Joint Conditional Random Field of Multiple Views with Online Learning for Image-based Rendering

Wenfeng Li and Baoxin Li
Department of Computer Science and Engineering, Arizona State University
{Wenfeng.Li, Baoxin.Li}@asu.edu

Abstract

There are many applications, such as image-based rendering, where multiple views of a scene are considered simultaneously for improved analysis through employing strong correlation among the set of pixels corresponding to the same physical scene point. While being a useful tool for modeling pixel interactions, Markov Random Field (MRF) models encounter challenges in such cases since they assume strong independence of the observed data for tractability, rendering it difficult to take advantage of having multiple correlated views. In this paper we propose joint Conditional Random Field (CRF) for multiple views in the context of virtual view synthesis in image-based rendering. The model is enabled by the adoption of steerable spatial filters for capturing not only the pixel dependence in a single image but also their correlations among multiple views. Furthermore, a novel on-line learning scheme is proposed for the CRF model, which learns the CRF parameters from the same input data for synthesizing virtual views. This effectively makes the model adaptive to the input and thus optimal results can be expected. Experiments are designed to validate the proposed approach and its effectiveness.

1. Introduction

Given multiple images captured from different viewpoints of a 3D scene, to synthesize a photorealistic virtual view from an arbitrary viewpoint is a main goal of image-based rendering (IBR) [4, 5], which has many applications and has received much attention in the relevant fields. In [4], Shum and Kang reviewed various IBR techniques and classified them into three categories based on whether the scene geometry is assumed or utilized: rendering with explicit geometry, rendering with implicit geometry, and rendering without geometry. In the first category, the virtual view can be produced by projecting pixels from all the reference images. View-dependent texture-mapping [6], 3D warping [7], and layered-depth images [8] are typical methods that belong to this category. If the camera geometry, usually presented as a projection matrix, is known, the 3D coordinate of every point is determined by the 2D coordinate on image and the depth z. The depth z may be obtained by stereo matching [9] using only the input images, which remains to be a challenge in general if a dense depth map with accuracy is desired, or by other special techniques (e.g., using a laser ranger finder or structured light), which may be difficult to assume for many IBR applications. The third category of IBR techniques, such as light field rendering [10], use a large number of cameras to capture many views and do not assume the scene geometry. In between these two extremes, other approaches attempt to find the best trade-off between demanding more images and requiring more accurate scene geometry. Our work in this paper belongs to this category, where only a few views (6 to 10 in our experiments) are used to synthesize a virtual view without computing accurate scene geometry.

We adopt the basic formulation for IBR as in [11], where virtual view synthesis is expressed as a maximum likelihood estimation (MLE) problem. In [11], a texture dictionary is used as the prior in a Markov Random Field (MRF) model. Woodford *et al.* [20, 3] extended this work by using a different MRF prior with field of experts and pairwise dictionaries. While being useful for modeling pixel interactions, MRF assumes strong independence of the observed data for tractability. Recently, Conditional Random Field (CRF) [2] was proposed, which does not suffer from such limitation. CRF directly models the posterior as a Gibbs distribution and allows arbitrary dependencies among the observed data. As spatial dependencies among pixels and textures are abundant in natural images, CRF draws a lot of interest from researchers in image processing and computer vision, and it has been applied to image segmentation [21], image labeling [22], and stereo matching [19], etc. In this paper, we propose a joint CRF framework that models not only the pixel dependence within a single image but also the correlation of the pixels across multiple views.

The power of CRF first comes from its flexible graph structure. As pointed out by Roth and Black [12], MRF is limited by their neighborhood structures. Most models used in image and vision problems are the 4-connectness neighborhood models. High-order models are possible but the learning is difficult [13, 14]. In contrary, the graph

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

model used in CRF is not restricted and it even does not have to be a graph. Earlier CRF application such as speech recognition uses simple chain graph [2]. Lattice structure and spatial filter is used in recent work for images [1, 17, 22]. In this paper, we propose to model the pixel dependencies within a local neighborhood based on the outputs of linear steerable filters across multiple views. The steerability of the filters leads to efficiency in computation since only a limit number of filtering directions are needed. The support of the filter can include more than a few of pixels and thus improve the robustness to noise.

CRF also facilitates better parameter learning. In MRF, the relative effects of the prior and the data likelihood are weighted by a regularization coefficient, which is fixed with few exceptions such as in the work of Zhang and Seitz [23]. We show in this paper how the learning can be done with a stochastic method to obtain a free-form curve for the unknown parameters. Further, we argue that the learning is performed on-line using only the given views of a scene (as opposed to an offline scheme based on images of different scenes). This essentially adapts the parameters of the model to the specifics of the given views and thus optimal results may be expected.

