International Journal of Computer Vision 63(2), 141–151, 2005

Image-Based Rendering Using Image-Based Priors

ANDREW FITZGIBBON
Engineering Science, The University of Oxford, UK
awf@robots.ox.ac.uk

YONATAN WEXLER
Computer Science and Applied Math, The Weizmann Institute of Science, Israel
yonatan.wexler@weizmann.ac.il

ANDREW ZISSERMAN
Engineering Science, The University of Oxford, UK
az@robots.ox.ac.uk

Received March 7, 2003; Accepted June 23, 2003

First online version published in February, 2005

Abstract. Given a set of images acquired from known viewpoints, we describe a method for synthesizing the image which would be seen from a new viewpoint. In contrast to existing techniques, which explicitly reconstruct the 3D geometry of the scene, we transform the problem to the reconstruction of colour rather than depth. This retains the benefits of geometric constraints, but projects out the ambiguities in depth estimation which occur in textureless regions.

On the other hand, regularization is still needed in order to generate high-quality images. The paper's second contribution is to constrain the generated views to lie in the space of images whose texture statistics are those of the input images. This amounts to an *image-based* prior on the reconstruction which regularizes the solution, yielding realistic synthetic views. Examples are given of new view generation for cameras interpolated between the acquisition viewpoints—which enables synthetic steadicam stabilization of a sequence with a high level of realism.

Keywords: new view synthesis, image-based rendering

1. Introduction

Given a small number of photographs of the same scene from several viewing positions, we want to synthesize the image which would be seen from a new viewpoint. This "view synthesis" (Fig. 1) problem has been widely researched in recent years. However, even the best methods do not yet produce images which look truly real. The primary source of error is in the trade-off between the inherent ambiguity of the problem, and the loss of high-frequency detail due to the regularizations which must be applied to alleviate that ambiguity. In this paper, we show how to constrain the generated images to have the same local statistics as natural images, effectively projecting the new view onto the space of real-world images. As this space is a small subspace of the space of all images, the result is strongly regularized synthetic views which preserve high-frequency details.

Strategies for view synthesis are divided into those which explicitly compute a 3D representation of the scene, and those in which the computation of scene geometry is implicit. The first class includes texture-mapped rendering of stereo reconstructions (Koch,

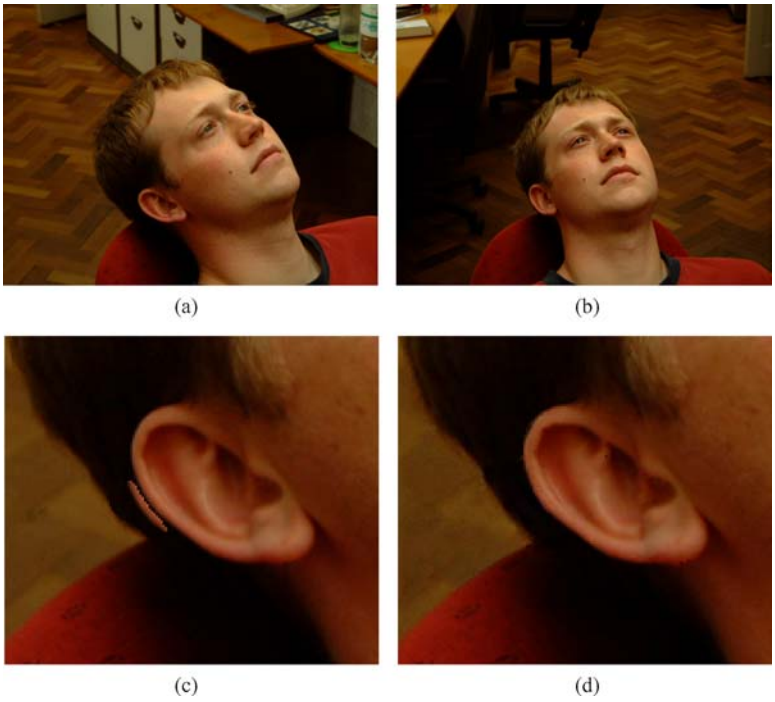

Figure 1. View synthesis. (a, b): Two from a set of 39 images taken by a hand-held camera. (c): Detail from a new view generated using state-of-the-art view synthesis. The new view is about 20° displaced from the closest view in the original sequence. Note the spurious echo of the ear. (d): The same detail, but constrained to only generate views which have similar local statistics to the input images.

1995; Scharstein, 1999; Scharstein and Szeliski, 2002), volumetric techniques such as space carving (Broadhurst and Cipolla, 2001; Kutulakos and Seitz, 1999; Matusik et al., 2000; Seitz and Dyer, 1997; Wexler and Chellappa, 2001), and other volumetric approaches (Szeliski and Golland, 1998). Implicit-geometry techniques (Gortler et al., 1996; Levoy and Hanrahan, 1996; Matusik et al., 2002; McMillan and Bishop, 1995) assemble the pixels of the synthesized view from the rays sampled by the pixels of the input images. In a newly emergent class of technique, to which this paper is most closely related, view-dependent geometry (Debevec et al., 1996; Irani et al., 2000; Koch et al., 2001; Rademacher, 1999) is used to guide the selection of the colour at each pixel.

