Image-Based Rendering Using Image-Based Priors

Samuel Audet (260184380) saudet@cim.mcgill.ca
Centre for Intelligent Machines
McGill University, Montréal, Québec, Canada

April 10, 2006 (Revised May 2, 2006)

Abstract

For this report, the view synthesis algorithm from the paper of the same title by Fitzgibbon et al. [1] was implemented. In this report, the geometric and probabilistic background of the algorithm, as well as necessary optimizations required to make the problem more tractable, are succinctly detailed. Results are then presented and analyzed. It was found that although the use of the texture prior improves the resulting rendered images, the initial photoconsistent estimate without use of the prior is of very good visual quality. There is still however a lot of room for improvements in terms of computational performance.

1 Introduction

As my course project for ECSE 626 - Statistical Computer Vision, I decided to implement the view synthesis algorithm developed by Fitzgibbon et al. [1]. In the view synthesis problem, we are interested in inferring new plausible images of a static 3D scene that are rendered from an existing set of images taken from the 3D scene from different orientations than the ones of interest. In this report, the algorithm is first briefly described in terms geometric and probabilistic formulations, and then in terms of optimized implementation. Finally, results are presented and analyzed.

2 Summary of the Algorithm

In all view synthesis problems, what we would like to accomplish is to generate new images corresponding to viewpoints (camera positions) that are interpolated from a given set of viewpoints and their corresponding images. The problem is formulated in terms of 3D space, since the world from which the images are taken from is a 3D world. For this reason, the formulation of the problem always has a multiple view geometry component. The authors then used a Bayesian probabilistic approach in order to find the most probable color to every pixel in the new virtual image. In this report, the geometric formulation of the algorithm is described in the first subsection. The Bayesian approach is detailed in the following subsections. Finally, in order to make the algorithm tractable, some simplifications and optimizations were made, and they are the subject of the final subsection of this section.

2.1 The Geometric Formulation

We are given n projection camera matrices $\mathbf{P} = P_1, ..., P_n$ and their associated images planes $\mathbf{I} = I_1, ...I_n$. A projection matrix incorporates all the linear information related to the internal parameters of the camera, its viewpoint and thus its position in space. For a given 3D point $\mathbf{X}$, its projection into the 2D image plane of a camera with matrix $P_{3\times4}$ is given by $\mathbf{x} = P_{3\times4}\mathbf{X}$. Moreover, $P = M_{3\times3}[I| - \mathbf{C}]$ where $\mathbf{C}$ is the position of the camera center in space, which is therefore equal to $\mathbf{C} = -M^{-1}p_4$ where p_4 is the fourth column vector of P. All 3D points in space will intersect with the camera center when projected onto the camera's image plane. The direction vector $\mathbf{D}$ departing from the camera center through a 2D point $\mathbf{x}$ on the image plane is $\mathbf{D} = [[M^{-1}\mathbf{x}]^T, 0]^T$ and is a point at infinity since $D_4 = 0$. Therefore, we can define the following equation in order to find finite 3D points that all project back to $\mathbf{x}$:

$$\mathbf{X}(z) = \mathbf{C} + z\mathbf{D}, \tag{1}$$

where z is a variable indicating "depth". More detailed information on this subject can be found in the book by Hartley and Zisserman [2], in the section entitled *The projective camera*.

2.2 The Bayesian Formulation

Let us define a new virtual image I_V, the new view we would like to infer from the given set of images and projection matrices. Note that the position and orientation of the new viewpoint, defined by a projection matrix P_V, must be computed as an interpolation from the set of real projection matrices. The scene is also assumed to be static so that to which surface a 3D point refers to does not change with time for the whole sequence of images. In this context, what we are interested in is therefore the maximum a posteriori (MAP) estimate of the image I_V given the images $\mathbf{I}$, their corresponding matrices $\mathbf{P}$, and the given new viewpoint P_V:

$$p(I_V|\mathbf{I},\mathbf{P},P_V) = \frac{p(\mathbf{I},\mathbf{P}|I_V,P_V)p(I_V|P_V)}{p(\mathbf{I},\mathbf{P}|P_V)}. \qquad (2)$$

