining a list of intervals, and then raised back to the 3D viewing ray, for intersection of all such intervals to produce the visual hall section for that ray. Coherence between the rays is given by the epipolar constraint, which indicates that the rays form a fan shape in the reference images. Therefore, a scanline algorithm can be used, as long as the silhouette edges are sorted angularly. The shading of the visual hull is performed by selecting the color for the visible visual hull point from the reference image with closest

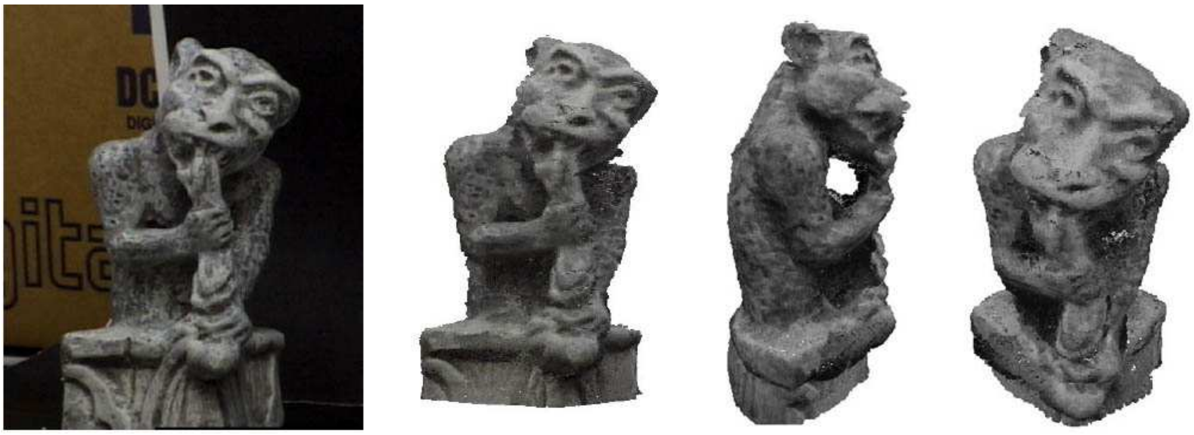

Fig. 10. Gargoyle statue source image and renderings.

angle to the desired ray in the new view. This selection is done for every ray, and it approximates view dependent shading effects. To implement visibility checks efficiently, 3D visibility checks are reduced to 2D by using epipolar planes.

The method results in a low detail volume with sharp edges (for few cameras), yet it is smoother for surfaces than a volumetric approach. The authors argue that using the visual hull as a surface for texturing, the reconstruction is made to look more realistic than common methods. This argument too suggests 3D surface extraction through structural inpainting with textural inpainting displacement is beneficial, and since inpainting both enforces more complex smoothing and tries to keep detail it should be much more effective.

One of the earlier and powerful image-based modelling endeavors is the modelling and rendering of architectures from photographs by Debevec et al. [27]. Initially, the user has to roughly model the scene, which is thereafter matched to the input images and adjusted to proper measurements. A view-dependent texture mapping is applied on the model. It is beneficial to mix texture maps of more than one photograph using a blending function that provides a smooth transition between multiple textures in cases where the model is not fully visible from any desired photograph. Any pixels missing after the texture mapping are rudimentarily inpainted.

Buildings are likely to have more complicated 3D structure than the user had modelled with simple geometric primitives, and the 2D texture mapping illusion of depth does not extend well for angles off the original view. Therefore, it is suggested to use stereo matching to extend the 3D structure. Just as in stereo matching, where homographies are used to transform each input image into the reference image plane coordinates, photographs are warped onto the model and projected into the reference plane. Thus points on the model will have no disparity. Surfaces that do not agree with the model will have a disparity in the warped image caused by the projective distance from the corresponding model point. The epipolar geometry is preserved as well. Occlusions can be handled approximately too, since we know the modelled geometry. The results of this method are stunningly realistic, and show that not only texture mapping over an established geometry improves realism but also the geometry assists with further 3D reconstruction. To generalize the process and make it less cumbersome, we need to automatically extract the shape of the desired object, and then texture map it. We had suggested in this section that inpainting can provide the framework and mechanics for accomplishing both, and additionally recover occluded surfaces not visible in any images, thereby providing realism not yet available for photorealistic rendering.