What all these techniques have in common, whether based on lightfields or explicit 3D models, is that there is no free lunch: in order to generate a new ray which is not in the bundle one is given, one must solve a form of the stereo correspondence problem. This is a difficult inverse problem, which is poorly conditioned: for a given set of images, many different solutions will model the image data equally well. Thus, in order to select between the nearly equivalent solutions the problem must be regularized by incorporating prior knowledge about the likely form of the solution. Previous work on new-view synthesis or stereo reconstruction has typically included such prior knowledge as *a priori* constraints on the (piecewise) smoothness of the 3D geometry, which results in artifacts at depth boundaries. In this paper, because the problem is expressed in terms of the reconstructed *image* rather than the reconstructed depth map, we can impose image-based priors, which can be learnt from natural images (Freeman and Pasztor, 1999; Grenander and Srivastava, 2001; Huang and Mumford, 1999; Srivastava et al., 2003).

The most relevant previous work is primarily in two areas: view-dependent geometry, and natural image statistics. Irani et al. (2002) expressed new view generation as the estimation of the colour at each generated pixel. Their representation implies, as does ours, a 3D geometry for the scene which is different for each synthetic viewpoint, and is thus related to view-dependent visual hull computation (Matusik et al., 2000; Wexler and Chellappa, 2001). As they note, this greatly improves the fidelity of the reconstructed image. However, it does not remove the fundamental ambiguity in the problem, which this paper directly addresses. In

addition, their technique depends on the presence of a dominant plane in the scene, where this paper deals with the case of a general 3D scene with general camera motion.

The use of image-based priors to regularize hard inverse problems is inspired by Freeman and Pasztor's (1999) work on learning priors for Bayesian image reconstruction. Our texture representation, as a library of exemplar image patches, derives from this and from the recent tecture synthesis literature (Efros and Leung, 1999; Wei and Levoy, 2000). In this paper we extend these ideas to deal with the strongly multimodal data likelihoods present in the image-based rendering task, allowing the generation of new views which are locally similar to the input images, but globally consistent with the new viewpoint.

2. Problem Statement

We are given a collection of n 2D images $\mathcal{I}_1$ to $\mathcal{I}_n$, in which $I_i(x, y)$ is the colour at pixel (x, y) of the ith image.[1] Colour is expressed as a 3- vector in an appropriate colorspace. The images are taken by cameras in different positions represented by 3×4 projection matrices $\mathtt{P}_1$ to $\mathtt{P}_n$, which are supplied. Figure 2 summarizes the situation. The projection matrix $\mathtt{P}$ projects homogeneous 3D points $\boldsymbol{X}$ to homogeneous 2D points $\mathbf{x} = \lambda(x, y, 1)^\top$ linearly: $\mathbf{x} = \mathtt{P}\boldsymbol{X}$ where the equality is up to scale. We denote by $I_i(\boldsymbol{X})$ the pixel in image i to which 3D point $\boldsymbol{X}$ projects, so

$$I_i(\boldsymbol{X}) = I_i(\pi(\mathtt{P}_i \boldsymbol{X})), \quad \pi(x, y, w) = (x/w, y/w) \tag{1}$$

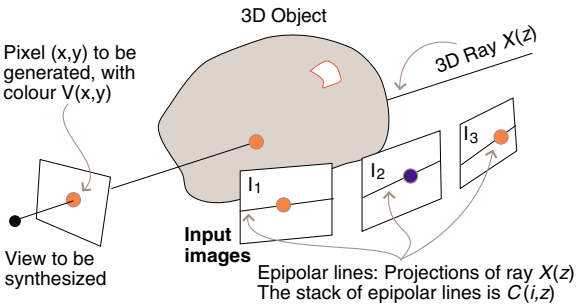

Figure 2. Geometric configuration. The supplied information is a set of 2D images $I_{1..n}$ and their camera positions $\mathtt{P}_{1..n}$. At each pixel in the view to be synthesized, we wish to discover the colour which is most likely to be a reprojection of a 3D object point, based on the implied projection into the source images.

The task of virtual view synthesis is to generate the image which would be seen by a virtual camera in a position not in the original set. Specifically, we wish to compute, for each pixel $V(x, y)$ in a virtual image $\mathcal{V}$ the colour which that pixel would observe if a real camera were placed at the new location. We assume we are dealing with diffuse, opaque objects, and that any deviations from this assumption may be considered part of imaging noise. The extensions to more general lighting assumptions are exactly those in space carving (Kutulakos and Seitz, 1999), and will not be dealt with here.

The objective of this work is to infer the most likely rendered view $\mathcal{V}$ given the set of input images $\mathcal{I}_1, \ldots, \mathcal{I}_n$. In a Bayesian framework, we wish to choose the synthesised view $\mathcal{V}$ which maximizes the posterior $p(\mathcal{V} \mid \mathcal{I}_1, \ldots, \mathcal{I}_n)$. Bayes' rule allows us to write this as

$$p(\mathcal{V} \mid \mathcal{I}_1, \ldots, \mathcal{I}_n) = \frac{p(\mathcal{I}_1, \ldots, \mathcal{I}_n \mid \mathcal{V}) p(\mathcal{V})}{p(\mathcal{I}_1, \ldots, \mathcal{I}_n)} \tag{2}$$

where $p(\mathcal{V})$ is the prior on $\mathcal{V}$, and the data term $p(\mathcal{I}_1, \ldots, \mathcal{I}_n \mid \mathcal{V})$ measures the likelihood that the observed images could have been observed if $\mathcal{V}$ were the true colours at the novel viewpoint. Because we shall maximize this posterior over $\mathcal{V}$, we need not compute the denominator $p(\mathcal{I}_1, \ldots, \mathcal{I}_n)$, and will instead optimize the function

$$q(\mathcal{V}) = p(\mathcal{I}_1, \ldots, \mathcal{I}_n \mid \mathcal{V}) p(\mathcal{V}) \tag{3}$$