In Section 2, we first briefly present the basic formulation for IBR and then propose the joint CRF model of multiple views, followed by an on-line learning algorithm for estimating the model parameters. Inference of a virtual view under the learnt model is straightforward and hence is discussed briefly. Experimental validation is given in Section 3, and Section 4 concludes the paper.

2. Proposed Approach

2.1. Probabilistic Formulation of IBR

The virtual view synthesis problem can be described as: given a set of captured views $X=\{X_j \mid j = 1,\ldots,N\}$ of a 3D scene, compute a virtual view Y, where $X_j = \{x^j(r, c)\}$ and $Y = \{y(r, c)\}$, with x and y representing pixels. For simplicity, X_j and Y are always treated as 2D matrices. We assume known camera geometry in the form of the projection matrices for both the reference views X_j's and the virtual view Y. (Otherwise, self-calibration and virtual view specification need to be applied first.) Figure 1 illustrates the geometry of the virtual view and the reference views. For each pixel $y(r, c)$ in the virtual view, it can be back-projected to a 3D ray which connects the virtual camera optical center and the image point. The projection of this ray on each reference view forms the epipolar lines. The only undetermined value is the depth z. The configuration of all z is noted as $D=\{z(r, c)\}$. Following Bayes' rule:

$$P(Y,D \mid X) = \frac{P(X \mid Y,D)P(Y,D)}{P(X)} \tag{1}$$

For given observed data X, (1) can be written as:

$$P(Y,D \mid X) \propto P(X \mid Y,D)P(Y,D) \tag{2}$$

Although D is in the posterior term, the goal of virtual view synthesis is different from stereo matching. Here Y is the main term of interest, and D is only a by-product. This means that as long as synthesized view Y is good enough, a poor estimate of D is not of concern. For example, for regions of an image with little textures, accurate per-pixel depth may be difficult to obtain. However, the synthesized view could still look realistic for those regions.

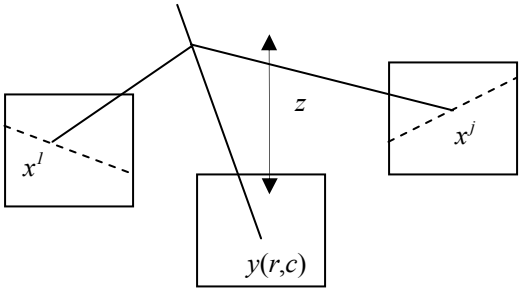

Figure 1: Geometry of virtual view and reference views.

A baseline approach to evaluating (2) is to consider the pixels as i.i.d. random variables, and thus the posterior can be written as the product of the per-pixel posterior,

$$\prod_{(r,c)}\prod_{j=1}^{N} P(x^j \mid y,z)P(y,z) \tag{3}$$

With any likelihood model, one can find an MLE solution by focusing on that term only. To improve upon MLE, the prior term needs to be considered, which may be modeled through a potential function in random fields such as in [11]:

$$P(y,z) = \prod_{V} P(y_v, z) \tag{4}$$

where V is defined as a set of neighbor indices on site (r, c), and 4-neighbor model is commonly used.

Incorporating a simple prior model still does not fully account for the strong spatial dependencies in the observed data. For example, assume that there is a strong edge in all the reference views, a corresponding edge should be expected in the virtual view. In below, we will show how to use a CRF model to capture such dependencies.

2.2. Joint CRF of Multiple Views

The virtual view synthesis problem can be formulated as one of labeling the pixels of the virtual image Y with a finite label set C, which is the color space, given the set of observed images $\{X_j\}$. Let $G=(V,E)$ be a graph such that Y is indexed by the vertices of G. Then (X,Y) is a conditional random field if, when conditioned on X, the random variables y obey the Markov property with respect to the graph, i.e., $P(y_u|X, y_w, w \neq u) = P(y_u|X, y_{w,}, w\sim u)$, where u, w are 2D coordinates in the image and $w \sim u$ means w is defined as neighbors of u in G. We construct the following CRF model

$$P(Y \mid X) = \frac{1}{Z}\exp\left(\sum_{e\in E,k}\lambda_k f_k(y|_e, X) + \sum_{v\in V,k}\mu_k g_k(y|_v, X)\right) \quad (5)$$

where Z is a normalizing constant. Following the terminologies in [17], g_k is called the association potential and f_k the interaction potential. λ_k and μ_k are the parameters of the CRF. $y|_s$ is the set of components of y associated with the sub-graphs of S. By defining the association and interaction potentials in ways that capture the dependency of pixels not only within the virtual view but also across other given images, we effectively obtain a joint CRF of multiple views, in which the interaction of the pixels are realized through the epipolar geometry and locally-supported spatial filters, as illustrated in Figure 2. The details of the potential functions will be discussed in subsequent subsections, where it will also become clear that the pixel interactions can be in terms of color and other filter outputs. This gives us a way of modeling both the color and the texture of the synthesized view globally conditioned on the reference views.