IV. Conclusions and Future Work

This paper provides a review of digital image inpainting, and tries to make the case for the necessity of its integration with image-based rendering.

Digital image inpainting has progressed from attempts to disocclude a grey scale object in 2D using straight lines to filling-in of arbitrary holes with texture and shape, and inpainting surfaces in 3D. The current convergence of research in structural inpainting seems to suggest that a Euler p-elastica matches the perceived shape best (impressively, it is also the mathematical culmination of previous inpainting methods), and textural inpainting is accomplished with texture exemplars.

Inpainting, however, still has long ways to go before it can be as useful for image-based rendering as envisioned in this review. While it seems obvious that an inpainting functional is tailored for depth surface extraction, there are no current methods that can inpaint large holes in 3D since detail would be lost. Textural inpainting also proceeds either in a statistical fashion or along linear isophotes alone, and cannot restore more complex surfaces, where restoring surfaces that are not limited to planes is essential for 3D texture (defined by color and position). Learning global statistics might help, but it is a daunting task. Exploiting statistical principles, we can require symmetry and continuation principles to be upheld. Then the suggested 3D inpainting would separate the smooth 3D surface from the 3D texture, and inpaint both *properly* with the assistance of statistical heuristics.

References

[1] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comp. Vision*, 47(1):7–42, June 2002.

[2] K. S. Kutulakos and S. M. Seitz. A theory of shape by space carving. *Int. J. Comp. Vision, Marr Prize Special Issue*, 38(3):199–218, 2000.

[3] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure App. Math.*, XLII:577–685, 1989.

[4] M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, Segmentation and Depth.* Lecture Notes in Computer Science, Vol. 662. Springer-Verlag: Berlin, 1993.

[5] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. *Computer Graphics (SIGGRAPH 2000)*, pages 417–424, July 2000.

[6] S. Z. Li. *Markov Random Field Modeling in Image Analysis.* Springer: Tokyo, 2001.

[7] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 6:721–741, November 1984.

[8] S. Masnou and J.-M. Morel. Level lines based disoclussion. *Proc. Int. Conf. Image Processing*, pages 259–263, 1998.

[9] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. *Proc. IEEE Comp. Vision and Pattern Rec.*, December 2001.

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

[11] T. F. Chan, S. H. Kang, and J. Shen. Euler's elastica and curvature based inpaintings. *SIAM J. App. Math.*, 63(2):564–592, 2002.

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

[13] M. Bertalmio, L. Vesa, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Trans. Image Processing*, 8:882–889, August 2003.

[14] A. Criminisi, P. Prez, and K. Toyama. Object removal by exemplar-based inpainting. volume 2, pages 721–728, Madison, WI, June 2003.

[15] H. Grossauer. A combined pde and texture synthesis approach to inpainting. *Euro. Conf. Comp. Vision*, 2004.

[16] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting of occluding and occluded objects. *IEEE Proc. Int. Conf. Image Processing*, 2005.

[17] A. Levin, A. Zomet, and Y. Weiss. Learning how to inpaint from global image statistics. *Proc. IEEE Int. Conf. Comp. Vision*, pages 305–313, October 2003.

[18] S. H. Kang, T. F. Chan, and S. Soatto. Landmark based inpainting from multiple views. *TR CAM 02-11*, March 2002.

[19] S. Ivekovic and E. Trucco. Dense wide-baseline disparities from conventional stereo for immersive videoconferencing. *IEEE Proc. Int. Conf. Pattern Rec.*, pages 883–897, 2002.

[20] J. Verdera, V. Caselles, M. Bertalmio, and G. Sapiro. Inpainting surface holes. *Int. Conf. Image Processing*, September 2003.

[21] S. E. Chen and L. Williams. View interpolation for image synthesis. *Computer Graphics (SIGGRAPH 1993)*, 27(Annual Conference Series):279–288, 1993.