This likelihood has two parts: the photoconsistency likelihood $p(\mathcal{I}_1, \ldots, \mathcal{I}_n \mid \mathcal{V})$ and the prior $p(\mathcal{V})$ which we shall call $p_{\text{texture}}(\mathcal{V})$.

2.1. *Photoconsistency Constraint*

The colour consistency constraint we employ is standard in the stereo and space-carving literature. We consider each pixel $V(x, y)$ in the synthesised view separately, so the likelihood is written as the product of per-pixel likelihoods

$$p(\mathcal{I}_1, \ldots, \mathcal{I}_n \mid \mathcal{V}) = \prod_{(x,y)} p(\mathcal{I}_1, .., \mathcal{I}_n \mid V(x, y)) \tag{4}$$

Consider the generation of new-view pixel $V(x, y)$. This is a sample from along the ray emanating from the camera centre, which we may assume to be the origin. Let the direction of this ray be denoted $\mathbf{d}(x, y)$. It

can be computed easily given the calibration parameters of the virtual camera. Let a 3D point along the ray be given by the function $\boldsymbol{X}(z) = z\mathbf{d}(x, y)$ where z ranges between preset values $z_{\min}$ and $z_{\max}$. For a given depth z, we can compute using (1) the set of pixels to which $\boldsymbol{X}(z)$ projects in the images $\mathcal{I}_{1..n}$. Denote the colours of those pixels by the function

$$C(i, z) = I_i(\boldsymbol{X}(z)). \tag{5}$$

Let the set of all colours at a given z value be written

$$C(:, z) = \{C(i, z)\}_{i=1}^{n}, \tag{6}$$

and the set, $\mathbf{C}$, of all samples—at location (x, y)—be

$$\mathbf{C} = \{C(i, z) \mid 1 \leq i \leq n, z_{\min} < z < z_{\max}\}. \tag{7}$$

Figure 3 shows an example of $\mathbf{C}$ at one pixel in a real sequence. Because the input-image pixels whose colours form $\mathbf{C}$ are the only pixels which influence new-view pixel (x, y), the photoconsistency likelihood further simplifies to (writing V for $V(x, y)$)

$$p(\mathcal{I}_1, \ldots, \mathcal{I}_n \mid V) = p(\mathbf{C} \mid V) \tag{8}$$

Now, by making explicit the dependence on the depth z and marginalizing (assuming $p(z \mid V)$ is uniform), we obtain

$$\begin{aligned} p(\mathbf{C} \mid V) &= \int p(\mathbf{C} \mid V, z)\, dz \\ &= \int p(C(:, z) \mid V, z)\, dz \end{aligned} \tag{9}$$

The noise on the input image colours $C(i, z)$ will be modelled as being drawn from distributions with density functions of the form $\exp(-\beta\rho(t))$, centred at V, where β is a constant specifying the width of the distribution. Thus the likelihood is of the form

$$p(C(:, z) \mid V, z) = \prod_{i=1}^{n} \exp -\beta\rho(\|V - C(i, z)\|) \tag{10}$$

The function ρ is a robust kernel, and in this work is generally the absolute distance $\rho(x) = |x|$, corresponding to an exponential distribution on the pixel intensities. In situations (discussed later) where a Gaussian distribution is more appropriate, the kernel becomes $\rho(x) = x^2$.