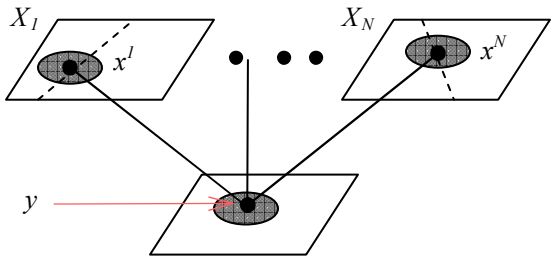

Figure 2: Joint conditional random field of multiple views.

Note that, in (5), the depth D is not expressed explicitly as it is only used to compute point correspondence but not the CRF inference. (We will show later that depth is actually coupled with pixel colors by the geometry constraint.)

2.2.1 *Association potential*

In the proposed CRF-based modeling in (5), we use the association potential $g(y,X)$ to measure the similarity of the synthesized pixel and the pixels from input images. Hence the association potential plays the role of the likelihood. We first study how the likelihood $P(x^j|y,z)$ should be modeled, where x^j denotes the point on (r^j, c^j) corresponding to y. In principle, to compute the likelihood, y and z has to be enumerated in the color space and the depth space. In practice, z is considered to follow a uniform distribution within certain range and can be discretized evenly into a finite set of sampling points. The range may be estimated from feature points used in the camera calibration stage. While enumerating z alone is tractable, enumerating both z and y is impractical, since the dimension of the color space is 256^3 for 24-bit color images and this has to be done for each pixel on all possible z values. Fortunately, we show in the below that this is not necessary. For one point (r, c) with z, the pixels sampled from the reference views are determined by the camera geometry. If the negative logarithm of the probability is used as an energy function, the energy that needs to be minimized is:

$$E = \sum_{j=1}^{N} \| y - x^j \|^2 \quad (6)$$

where the Euclidian distance in the RGB color space is used to measure the similarity of pixels. As the function has a quadratic form, the best color that minimizes (6) is the mean value of all x^j's.

In order to be robust to noise, an energy function with robust kernel is often used and we write the association potential with negative energy function as

$$g = -\sum_{j=1}^{N} \min(\| y - x^j \|^2, \tau) \quad (7)$$

where τ is a cut-off threshold. However, to find a y to minimize (7) is not trivial as there is no close form solution. An iterative algorithm has to be involved by starting with an initial guess.

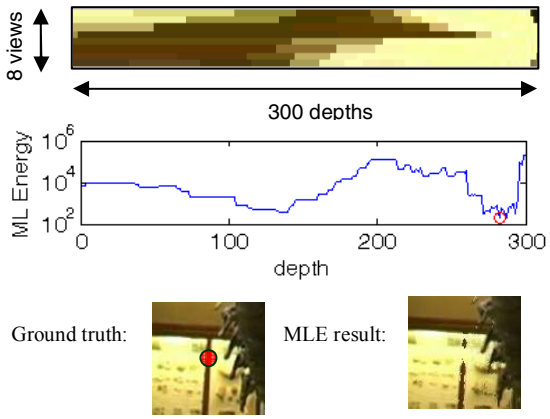

Figure 3: Using Maximum Likelihood Estimation to synthesize one point (marked as red) leads to error.

2.2.2 *Interaction potential*

One problem with the likelihood model is that it is based on individual pixels without considering the strong dependency among pixels. For example, texture-less regions may always give the most consistent color and may overshadow small textures. This is illustrated in Figure 3, where a point on a bar pattern (marked in red) is mistakenly synthesized as the surrounding background color. This can be a serious problem in virtual view synthesis as the MLE solution is often used as a starting point to update certain prior model that is typically defined on a neighborhood. In the example of Figure 3, the MLE solution is too far from the truth and thus it is unlikely to be useful for updating the prior model to obtain the correct value. Woodford et al. proposed a solution in [3], where multiple modes are

computed and stored in memory so that the likelihood and prior term can be optimized at the same time.