The term $p(\mathbf{I},\mathbf{P}|I_V,P_V)$ is the likelihood of the image I_V. It represents the forward problem where the joint probability of the images $\mathbf{I}$ and matrices $\mathbf{P}$ is defined by given image I_V and matrix P_V. Since the function is defined only in terms of projection matrices and of color intensity values from the images containing no 3D structural information, it was called by the authors the *photoconsistency constraint* since maximizing this likelihood will result in the best match with regards to color intensities, but with no regards for the actual 3D structure of the scene. The usual way to regularize the problem is by imposing a smooth continuous depth map. However, in real world scenes, this continuity does not hold, for example at the edge of an object in front of a background wall.

Therefore, the authors used a different approach to regularization which makes use of textures (pixels and their neighborhoods) as prior. The prior term $p(I_V|P_V)$ in equation 2 defines the probability of texture occurrence. Naturally occurring textures are not random and have a certain structure to them. Making use of this information, common natural textures can thus "boost" the likelihood term, but it can also be penalized for improbable occurrences.

Finally, note that since we are optimizing for I_V, the normalizing denominator does not need to be computed.

2.3 The Likelihood: Photoconsistency

For the photoconsistency constraint, it is assumed that the color at each of the m pixel sites is independent, and so the likelihood can be rewritten as follows:

$$p(\mathbf{I},\mathbf{P}|I_V,P_V) = \frac{\prod_{\mathbf{i=1}}^{m} p(\mathbf{I},\mathbf{P}|I_V(\mathbf{x}_i),P_V)}{p(\mathbf{I},\mathbf{P}|P_V)^{m-1}}. \qquad (3)$$

Again, since the denominator is not a function of I_V, it does not need to be computed.

For a given 2D point $\mathbf{x}$ and matrix P_V, we can generate all 3D points $\mathbf{X}(z)$ that map to $\mathbf{x}$ by equation 1. Assuming that at least one of the viewpoints has a clear view of the same object as imaged by P_V in I_V, and that the projected color is accurate (i.e.: Lambertian surface) then the set of all possible colors for $I_V(\mathbf{x}) = I_V(\mathbf{X}(z))$ at depth z can be reduced to a set of colors from the views $\mathbf{I}$. The function of possible colors is thus defined as:

$$c(i,z) = I_i(P_i\mathbf{X}(z)) \text{ for } 1 \leq i \leq n, \qquad (4)$$

and set $\mathbf{c}$ of colors is defined as:

$$\mathbf{c} = \{c(i,z)|1 \leq i \leq n, z \geq 0\} \qquad (5)$$

Since $\mathbf{c}$ contains all the information from $\mathbf{I}$ and $\mathbf{P}$ that is relevant to $I_V(\mathbf{x})$, we can reformulate the likelihood term for a given pixel as follows:

$$p_{photo}(I_V(\mathbf{x})) = p(\mathbf{I},\mathbf{P}|I_V(\mathbf{x}),P_V) = p(\mathbf{c}|I_V(\mathbf{x}),P_V). \qquad (6)$$

2.4 The Prior: Texture

Although not described as such by the authors, the prior was understood to be defined as a Markov Random Field [3] with conditional probability:

$$p_{texture}(I_V(\mathbf{x})) = p(I_V(\mathbf{x})|P_V, N(I_V,\mathbf{x})), \qquad (7)$$

where $N(I_V,\mathbf{x})$ is a patch, a sub-image, of the 5×5 neighborhood system surrounding $\mathbf{x}$ in I_V.