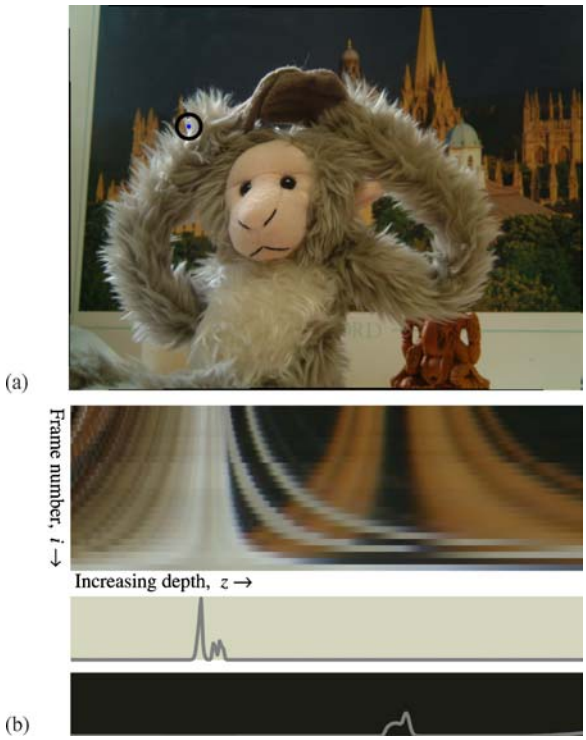

Figure 3. Photoconsistency. One image is shown from a sequence of 27 captured by a hand-held camera. The circled pixel x's photoconsistency with respect to the other 26 images is illustrated in (a). The upper image in (b) shows the reprojected colours $C(:, z)$ as columns of 26 colour samples, at each of 500 depth samples. The colours are the samples $C(i, z)$ where the frame number i varies along the vertical axis, and the depth samples z vary along the horizontal. Equivalently, row i of this image is the intensity along the epipolar line generated by x in image i. Below are shown photoconsistency likelihoods $p(\boldsymbol{C} \mid V, z)$ for two values of the colour V (backgrounds to the plots). As this pixel is a co-location of background and foreground, these two colours form modes of $p(\boldsymbol{C} \mid V)$ when z is maximized. This multi-modality is the essence of the ambiguity in new-view synthesis, which prior knowledge must remove.

In order to choose the colour V, we shall be computing (Section 3.1) the modes of the function $p(C(:, z) \mid V(x, y))$. As defined above, this requires the computation of the integral (9), which is computationally undemanding. However, because the value of β is difficult to know, and because the function is sensitive to its value, the integral must also be over a hyperprior on β, rendering it much more challenging. Approximating the marginal by the maximum gives us an approximation, denoted p_{photo},

$$p_{\text{photo}}(V(x, y)) \approx \max_{z} p(C(:, z) \mid V, z) \tag{11}$$

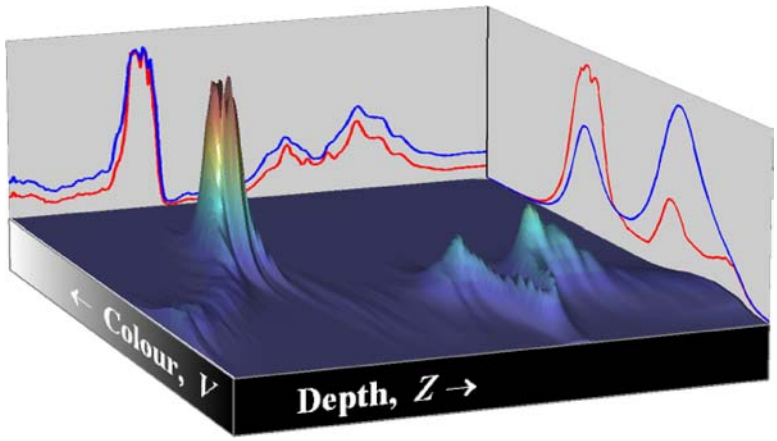

Figure 4. The function $p(C(:, z) \mid V, z)$ plotted for the pixel studied in Fig. 3, with grayscale images, so V is a scalar, and $\rho(x) = |x|$. The projected graphs show the marginals (blue) and the maxima (red). The marginalization over colour (V) has fewer minima than that over z, and the two modes corresponding to foreground and background are clearly seen.

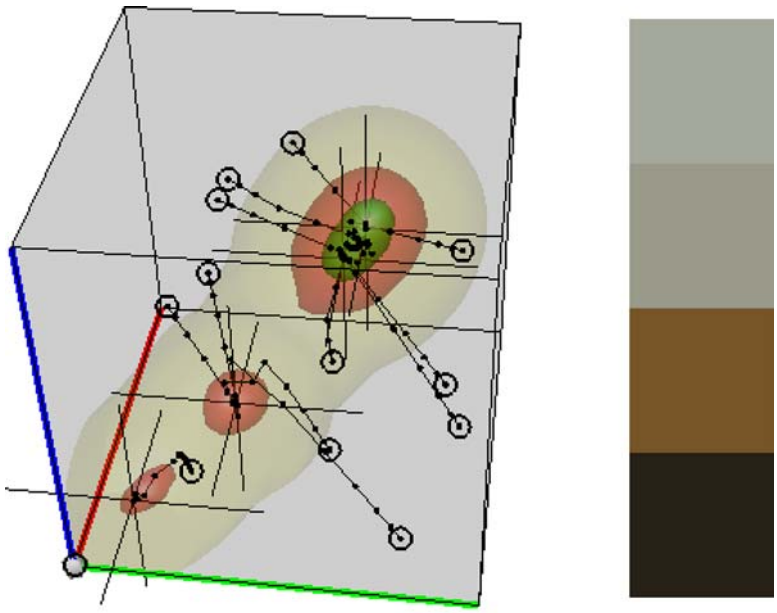

Figure 5. *Minima of* E_{photo}. (a) Isosurfaces in RGB space of the photoconsistency function $E_{\text{photo}}(V)$ at the pixel studied in Fig. 3. Minima are computed by gradient descent from random starting positions, of which twelve are shown (black circles), with the gradient descent trajectories plotted in black. Four modes were retained after clustering; their locations are marked by white 3D "axes" lines in (a), and their RGB colours are shown in (b).

which avoids both of these problems. In the implementation, the maximum over z is computed by explicitly sampling, typically using 500 values. Figure 4 shows a plot of $p(C(:, z) \mid V, z)$ for grayscale C at a typical pixel. Figure 5 shows isosurface plots of $p_{\text{photo}}(V)$ in RGB space for the same pixel.