Considering the fact that a simple pixel-value-based likelihood modeling is not sufficient, we propose to use an interaction potential function to capture the correlations among the reference views. Specifically, we define the interaction potential in such a way that it measures the similarity of the pixels across views in terms of edge and texture. This is achieved by using a set of spatial filters to robustly track pixel variance across multiple views (the filters are further discussed in Section 2.2.3). For each y on site (r, c), its corresponding points on the reference views with a given z are $\{x^j\}$ and the responses with the k-th filter on those points are $\{r_k^j\}$. The interaction potential term can be written as a negative energy function

$$f_k = -\sum_{j=1}^{N} \min(\| s_k - r_k^j \|^2, \tau) \tag{8}$$

where s_k is the k-th filter response on the synthesized view. The role of the interaction potential is to model the dependence of the synthesized pixels (textures on the synthesized image) conditioned on the observed data (textures on the reference images). However if s_k is computed by filtering the synthesized image, it can only be computed after initial values of y are obtained. Relying on result from only the association potential will undermine the use of the interaction potential. We use a two stage strategy: in the first stage, association and interaction potential are used simultaneously to search for a best depth z for each pixel, but s_k is computed from r_k^j in the same way as y is computed from x^j; In the second stage, we optimize the random fields while s_k is computed from y. By doing so, textures are preserved and reinforced, false responses to textures may rise but will be penalized in the second stage.

In our implementation, we use only one filter set (the first row of Figure 4) which captures the color gradients. This can be justified by reports from the literature that intensity gradient is the richest feature in natural images and is most useful for feature tracking. More filters can be used as in [13] to further improve the performance but at the cost of more computational power. Hereafter, we drop the subscript k for the interaction potential and write the CRF model as

$$P(Y \mid X) = \frac{1}{Z} \exp\left(\sum_{(r,c)} \lambda\, f + \sum_{(r,c)} \mu\, g \right)$$

$$g = -\sum_{j=1}^{N} \min(\| y - x^j \|^2, \tau_g)$$

$$f = -\sum_{j=1}^{N} \min(\| s - r^j \|^2, \tau_f) \tag{9}$$

where λ and μ are the CRF parameters.

2.2.3 *Using steerable spatial filters*

The multi-view CRF defined above relies on the modeling of pixel dependency through measuring the similarity among the outputs of the steerable filters, which is illustrated in Figure 4. In the figure, the filters in the first column have the following mathematical expression:

$$H(x, y) = G_n(x) * G_0(y) \tag{10}$$

where $G_n(u)$ denotes the n-th order Gaussian derivative filter on direction u. Each row in Figure 4 is a rotated version of its first filter. According to the order of Gaussian derivative, each row has different number of rotated filters as the bases. This is determined by the steerable filter theorem [16] that states that any n-th order Gaussian derivative filter can be steered by n+1 rotated bases. For instance, for the first order Gaussian derivative filter, its rotated version can be represented by a linear combination of the two bases in the first row.

$$R^{\theta}(x, y) = R_1 \cos\theta + R_2 \sin\theta \tag{11}$$

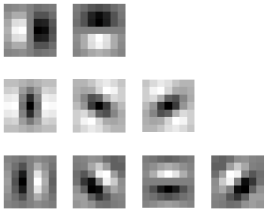

Figure 4: Steerable linear spatial filters.

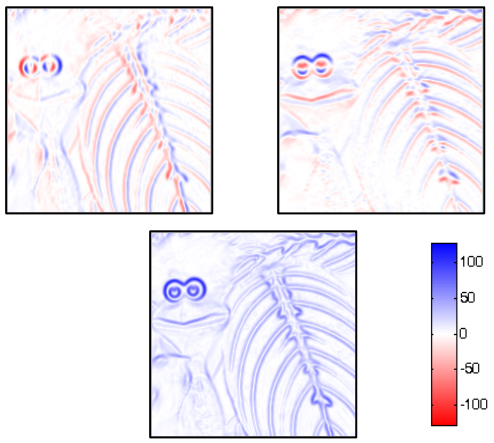

Figure 5: One input image after being applied the two basis filters separately (top left and top right). Note that one detects vertical edges and the other detects horizontal edges. After being steered to strongest gradient orientation (bottom), all edges are detected.

This steerability feature provides an efficient way to handle the rotation of the image patches across multiple views. Using the first row as example, which can be viewed as edge detector, for each point on the images, the filter responses R_1 and R_2 with two bases are first computed, then the orientation with the strongest response can be obtained analytically [15,16] by $\theta = \tan^{-1}(R_1/R_2)$. A new filtered image is generated by steering all points to the orientation of its strongest response as illustrated in Figure 5.