2.5 A Gibbs Random Field Formulation

As it turns out, both the prior and the likelihood can be defined in terms of Gibbs Random Fields [3], even if the authors did not explicitly state it as such. In the case of the likelihood, the noise was modeled with Gaussian densities, and the Gaussian density is a special case of the Gibbs density. For the prior, since it is defined as a Markov Random Field, it is "automatically" equivalent to a Gibbs Random Field [3]. Assuming the given viewpoints are independent from each other and marginalizing over z, the form of equation 6 (the likelihood) was defined as follows:

$$p_{photo}(I_V(\mathbf{x})) = Z_p^{-1} \int_z \prod_{i=1}^{n} \exp(-\beta\rho(||I_V(\mathbf{x})-c(i,z)||))dz, \qquad (8)$$

and the form of the prior, as follows:

$$p_{texture}(I_V(\mathbf{x})) = Z_t^{-1}\exp(-\lambda \min_{T\in\mathbb{T}} ||T - N(I_V, \mathbf{x})||), \tag{9}$$

where β and λ are parameters actually representing the inverse temperatures of the random fields, and Z_p and Z_t being normalization constants, again do not need to be computed since they do not depend on I_V. $\rho(x) = |x|$ is a robust kernel (more on this in the results section) and $\mathbb{T}$ represents the set of natural textures. The energy values of the fields are therefore:

$$E_{photo}(I_V(\mathbf{x})) = \int_z \sum_{i=1}^{n} -\beta\rho(||I_V(\mathbf{x}) - c(i,z)||)dz \tag{10}$$

and

$$E_{texture}(I_V(\mathbf{x})) = -\lambda \min_{T\in\mathbb{T}} ||T - N(I_V, \mathbf{x})||. \tag{11}$$

The authors have noted that E_{photo} is very easily affected by small changes to β and that it would need to be optimized, possibly over another integral. Due to this difficulty, they decided to approximate the value of E_{photo} as follows after noting that it did not usually affect the modes of the distribution:

$$E_{photo}(I_V(\mathbf{x})) \approx \min_z \sum_{i=1}^{n} -\rho(||I_V(\mathbf{x}) - c(i,z)||). \tag{12}$$

Finally, the optimization problem over I_V can be summarized as finding the minimum of:

$$E(I_V) = \sum_{i=1}^{m} E_{photo}(I_V(\mathbf{x}_i)) + E_{texture}(I_V(\mathbf{x}_i)). \tag{13}$$

2.6 Optimized Implementation

As the first practical limitation, z is quantized, typically using 500 values bounded by presets z_{min} and z_{max}.

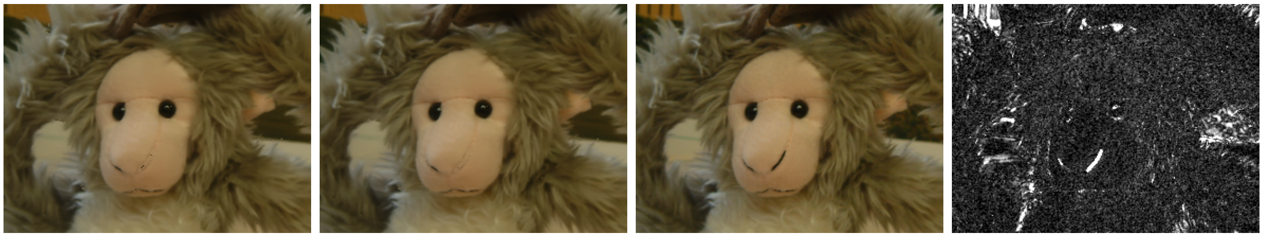
Next, the whole Gibbs Random Field as defined by equation 13 could be estimated directly using global optimization methods such as simulated annealing or Monte Carlo Markov Chain samplers such as the Metropolis and the Gibbs samplers, but both the likelihood and prior terms are not only very expensive to compute, the function contains a lot of local minima. Attaining a global minimum, or even a good local minimum would certainly take a very long time. In order to make the problem more tractable, the modes of the likelihood (equation 12) are first identified using 12 iterations of gradient descent randomly restarted 20 times (a very good random number generator is required). This is justified by the authors by the fact that the likelihood over the depth z contains only a few minima (about 5 on average) which usually contains the color being searched for. For the implementation used in this report, instead of a fixed number of iterations of gradient descent, a steepest descent approach with finite differences was used. Very negligible progress ($< 0.1\%$) made on the energy value or on the color value itself were used as termination criteria. Resulting modes are then clustered for efficiency.