2.2. *Incorporating the Texture Prior*

The function $p_{\text{photo}}(V)$ will generally be multimodal, due firstly to physical factors such as occlusion and partial pixel effects and secondly to deficiencies in the image-formation model, such as not modelling

specular reflections or having an inaccurate model of imaging noise. Thus the data likelihood at the true colour may often be lower than the likelihood at other, spurious values. Consequently, selecting the maximum-likelihood V at each pixel yields images with significant artefacts, such as those shown in Fig. 1(c). We would like to constrain the generated views to lie in the space of real images by imposing a prior on the possible generated images. Defining such a prior is in the domain of the analysis of natural image statistics, an active area of recent neurophysiological and machine learning research (Grenander and Srivastava, 2001; Huang and Mumford, 1999; Srivastava et al., 2003). Because it has been observed that correlation between pixels falls off quickly as a function of distance, we can make the assumption that the probability density can be written as a product of functions operating on small neighborhoods. Let the generated image $\mathcal{V}$ have pixels $V(x, y)$. Then the prior has the form

$$p_{\text{texture}}(\mathcal{V}) = \prod_{x,y} p_{\text{texture}}(\mathcal{N}(x, y)) \tag{12}$$

where the function $\mathcal{N}(x, y)$ is the set of colours of neighbours of (x, y). Here we use 5×5 neighbourhoods, so

$$\mathcal{N}(x, y) = \{V(x + i, y + j) \mid -2 \leq i, j \leq 2\}. \tag{13}$$

As the form of p_{texture} is typically very difficult to represent analytically (Huang and Mumford, 1999), we follow (Efros and Leung, 1999; Freeman and Pasztor, 1999) and represent our texture prior as a library of texture patches. The likelihood of a particular neighbourhood is measured by computing its distance to the closest database patch. Thus, we are given a texture database of 5×5 image patches, denoted $\mathbb{T} = \{\mathcal{T}_1, ..., \mathcal{T}_N\}$ where N is typically extremely large. The definition of p_{texture} is then

$$p_{\text{texture}}(\mathcal{N}(x, y)) = \exp\left(-\lambda \min_{\mathcal{T} \in \mathbb{T}} \|\mathcal{T} - \mathcal{N}(x, y)\|^2\right)$$

where λ is a tuning parameter. This is a closest-point problem in the set of 75-d points ($75 = 5 \times 5 \times 3$) in $\mathbb{T}$ and may be efficiently solved using a variety of algorithms, for example vector quantization and BSP tree indexing (Wei and Levoy, 2000).

2.3. Combining Photoconsistency and Texture

Finally, combining the data and prior terms, we have the expression for the quasi-likelihood

$$q(\mathcal{V}) = \prod_{x,y} p_{\text{photo}}(V(x, y))\, p_{\text{texture}}(\mathcal{N}(x, y)).$$

In the implementation, we minimize the negative log of q, yielding the energy formulation

$$E(\mathcal{V}) = \sum_{x,y} E_{\text{photo}}(V(x, y)) + \sum_{x,y} E_{\text{texture}}(\mathcal{N}(x, y)) \tag{14}$$

where E_{photo} measures the deviation from photoconsistency at pixel (x, y) and E_{texture} measures the a-priori likelihood of the texture patch surrounding (x, y). From (11), the definition of E_{photo} at a pixel (x, y) with 3D ray $\boldsymbol{X}(z)$ is

$$E_{\text{photo}}(V) = \min_{z_{\min} < z < z_{\max}} \sum_{i=1}^{n} \rho(\|V - I_i(\boldsymbol{X}(z))\|) \tag{15}$$

The texture energy is the negative log of p_{texture}, giving

$$E_{\text{texture}}(\mathcal{N}(x, y)) = \lambda \min_{\mathcal{T} \in \mathbb{T}} \|\mathcal{T} - \mathcal{N}(x, y)\|^2 \tag{16}$$

The view synthesis problem is now one of minimization of E over the space of images. This is a difficult global optimization problem, and making it tractable is the subject of the next section.

3. Implementation

The optimization of the energy defined above could be directly attempted using a global optimization strategy such as simulated annealing. However, both the prior and the data term E_{photo} are expensive to evaluate, with multiple local minima at each pixel, meaning that attaining a global optimum will be difficult, and certainly time consuming. To render the optimization tractable, we exploit the simplification of the energy function conferred by estimating colour rather than depth. That is, we compute the set of modes of the photo-consistency term for each pixel, and restrict the solution for that pixel to this set. Then the texture prior is used to select the values from this set. This reduces the problem from a search over a high-dimensional space to an enumeration of the possible combinations.

Although the data likelihood $p(\mathbf{C} \mid V)$ is multimodal, there are typically far fewer modes than there are maxima of $p(C(:, z) \mid V, z)$ over depth, so we can hope to explicitly compute the modes of $p(\mathbf{C} \mid V)$ as the first step. This means that the optimization becomes a discrete labelling problem, which although still complex, can be analysed much more efficiently.