The mean of filter responses on all color channels is used to find the maximum gradient since the gradient on each color channel may not be consistent. Once the orientation is determined, new responses are computed on each color channel separately. This can be efficiently done by using (11) as the filters are steerable. Using gradient on three channels may seem redundant as they are not independent. However, our experiments show that doing so slightly improve the performance. In our experiments, the orientations are not used again when computing the similarity measure. This worked out fine since the possibility of multiple points having the same filter response in three channels only with different orientations is small.

2.3. Online Learning

To learn the CRF parameters in (5), there has to be a set of images with depth map as ground truth. Obtaining such data is expensive and/or time-consuming. Even if this may not be an issue if we use synthesized data from some simulation software, one wonders if the parameters learned from a training set are optimal or even good enough for a new set of data. To address this problem, we introduce the following online learning approach: we learn the model parameter by using one reference view as the ground truth and attempting to synthesize this view from other views. In this way, the parameter learning is purely based on the input data itself and thus presumably adaptive to the input (which is indeed the case as will be illustrated later).

The task of learning is to estimate the parameter $\Theta = \{\lambda, \mu\}$ which best explains the given data according to the model in (5). It is equivalent to maximizing the following log conditional likelihood

$$\begin{aligned} L(\Theta) &= \sum_{r,c} (\lambda f + \mu g) - \log Z \\ &= \sum_{r,c} \Theta^T \mathbf{F} - \log Z \end{aligned} \tag{12}$$

where $\mathbf{F}=[f;g]$. Differentiating with respect to Θ, we have

$$\frac{\partial L(\Theta)}{\partial \Theta} = \sum_{r,c} \mathbf{F} - \left\langle \sum_{r,c} \mathbf{F} \right\rangle_{p(y,z)} \tag{13}$$

The second term in (13) denotes the expectation under probability distribution $p(y,z)$. In CRF with simple graph like chain model, the expectation can be computed efficiently with a dynamic programming method, similar to the forward-backward algorithm for hidden Markov models. However, for images, the CRF graph model is more complex, making it intractable to compute the expectation. Scharstein and Pal [19] used graph-cuts to minimize the energy function when the partition function in their model is constant. We use a gradient descent method which is similar to the one used in [1] where parameters are learned by penalizing the difference between the mode of the conditional distribution and the ground truth. From practical consideration, we limit the learning to the center of cropped virtual view to guarantee that each pixel in the virtual view can be mapped to a valid area of all reference views on all possible depth.

CRF parameters must be adaptive to input signals which are $\{x^j\}$ and $\{r^j\}$ in this particular application. The dimension is $6N$, where N is the number of reference views. Using such high dimensional data to model the parameters is difficult and unnecessary. Note that the goal of the association potential and the interaction potential is to find a reconstructed color y and filter response s by measuring the consistency among $\{x^j\}$ and $\{r^j\}$. For a good estimation, $y \approx x^1 \approx \ldots x^N$ and $s \approx r^1 \approx \ldots r^N$, therefore we can use y and s as input signals. The motivation of the parameter design is that when a filter response is strong, the model should trust more on the interaction potential and less on the association potential. In this paper we focus on exploiting such weighting effect by writing λ as a constant and μ as a function of the norm of s.

As the form of the function $\mu(\|s\|)$ is unknown, we use a non-parametric function with 128 discrete points $\mu[i]$, i=0, … 127 and for an arbitrary value of $\|s\|$, μ is computed by a linear interpolation of two points with indices nearest to $\|s\|$. The filters are also designed to limit the norm of the responses in the range of 0 to 127.

A gradient search method must have a good initialization. We observe that the MLE method can synthesize most pixels reasonably well and thus we can use the MLE solution for the initialization. In practice, by letting λ to a very small value, i.e., $\lambda = 0.05$ and $\mu[i]=1$ for all i, we obtain a model which is almost identical to an MLE model and this will be used to initialize the learning stage.