Now from this limited set of modes, one still needs to include the texture prior in order to regularize the problem. A database needs to be defined. Ideally a very large database of natural textures should be used, but in practice, it suffices to take patches from the input images and use those patches as textures for the evaluation of the prior. More precisely, for one given pixel in the new virtual view, one patch with center $\mathbf{x} = P_i\mathbf{X}(z)$ is taken from each image i at all depths z. Those patches are used directly with equation 11 as the set $\mathbb{T}$.

In order to compute 13, the authors used the iterated conditional modes (ICM) algorithm. With this algorithm, the most likely photoconsistent mode over all pixels is first computed, giving a first estimate of the new image I_V^0. From this first estimate, each pixel is considered and its value set to the minimum of equation 13. In effect, it tries to locally minimize the energy, one pixel at a time, but by giving priority to the photoconsistency likelihood. A few iterations over all the pixels in the image are necessary. However, one iteration is still very expensive, and the authors used an approximation of this method. From the modes previously computed, they let the algorithm choose which of the modes minimizes 13. Moreover, the modes are sorted by increasing amounts of energy so that $E_{photo}(I_V(\mathbf{x})) \approx ||I_V(\mathbf{x}) - I_V^{i-1}(\mathbf{x})||$ can be used as approximation to equation 12 for iteration i, since a change in $I_V(\mathbf{x})$ can only result in a configuration with more photoconsistency energy, or at least this is the assumption. It can be shown that this procedure can be accomplished by selecting as follows the value of a pixel $\mathbf{x}$ for iteration i:

$$I_V^i(\mathbf{x}) = \arg \min_{c\in I_V^{modes}(\mathbf{x})} \left|\left| \frac{I_V^{i-1}(\mathbf{x}) + \lambda T(\mathbf{0})}{1+\lambda} - c \right|\right|, \tag{14}$$

where $I_V^{modes}(\mathbf{x})$ are the photoconsistency modes at pixel $\mathbf{x}$ and $T(\mathbf{0})$ is the center pixel value of the texture patch that minimizes equation 11. $\lambda = 1.0$ was used.



(a) ML estimated image. (b) MAP estimated image. (c) Real image. (d) Error image $|(b) - (c)|$.

Figure 1: Comparison between the reconstructed image and the real image for viewpoint number 76 of the monkey sequence. The last remaining 26 views of the sequence of 89 images were used for reconstruction. The MAP estimate is from the third iteration. The error image's intensity range is [0,10] with saturation for larger errors.

3 Results and Analysis

The implementation of the algorithm done for this report was tested over a range of projection camera matrices interpolated from the given monkey sequence available on the authors' Web site `http://www.robots.ox.ac.uk/~awf/ibr/`. The results for one test image, figures 1, are used here for discussion. Image and camera matrix number 76 were removed from the set of input images and matrices. Out of the full sequence of 89 views, reconstruction was then attempted using the 26 last remaining views, which are closest to view number 76.

Strangely enough, possibly thanks to the use of steepest descent, the images obtained from the new implementation are much cleaner than the ones presented in the paper. As one can see from figure 1(a), the photoconsistent maximum likelihood estimate of the view is already very clean, with only a few outliers present. The texture prior smooths things out a bit (figure 1(b)), but almost nothing needs to be cleaned up. The only two big remaining problems with this image when compared to the real image (figure 1(c)) are with the brown structure on the background wall in the upper left corner and with the black strip on the monkey's nose. For most of the pixels in those areas, the colors found to have the minimum energy are incorrect. Even though the true colors are actually also modes discovered by the photoconsistent likelihood, the texture prior cannot recover from this mistake since these patches of wrong color still produce valid smooth natural textures. The error image (figure 1(d)) clearly indicates the problematic areas. However, it is remarkable that 98.9% of the pixels are within a Manhattan distance of 10 from their true colors in the RGB color space.