3.1. Enumerating the Minima of $E_{photo}(V)$

The goal then is to generate a list of plausible colours for each rendered pixel $V(x, y)$. One option would be to sample from $p_{\text{photo}}(V)$ using MCMC, but this is computationally unattractive. A more practical alternative is to find *all local minima* of the energy function $E_{\text{photo}}(V)$. On the face of it, this seems a tall order, but as Fig. 5 indicates, there are typically few minima in a generally well-behaved space. Inspection of several such plots on a number of scenes suggests that this behaviour is typical. Finding all local minima of such functions is task for which several strategies have emerged from the computational chemistry community, and have been introduced to computer vision by Sminchisescu and Triggs (2002). The most expensive is to densely sample the space of V (here 3D RGB space), and this is the strategy used to obtain the isosurface plot shown in Fig. 5. A more efficient strategy to isolate the minima is to start gradient descent from several randomly chosen starting points, and iterate until local minima are found. Finally clustering on the locations of the minima produces a set of distinct colours which are likely at that pixel. On the images we have tested, 12 steps of gradient descent on each of 20 random starting colours V takes a total of about 0.1 seconds in Matlab, and produces between four and six colour hypotheses at each pixel.

3.2. Texture Reference and Rectification

The second implementation issue is the source of reference textures. To build a general tool for projection of images onto natural images, a large database of images of natural scenes would be the ideal choice. In this case, however, we are operating in a limited problem domain. We expect that the newly synthesized views will be similar *locally* to the input views with which the algorithm is provided. Therefore, the texture library is built of patches from the input images. This provides excellent performance with a small library, and the photoconsistency term means that the system cannot "overlearn" by simply copying large patches from the nearest source image to the newly rendered view. For speed, we can also use the known z range to limit the search for matching texture windows in source image $\mathcal{I}_i$ to the bounding box of $\{\mathrm{P}_i \boldsymbol{X}(z) \mid z_{\min} < z < z_{\max}\}$.

3.3. Optimization

Given the modes of the photoconsistency distribution at each pixel, the optimization of (14) becomes a labelling problem. Each pixel is associated with an integer label $l(x, y)$, which indicates which mode of the distribution will be used to colour that pixel, with a corresponding photoconsistency cost which is precomputed. This significantly reduces the cost of function evaluations, but the optimization is still a computationally challenging problem. For this work, we have implemented a variant of the iterated conditional modes (ICM) algorithm (Besag, 1986), alternately optimizing the photoconsistency and texture priors. The algorithm begins by selecting, for each pixel, the most likely mode of the photoconsistency function, yielding an initial estimate V^0. Then, at each ICM iteration, each pixel is varied until the 5×5 window surrounding it minimizes the sum $E_{\text{photo}} + E_{\text{texture}}$ at that pixel. This optimization is potentially extremely expensive, implying the evaluation of $E_{\text{photo}}(V)$ for the value V in the centre of each texture patch T. However, because the minima of E_{photo} are available, a fast approximation is obtained simply by writing $E_{\text{photo}}(V) \approx \|V - V^{r-1}\|^2$, where V^{r-1} is the colour obtained at the previous iteration. If all other pixels in $\mathcal{V}$ are fixed, the task is to choose V to minimize $E_{\text{photo}} + \lambda E_{\text{texture}}$, approximated by (using (16))

$$V^r = \operatorname*{argmin}_V \min_{\mathcal{T} \in \mathbb{T}} (\|V - V^{r-1}\|^2 + \lambda \|\mathcal{T} - \mathcal{N}(V)\|^2) \tag{17}$$

where $\mathcal{N}(V)$ is the image neighbourhood around V. Splitting the second term into a contribution from the centre pixel of $\mathcal{T}$ and the remainder of the neighbourhood, it can be shown that this amounts to setting the centre pixel to a linear combination of (a) the photoconsistency mode, and (b) the value that would be predicted by sampling-based texture synthesis. If V^{r-1} is the value predicted by photoconsistency at the previous iteration, and T is the value at the centre pixel of the best matching texture patch $\mathcal{T}$, then the pixel should be

Input: Images I_1 to I_n,
Camera positions $\mathtt{P}_1$ to $\mathtt{P}_n$
Texture library $\mathbb{T} \subset \mathbb{R}^{75}$
Output: New view $\mathcal{V}$, from camera at origin $\mathtt{P} = [\mathtt{I}, \mathbf{0}]$.

Preprocessing:
for each pixel (x, y)
Compute ray direction $\boldsymbol{d}(x, y)$, e.g. $\boldsymbol{d} = (x, y, 1, 0)^\top$.
Choose m depths to sample e.g. $\{z_j = z_{\min} + j\Delta z\}_{j=1}^{m}$
Compute $n \times m \times 3$ array of pixel colours
$C_{ij} = I_i\left(\mathtt{P}_i * z_j \boldsymbol{d}(x, y)\right)$
Compute K local minima, denoted $V_{1..K}(x, y)$, of
$E_{\text{photo}}(V) = \min_j \sum_i \rho(\|C_{ij} - V\|)$
Sort so that $E_{\text{photo}}(V_k) < E_{\text{photo}}(V_{k+1}) \forall k$
Set initial estimate of new view $V^0(x, y) = V_1(x, y)$
end
Update at iteration r:
for each pixel (x, y)
Extract window
$\mathcal{N} = \{V^{r-1}(x + i, y + j) | -2 \le i, j \le 2\}$.
Find closest texture patch
$\mathcal{T} = \operatorname{argmin}_{\mathcal{T} \in \mathbb{T}} \|\mathtt{M}(\mathcal{N} - \mathcal{T})\|^2$
$\mathtt{M}$ is a mask which ignores the centre pixel.
Set $V^r(x, y)$ to the mode $V_k(x, y)$ nearest the value computed by (18).
end