The gradient descent problem can be described as, given an objective function $F(x)$ at point x', find the direction Δx so that $F(x'+\Delta x) < F(x')$, and x' is updated with $x'+\Delta x$. Based on (5), for one pixel in the virtual view based on the current CRF parameters, the objective function can be defined as

$$F = -\mu(\|s\|)g - f \tag{14}$$

For the training step, the depth z of one pixel can be chosen as the value that renders the synthesized color closest to the real color, and we can obtain s' for this (ground truth) depth z. Let its corresponding objective function value be F', the goal is to let $F'<F$. Since in (14) all other terms are fixed except μ, the new parameters should let $\mu(||s||)<\mu(||s'||)$. This could be achieved by decreasing the parameter points $\mu[\text{ceil}(||s||)]$ and $\mu[\text{floor}(||s||)]$. Note that, in principle, to re-evaluate the new parameters, the whole image must be synthesized for each processed pixel. To avoid such inefficiency, we accumulate the desired update for $\mu(||s||)$ for all pixels of an image and make the update only once. The entire procedure is summarized in the following algorithm.

On-line learning of the CRF parameters

1. Pick one reference view from the input images as training target view.
2. Use the other N-1 reference views and the current parameters to synthesize a view.
3. Check all pixels that have large errors and take the following vote:
   -Initialize Δ= [0, 0, … 0]
   -Use the ground truth pixel to find the best depth z and compute its corresponding filter response s'.
   -If $s' > s$
   Increase Δ[ceil(||s'||)] and Δ[floor(||s'||)]
   Else
   Decrease Δ[ceil(||s'||)] and Δ[floor(||s'||)]
   End if
4. Update parameters with $\mu = \mu$ - step $\cdot\Delta$
5. Exit if meets stop criteria, otherwise go to 2

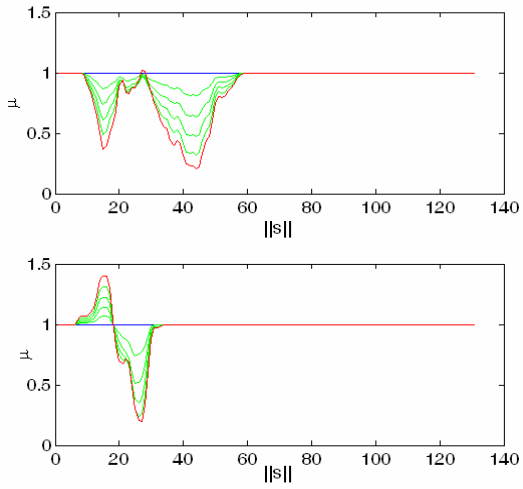

Figure 6: Two sets of CRF parameters learned from two different image sets. Green curves show the transition of parameters from initial value (blue) to final result (red) in every 10 iterations.

Figure 6 shows how μ is updated through learning. We empirically fixed the maximum iteration to be 50 as the stop criteria. Note that the result is not a monotonous curve. It is also found that the parameters greater than 60 are not updated at all. This is because in the images used for training, no filter response has the value greater than 60 and thus those parameters will not affect the inference with the conditional model. This also shows that the proposed CRF-based model and its parameter learning are indeed adaptive to the input data, verifying the idea of online learning. This point is further illustrated by the two plots in Figure 6 that are two sets of parameters learned from two different datasets, showing the dramatic differences in the learnt parameters. With parameters updated in learning, a sequence of virtual view can also be synthesized and Figure 7 demonstrates that the root-mean-square error decreases and the image quality increases. This proves that the adaptation of parameters is effective.

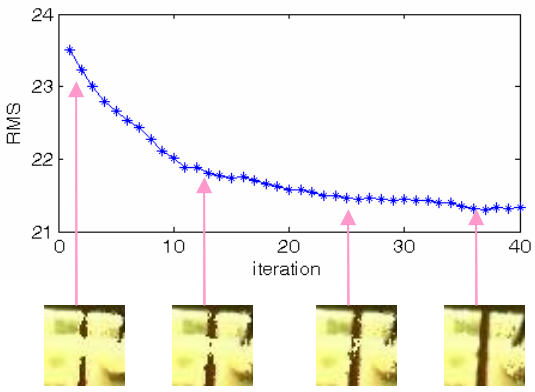

Figure 7: Synthesized virtual view with updated parameters in learning.

2.4. Inference

The random field optimization has to be approximated as the problem is NP-hard. Gibbs sampling could work well in such a problem but may take a long time to converge. We use a two-stage inference procedure. First we find initial values of y and s from reference images. In the second stage s is computed by filtering y and both the association potential and the interaction potential are used to maximize the conditional probability. As the result from the first stage is good enough, we found that a simple Iterated Conditional Mode (ICM) approach suffices this objective. For each point, a better mode is probed by looking at its neighbors, if the color of one of its neighbors gives greater value of conditional probability, it is accepted as the new color.