[22] M. Levoy and P. Hanrahan. Light field rendering. *Computer Graphics (SIGGRAPH 1996)*, pages 31–42, August 1996.

[23] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. *Computer Graphics (SIGGRAPH 1996)*, pages 43–54, August 1996.

[24] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Computer Graphics (SIGGRAPH 1995)*, pages 39–46, September 1995.

[25] P. J. Narayanan, P. W. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. *Proc. IEEE Int. Conf. Comp. Vision*, pages 3–10, January 1998.

[26] R. Szeliski. Stereo algorithms and representations for image-based rendering. *British Machine Vision Conf.*, pages 314–328, 1999.

[27] P. E. Debevec, Camillo J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Computer Graphics (SIGGRAPH 1996)*, 30(Annual Conference Series):11–20, August 1996.

[28] M. Irani, T. Hassner, and P. Anandan. What does a scene look like from a scene point? *Euro. Conf. Comp. Vision*, pages 883–897, 2002.

[29] J. Sun, H.-Y. Shum, and N.-N. Zheng. Stereo matching using belief propagation. *Euro. Conf. Comp. Vision*, 2:510–524, 2002.

[30] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. *Proc. IEEE Comp. Vision and Pattern Rec.*, pages 1067–1073, 1997.

[31] J. Shade, S. Gortler, L.-W. He, and R. Szeliski. Layered depth images. *Computer Graphics (SIGGRAPH 1998)*, pages 231–242, July 1998.

[32] G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. *Proc. IEEE Comp. Vision and Pattern Rec.*, 2005.

[33] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. pages 369–374, 2000.

[34] Z. Tauber. Disocclusion and photorealistic object reconstruction and reprojection. *SFU TR 2005-12*, 2004.

[35] D. Ramanan and K. E. Barner. Nonlinear image interpolation through extended permutation filters. *Proc. Int. Conf. Image Processing*, 2000.

[36] J. Shen. Inpainting and the fundamental problem of image processing. *SIAM News, Invited*, December 2002.

[37] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of Bounded Variations and Free Discontinuity Problems.* Oxford Science Publications. Clarendon Press, 2000.

[38] I. M. Gelfand and S. V. Fomin. *Calculus Of Variations.* Dover Publications, Inc.: Mineola, New York, 2000.

[39] S. Walden. *The Ravished Image.* St. Martin's Press: New York, 1985.

[40] D. King. *The Commissar Vanishes.* Henry Holt and Company, 1997.

[41] T. K. Shih, R. C. Chang, L. C. Lu, and L. H. Lin. Large block inpainting by color continuation analysis. *IEEE Proc. Multimedia Modelling Conf.*, pages 196 – 202, 2004.

[42] T. K. Shih and R. C. Chang. Digital inpainting survey and multilayer image inpainting algorithms. *IEEE Intl. Conf. Info. Tech. and App.*, 1:15 – 24, 2005.

[43] M. M. Oliveira, B. Bowen, R. McKenna, and Y.-S. Chang. Fast digital inpainting. *Proc. Int. Conf. Vis., Imaging and Image Processing*, pages 261–266, September 2001.

[44] P. Perona. Orientation diffusions. *IEEE Trans. Image Processing*, 7(3):457–467, 1999.

[45] J. Gu, S. Peng, and X. Wang. Digital image inpainting using monte carlo method. *IEEE Proc. Int. Conf. Image Processing*, 2:961 – 964, 2004.

[46] A. Z. B. Barcelos, M. A. Batista, A. M. Martins, and A. C. Nogueira. Level lines continuation based digital inpainting. *IEEE Proc. Brazilian Symp. Comp. Graphics and Image Proc.*, pages 50 – 57, 2004.

[47] P. Tan, S. Lin, L. Quan, and H.-Y. Shum. Highlight removal by illumination-constraint inpainting. *Proc. IEEE Int. Conf. Comp. Vision*, pages 826–835, October 2003.

[48] H. Jiang and C. Moloney. Error concealment using a diffusion based method. *IEEE Proc. Int. Conf. Image Processing*, pages 832–835, 2002.