Figure 6. Pseudocode for iterative computation of new view $\mathcal{V}$. The preprocessing is expensive (about 0.1 sec/pixel), the iterations cost as much as patch-based texture synthesis.

replaced by

$$V^r = \frac{V^{r-1} + \lambda T}{1 + \lambda} \tag{18}$$

Finally, replacing V^r by the closest mode at each iteration ensures that the synthesized colour is always a subset of the photoconsistency minima. Note that this does not undo the good work of the robust kernel in computing the modes of E_{photo}, but allows the texture prior to efficiently select between the robustly-computed colour candidates at each pixel. This also prevents the algorithm from copying large sections of the texture source. Figure 6 summarizes the steps in the algorithm.

3.4. Choice of Robust Kernels

In the preceding, the choice of robust kernels for the photoconsistency likelihood has been mentioned several times. In practice, there is a significant tradeoff between speed and accuracy implied by choosing other than the squared-error kernel $\rho(x) = x^2$ kernel, as the mode computation can be significantly optimized for the squared-error case. The problem arises when there is significant occlusion in the sequence, as on the example pixel in Fig. 3, and it becomes necessary to produce a view which looks "behind" the foreground pixel. Using the squared-error kernel, the true colour (in this case, black) is not a minimum of E_{photo}, because the column $C(:, z)$ at the depth corresponding to the background contains some white pixels which are significant outliers to the Gaussian distribution $\exp(-\rho(\cdot))$. The true colour *is* a minimum using the absolute distance $\rho(x) = |x|$ or Huber kernels, which are less sensitive to such outliers. To provide a rule of thumb, the squared-error kernel is fast, and works well for interpolation, but the absolute distance kernel is needed for extrapolation.

4. Examples

Image sequences were captured using a hand-held camera, and the sequences were calibrated using

Figure 7. Leave-one-out test. Using 26 views to render a missing view allows comparison to be made between the rendered view and ground truth. (a) Maximum-likelihood view, in which each pixel is coloured according to the highest mode of the photoconsistency function. High-frequency artifacts are visible throughout the scene. (b) View synthesized using texture prior. The artifacts are significantly reduced. (c) Ground-truth view. (d) Difference image between (b) and (c).

Figure 8. Steadicam test. Three novel views of the monkey scene from viewpoints not in the original sequence. The complete sequence may be found at `http://www.robots.ox.ac.uk/~awf/ibr`.

Figure 9. 3D composite from 2D images. The camera motion from the live-action background plate is applied to the head sequence, rendering new views of the face.

commercially available camera tracking software (2d3 Ltd. http://www.2d3.com, 2002). A number of examples of the algorithm performance were produced. Single still frames are reproduced here, and complete MPEG sequences may be found at `http://www.robots.ox.ac.uk/~awf/ibr`.

The first experiment is a leave-one-out test, so that the recovered images can be compared against ground truth. Each frame of the 27-frame "monkey" sequence was reconstructed based on the other 26 frames. Figure 7 shows the results for a typical frame, comparing the ground truth image first to the synthesized view using photoconsistency alone, and then to the result guided by the texture prior. Visually, the fidelity is high, and the image is free of the high-frequency artifacts which the photoconsistency-maximizing view exhibits. Artifacts do occur in the background visible under the monkey's arm, where few of the source views have observed the background, meaning it does not appear as a mode of the photoconsistency distribution. The difference image in Fig. 7(d) is simply the length of the RGB difference vector at each pixel, but shows that the texture prior does not bias the generated view, for example by copying one of the texture sources.

The second example shows performance on a "steadicam" task, where the scene is re-rendered at a

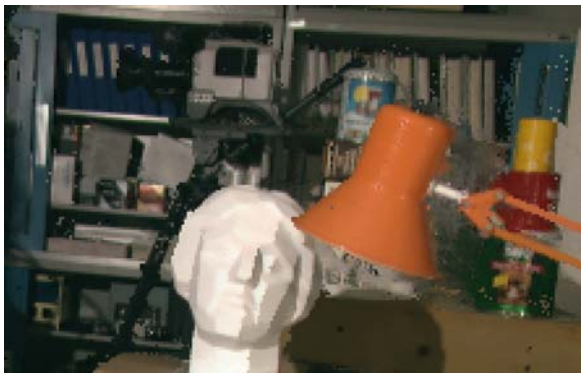

Figure 10. *Tsukuba*. Fine details such as the lamp arm are retained, but some ghosting is evident around the top of the lamp.

set of viewpoints which smoothly interpolate the first and last camera position and orientation. The reader is encouraged to consult the videos on the webpage above to confirm the absence of artifacts, and the subtle movements of the partial occlusions at the boundaries.

Figure 1 shows example images from one sequence and illustrates the improvement obtained. The erroneous areas surrounding the ear are removed, while the remainder of the image retains its (correct) solution. At high magnification, it is in fact possible to see that the optimized solution has added back some high-frequency detail in the image. This is because the local statistics of the texture library are being applied to the rendered view.