Other incorrectly colored pixels also arise from the fact that the true colors are not always minima of equation 12. The authors introduced a robust kernel (M-estimator) $\rho(x) = |x|$ in order to mitigate the issue, but it does not completely solve it. The problem occurs when, at a given depth, more than half of the colors returned by the available views differ from the real color of interest. In this case, the absolute value kernel usually fails and the color of interest does not become a minimum. Two main scenarios were identified where this problem occurred. First, when one large patch of mostly uniform color occludes the object of interest in more than half of the views. Second, when a large enough patch of mostly uniform color somewhere in space is actually mapped *behind* or *in front of* the object of interest. The patch is large enough to permit multiple viewpoints to see the same color, even though it violates geometric constraints, as it is mapped to an invalid depth. There are a few things not all mentioned in the paper that can be done to help:

- Manually tweak z_{min} and z_{max}, or even create intervals within the range of z so that empty areas of space are not considered.

- As input data, use only a few views close to the virtual view. Closer views have more overlap and less occlusion, reducing the possibility of unrelated information from interacting together.

- Use a more robust kernel that will create minima even in the situations discussed above. Unfortunately, more robust kernels such as $\rho(x) = 1 - \exp(-|x|/k)$ and $\rho(x) = \min(|x|, k)$ also create a lot of other spurious minima, which makes it very hard for the gradient descent algorithm to narrow down the number of possible colors.

- Perform regularization on the depth map as well, using for example an operator akin to a median smoothing filter, in order to smooth out high frequency discontinuities while leaving unaffected discontinuities at the edges of objects. Other techniques analogous to space carving should also provide significant improvements as the authors pointed out.

On the other hand, one might think that with all those optimizations and approximations that the algorithm would run reasonably fast, but alas, it is still excruciatingly slow. To render a full 640 × 480 image, even with a well coded implementation running on the fastest personal computer, it might well take over one whole 24 hour day.

4 Conclusion

For this report, the view synthesis algorithm of Fitzgibbon et al. [1] was implemented and tested. The details of the algorithm were summarized, and results presented as well as analyzed. It was found that although the algorithm effectively regularizes the synthesized image using texture information as prior, the quality of the initial photoconsistent estimate is already very high. The algorithm however still needs major improvements in terms of computational performance.

With regards to myself, a lot of the concepts that I learned during my ESCE 626 - Statistical Computer Vision course greatly helped me understand the material of the paper. In particular, I was able to immediately understand the initial Bayesian formulation and thus the reason why equation 3 holds. I was also able to notice that the forms used for the likelihood and the prior are actually Gibbs densities. This allowed me to use tools related to Gibbs Random Fields and to fully grasp the simplifications made and other concepts such as why global optimization approaches like simulated annealing or Markov Chain Monte Carlo sampling would be too slow for this application. Amazingly, I find, it truly made me see things in a different light, and put me in a better position to built possible improvements on this algorithm in the future.

Acknowledgments

Thanks to Tal Arbel for all the great material learned in class. Also, help from Cathy Laporte and Andrew Fitzgibbon understanding the material of the paper and the limitations of the algorithm is gratefully acknowledged. Thanks as well to Paul Tupper for gladly providing information about gradient descent and related techniques. The author also wishes to thank the Natural Sciences and Engineering Research Council of Canada for the generous postgraduate scholarship funding received.

References

[1] A. Fitzgibbon, Y. Wexler, and A. Zisserman, "Image-based rendering using image-based priors," in *Proceedings of the International Conference on Computer Vision*, vol. 2, pp. 1176–1183, Oct. 2003. `http://www.robots.ox.ac.uk/~awf/ibr/`.

[2] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second ed., 2004.

[3] S. Z. Li, *Markov random field modeling in computer vision*. London, UK: Springer-Verlag, 1995.