3. Experiments

We use the datasets from [3,11,13] in our experiments. For comparison, we also implement the MLE method and its variation with a robust kernel as in (7). To find the best color y to minimize (7), a Gaussian mixture model (GMM) is used to find the most consistent cluster of pixels with an EM algorithm [18]. The cluster of pixels is fitted with a Gaussian component and outliers are modeled with a uniform distribution. We also tested the mean-shift algorithm to estimate the best color iteratively as suggested by [3], which performed slightly worse than the GMM method in terms of the quality of the synthesized images. This is because GMM method is more robust to the cut-off threshold used in the robust kernel. Therefore, in the following, we only compare with the mixture model.

The leave-one-out test result is listed in Figure 9 where one view is used as the ground truth and a new view is synthesized with its projection matrix. Eight closest views are used as reference views. The results show that MLE

tends to smooth the synthesized images and sometimes blends different layers, which renders visually pleasing images but creates significant artifacts where occlusion occurs. GMM works much better to handle occlusion. The proposed CRF-based method outperforms the above two methods in all experiments in term of both root-mean-square (RMS) error and error rate which is defined in [3] as the percent of pixels with sum of squared errors greater than 1000. As we do not have the exact configuration to reproduce the other works for detailed comparison, based on the results, we only claim that our results are comparable to the state-of-the-art approaches such as [3] and [20] in both RMS error and visual quality.

Figure 10 are two complete virtual views synthesized with our approach. Notice that our results with CRF preserve some fine details like the stem and textures on the leaves, comparing with magnified blocks from MLE results.

We also tested our method with drastically different reference views. The results are given in Figure 11. As expected, the performance is degraded compared with the case with close reference views. But still the performance is reasonable especially given the large difference of the reference views. Those failures are mainly due to breaking of Lambertian surface assumption where there is strong reflection and the colors for one point are different from different viewpoints.

For the speed performance in term of rendering time, the proposed method requires 3 times more than the MLE method, which is comparable to those MRF methods as reported by [3] (with about 5 to 8 times of the MLE method in their implementation). While applying filters on all reference views and steering the results require extra time, our experiments show this time is relative much smaller than the rendering time and thus can be ignored.

4. Conclusion and Future Work

In this paper, we proposed joint CRF of multiple views for virtual view synthesis. We also presented an online learning algorithm for estimating the optimal parameters for the model. Our experiments show that the model is effective and the learning algorithm is a feasible solution. For future work, we expect to expand the set of filters, which will demand modification to the learning algorithm. We will also extend the framework to views with only weak calibration.

5. References

[1] M. F. Tappen, C. Liu, E. H. Adelson, and W. T. Freeman, Learning Gaussian conditional random fields for low-level vision, CVPR 2007.

[2] J. Lafferty, A. McCallum and F Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, ICML 2001, pp. 282-289.

[3] O. Woodford, I. Reid, and A. Fitzgibbon, Efficient new-view synthesis using pairwise dictionary priors, CVPR 2007.

[4] H. Y. Shum and S. B. Kang, A review of image-based rendering techniques, IEEE/SPIE *Visual Communications and Image Processing* (VCIP), pp. 2-13, 2000.

[5] C. Zhang and T. Chen, A survey on image-based rendering: representation, sampling and compression, *Signal Processing: Image Communication*, vol. 19(1), pp. 1-28, January 2004.

[6] Paul E. Debevec, George Borshukov, and Yizhou Yu. Efficient view-dependent image-based rendering with projective texture-mapping. *In 9th Eurographics Rendering Workshop*, Vienna, Austria, June 1998.

[7] W. Mark, L. McMillan, and G. Bishop. Post-rendering 3d warping. *In Proc. Symposium on I3D Graphics*, pp. 7–16, 1997.

[8] J. Shade, S. Gortler, L.-W. He, and R. Szeliski, Layered depth images, *In Computer Graphics (SIGGRAPH'98) Proceedings*, pp. 231–242, 1998.

[9] D. Scharstein, Stereo vision for view synthesis, CVPR 1996, pp. 852-858.

[10] M. Levoy and P. Hanrahan. Light field rendering, *In Computer Graphics (SIGGRAPH'96) Proceedings*, pp. 31–42, 1996.

[11] A. W. Fitzgibbon, Y. Wexler and A. Zisserman, Image-based rendering using image-based priors, ICCV 2003, pp. 1176-1183.

[12] S. Roth and M. J. Black, Steerable random fields, ICCV 2007, pp. 1-8.

[13] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors, CVPR 2005, pp. 860–867.

[14] J. J. McAuley, T. S. Caetano, A. J. Smola and M. O. Franz, Learning high-order MRF priors of color images, ICML 2006, pp. 617-624.