5. Conclusion

This paper has shown that view synthesis problems can be regularized using texture priors. This is in contrast to the depth-based priors that previous algorithms have used. Image-based priors have several advantages over the depth-based ones. First, depth priors are difficult to learn from real images, so artificial approximations are used. These approximations are equivalent to assuming very simple models of the world—for example, that it is piecewise planar—and thus introduce artifacts into the generated views. In contrast, image-based priors are easy to obtain from the world. If the problem domain is restricted, as it is here, a small number of images can be used to regularize the solution to a complex inverse problem.

There are many areas for further work: (1) image-based priors as implemented here are expensive to evaluate. For a typical depth prior, evaluation of the prior in a pixel neighbourhood requires computation of the order of a few machine instructions. As image-based priors are stored in large lookup tables, the cost of evaluating them is many times higher. (2) In this paper, only one optimization strategy was investigated. It is hoped that examination of other strategies will lead to significantly quicker solutions. (3) Occlusion is handled here by the robust kernel ρ. More geometric handling of occlusion, analogous to space carving's improvement over voxel colouring, ought to yield better results. (4) When rendering sequences of images, it is valuable to impose temporal continuity from frame to frame. This paper has not addressed this issue, so the rendered sequences show some flicker. On the other hand this does allow the stability of the per-frame solutions to be evaluated.

Acknowledgments

Funding for this work was provided by the DTI/EPSRC Link grant V2I. Fitzgibbon would like to thank the Royal Society for its generous support.

Note

1. Notation guide: calligraphic letters $\mathcal{L}$ are images or windows from images. Uppercase roman letters L are RGB (or other colourspace) vectors. Bold roman lowercase $\boldsymbol{x}$ denotes 2D points, also written (x, y), and bold roman uppercase are 3D points $\boldsymbol{X}$. Matrices are in fixed-width font, viz $\mathtt{M}$.

References

2d3 Ltd. http://www.2d3.com, 2002.

Besag, J. 1986. On the statistical analysis of dirty pictures. *J. Royal Stat. Soc. B*, 48(3):259–302.

Broadhurst, A. and Cipolla, R. 2001. A statistical consistency check for the space carving algorithm. In Proc. *ICCV*.

Debevec, P.E., Taylor, C.J., and Malik, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proceedings, ACM SIGGRAPH,* pp. 11–20.

Efros, A. and Leung, T. 1999. Texture synthesis by non-parametric sampling. In *Proc. ICCV*, pp. 1039–1046.

Freeman, W. and Pasztor, E. 1999. Learning low-level vision. In *ICCV*, pp. 1182–1189.

Gortler, S.J., Grzeszczuk, R., Szeliski, R., and Cohen, M.F. 1996. The lumigraph. In *SIG-GRAPH96*.

Grenander, U. and Srivastava, A. 2001. Probability models for clutter in natural images. *IEEE PAMI*, 23(4):424–429.

Huang, J. and Mumford, D. 1999. Statistics of natural images and models. In *Proc. CVPR*, pp. 1541–1547.

Irani, M., Hassner, T., and Anandan, P. 2002. What does the scene look like from a scene point? In *Proc. ECCV*.

Koch, R. 1995. 3D surface reconstruction from stereoscopic image sequences. In *Proc. ICCV*, pp. 109–114.

Koch, R., Heigl, B., and Pollefeys, Marc 2001. Image-based rendering from uncalibrated light-fields with scalable geometry. In G. Gimel'farb (Eds.) R. Klette, T. Huang, Multi-Image Analysis, Springer LNCS 2032, pp. 51–66.

Kutulakos, K. and Seitz, S. 1999. A theory of shape by space carving. In *Proc. ICCV*, pp. 307–314.

Levoy, M. and Hanrahan, P. 1996. Light field rendering. In *SIGGRAPH96*.

Matusik, W., Buehler, C., Raskar, R., McMillan, L., and Gortler, S. 2000. Image-based visual hulls. In *Proc. ACMSIGGRAPH*, pp. 369–374.

Matusik, W., Pfister, H., Beardsley, P.A., Ngan, A., Ziegler, R., and McMillan, L. 2002. Image-based 3d photography using opacity hulls. In *Proc. ACMSIGGRAPH*.

McMillan, L. and Bishop, G. 1995. Plenoptic modeling: An image-based rendering system. In *SIGGRAPH95*.

Rademacher, P. 1999. View-dependent geometry. In *Proc. ACM. SIGGRAPH* pp. 439–446.

Scharstein, D. 1999. *View Synthesis Using Stereo Vision*, Vol. 1583 *of LNCS.* Springer-Verlag.

Scharstein, D. and Szeliski, R. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(l):7–42.

Seitz, S.M. and Dyer, C.R. 1997. Photorealistic scene reconstruction by voxel coloring. In *Proc. CVPR*, pp. 1067–1073.

Sminchisescu, C. and Triggs, B. 2002. Building roadmaps of local minima of visual models. In *Proc. ECCV*, Vol. 1, pp. 566–582.

Srivastava, A., Lee, A., Simoncelli, E., and Zhu, S. 2003. On advances in statistical modeling of natural images, 18(l):17–33.

Szeliski, R. and Golland, P. 1998. Stereo matching with transparency and matting. In *Proc. ICCV*, pp. 517–524.

Wei, L.-Y. and Levoy, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proc. ACM SIGGRAPH*, pp. 479–488.

Wexler, Y. and Chellappa, R. 2001. View synthesis using convex and visual hulls. In *Proc. BMVC.*