[49] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. *Proc. IEEE Comp. Vision and Pattern Rec.*, 2:860 – 867, 2005.

[50] E. Giusti. Minimal surfaces and functions of bounded variation. 1984.

[51] T. F. Chan and J. Shen. Mathematical models for local nontexture inpainting. *SIAM J. App. Math.*, 62(3):1019–1043, 2001.

[52] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259 – 268, 1998.

[53] S. Osher and Sethian J. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *J. of Computational Physics*, (79):12–49.

[54] D. Tschumperle and R. Deriche. Vector-valued image regularization with pde's: A common framework for different applications. *Proc. IEEE Comp. Vision and Pattern Rec.*, 1:651–656, 2003.

[55] D. Tschumperle and R. Deriche. Vector-valued image regularization with pde's: A common framework for different applications. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 27:506–517, 2005.

[56] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *TR 2002-35.*

[57] Y. Q. Xu, S. C. Zhu, B. N. Guo, and H. Y. Shum. Asymptotically admissible texture synthesis. *Proc. Intl. Workshop on Stat. and Comp. Theories of Vision*, 2001.

[58] J.-F. Aujol and A. Chambolle. Dual norms and image decomposition models. *Int. J. Comp. Vision*, 63(1):85–104, June 2005.

[59] A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. *Proc. IEEE Int. Conf. Comp. Vision*, pages 1033–1038, 1999.

[60] L.-Y. Wei and M. Levoy. Order-independent texture synthesis. *TR 2002-01*, 2002.

[61] J. Jia and C. K. Tang. Image repairing: Robust image synthesis by adaptive nd tensor voting. *Proc. IEEE Comp. Vision and Pattern Rec.*, 1:643–835, 2003.

[62] H. Jin, S. Soatto, and A. J. Yezzi. Multi-view stereo beyond lambert. *Proc. IEEE Comp. Vision and Pattern Rec.*, 2003.

[63] O. Faugeras and R. Keriven. Variational principles, surface evolution, pde's, level set methods, and the stereo problem. *IEEE Trans. Image Processing*, 7(3):336–344, March 1998.

[64] Y. Duan, Liu Yang, H. Qin, and D. Samaras. Shape reconstruction from 3d and 2d data using pde-based deformable surfaces. *Euro. Conf. Comp. Vision*, 2004.

[65] F. Han and S.-C. Zhu. Bayesian reconstruction of 3d shapes and scenes from a single image. *Proc. IEEE Int. Conf. Comp. Vision*, 2003.

[66] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision, Second Edition.* Cambridge University Press, 2003.

[67] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision.* Prentice Hall, 1998.

[68] P. Favaro, A. Duci, Y. Ma, and S. Soatto. On exploring occlusions in multiple view geometry. *Proc. IEEE Int. Conf. Comp. Vision*, October 2003.

[69] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. *Proc. IEEE Comp. Vision and Pattern Rec.*, 4:921–924, Aug 2004.

[70] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *Proc. Euro. Conf. Comp. Vision*, III:65–81, May 2002.

[71] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. *Proc. IEEE Comp. Vision and Pattern Rec.*, 2005.

[72] B. Lichtenbelt, R. Crane, and S. Nagvi. *Introduction to Volume Rendering.* (Hewlett-Packard Professional Books). Prentice Hall, 1998.

[73] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics (SIGGRAPH 1987)*, pages 163–169, July 1987.

[74] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice in C, second edition.* Addison-Wesley Professional, 1995.

[75] J.E. Solem, F. Kahl, and A. Heyden. Visibility constrained surface evolution. *Proc. IEEE Comp. Vision and Pattern Rec.*, 2005.

[76] M. M. Oliveira and G. Bishop. Image-based objects. *Proc. ACM Symposium Interactive 3D Graphics*, April 1999.

[77] E. Boyer and J.-S. Franco. A hybrid approach for computing visual hulls of complex objects. *Proc. IEEE Comp. Vision and Pattern Rec.*, 1:695–701, 2003.