[15] D. G. Jones and J. Malik, A computational framework for determining stereo correspondence from a set of linear spatial filters, ECCV 1992, pp. 395-410.

[16] W. T. Freeman and E. H. Adelson, The design and use of steerable filters, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(9), pp. 891-906, 1991.

[17] S. Kumar and M. Hebert, Discriminative fields for modeling spatial dependencies in natural images, NIPS 2003.

[18] D. J. Miller and J. Browning, A mixture model and EM based algorithm for class discovery, robust classification, and outlier rejection in mixed labeled/unlabeled data sets, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(11), pp. 1468-1483, 2003

[19] D. Scharstein and C. Pal, Learning conditional random fields for stereo, CVPR 2007.

[20] O. Woodford, I. Reid, P. Torr and A. Fitzgibbon, Fields of experts for image-based rendering, BMVC 2006.

[21] J. Verbeek and B. Triggs, Scene segmentation with CRFs learned from partially labeled images, NIPS 2007.

[22] X. He, R. S. Zemel, M. A. Carreira-Perpinan, Multiscale conditional random fields for image labeling, CVPR 2004, pp. 695-702.

[23] L. Zhang and S. M. Seitz, Parameter estimation for MRF stereo, CVPR 2005, pp. 288-295.

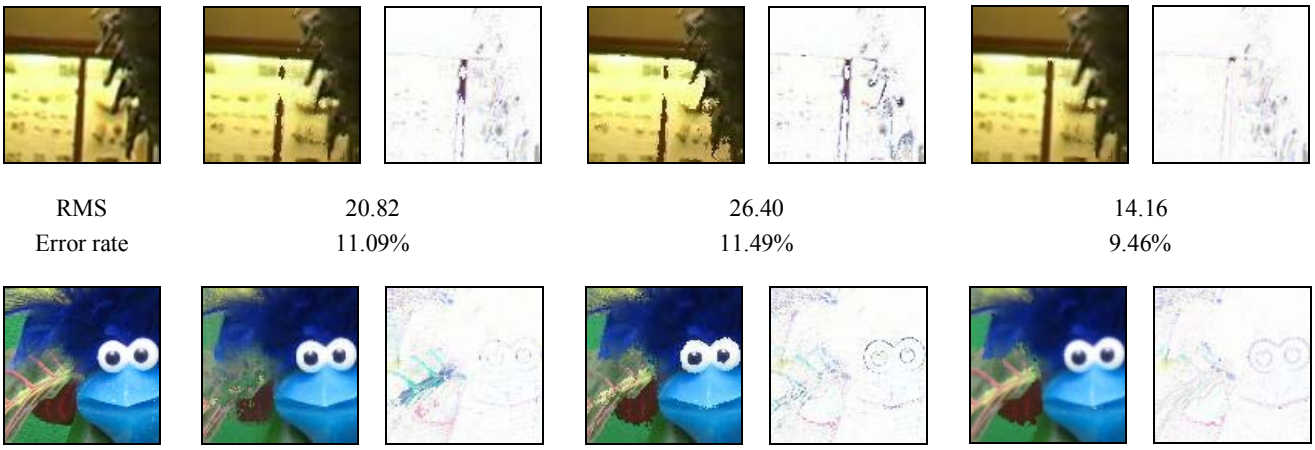

Figure 9: Leave-one-out test on Edmontosaurus and plant & toy data. Shown to the left of each synthesized view is the error frame.

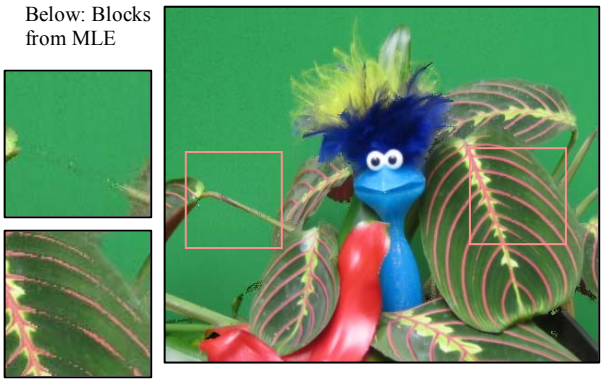

Figure 10: Complete synthesized view with CRF, top: Plant & toy, two blocks synthesized with MLE for comparison; bottom: Monkey.

Figure 11: Synthesized image with drastically different reference views (top), using N=8. Two farthest reference views (bottom) used to render the new view. This not-so-good result is intentionally kept to illustrate the robust of the algorithm when the input views are very different.