

Texture Replacement in Real Images

Yanghai Tsin[†]

Yanxi Liu[†]

Visvanathan Ramesh[‡]

[†] The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{ytsin, yanxi}@cs.cmu.edu

[‡] The Imaging Department
Siemens Corporate Research
Princeton, NJ 08540
rameshv@scr.siemens.com

Abstract

Texture replacement in real images has many applications, such as interior design, digital movie making and computer graphics. The goal is to replace some specified texture patterns in an image while preserving lighting effects, shadows and occlusions. To achieve convincing replacement results we have to detect texture patterns and estimate lighting map from a given image. Near regular planar texture patterns are considered in this paper. Given a sample texture patch, a standard tile is computed. Candidate texture regions are determined by mutual information between the standard tile and each image patch. Regions with high mutual information scores are used to estimate the admissible lighting distributions, which is represented by cached statistics. Spatial lighting change constraints are represented by a Markov random field model. Maximum a posteriori estimation of the texture segmentation and lighting map is solved in a stochastic annealing fashion, namely, the Markov Chain Monte Carlo method. Visually satisfactory result is achieved using this statistical sampling model.

1. Introduction

Texture replacement in real images has many applications, such as interior design, digital movie making and computer graphics. The goal is to replace some specified texture patterns in an image while preserving lighting effects, shadows and occlusions. To achieve visually convincing replacement two separate maps must be obtained, namely the texture segmentation map and the lighting map. Human labeling of the texture map can be very tedious, and it is not clear for a human how to determine lighting at a certain pixel. It is thus appealing to compute the two maps automatically. This paper is an attempt to solve the problem of replacing planar near regular patterns.

Computing the two maps corresponds to two computer vision problems, the texture segmentation problem[1, 8] and the color constancy problem[5, 2]. Texture segmentation algorithms have mainly been applied to cases with

uniform lighting. Healey and Wang [7] developed an illumination invariant texture recognition algorithm. Their method is based on statistics of texture regions and can not determine texture boundary, which is an important issue in our application.

The similarity between texture replacement problem and the color constancy problem is that both attempt to recover lighting changes from image appearance. Theories of color constancy have been established and we can borrow some results from the area. We will adopt the Bayesian color constancy algorithms [2, 5] to deal with illumination change. The Bayesian color constancy framework elegantly incorporates prior knowledge, such as lighting and reflectance distribution, into an inference engine.

Due to the complexity of a real illumination environment and camera artifacts, it is usually very difficult to derive a parametric model for the observed lighting distribution. Instead we put the problem in a particle filtering framework[9, 5]. We estimate light samples from positively identified texture tiles, and a resampling technique is used to estimate sample density when needed. To speed up the resampling procedure we adopt the cached statistics model[13].

2. Lighting Distribution Models

We will discuss two important lighting distribution models in this section. These distributions describe the physical feasibility of the light spectrum and the frequency of the spatial lighting change respectively. First we define what we mean by “lighting”.

We adopt the diagonal transformation model to describe apparent color changes [4]. The accuracy of this model depends on the spectral sensitivity function of the imaging device. And we find it is accurate enough for our application. Denoting the color under a canonical lighting as $\rho^c \in \mathcal{R}^3$, the color observed under the new lighting is

$$\hat{\rho} = l. * \rho^c \quad (1)$$

where the operator $\cdot*$ is defined as element-wise product of two vectors and we call the linear scaling coefficients $l \in \mathcal{R}^3$ as *lighting*.

The roles of l and ρ in equation (1) are symmetric, i.e. a reference color whose value is equal to l can produce the same color observation $\hat{\rho}$ if the lighting happens to be equal to ρ^e . This inherent ambiguity must be addressed in our problem, otherwise we can always get a trivial solution, where every point in the image is generated by the underlying texture model, and the presence of the occlusion is attributed to lighting changes. For example, leaves of a tree can be interpreted as being projected to the wall by a green light. This solution explains the image perfectly. However, it won't generate the right texture replacement effect because it misinterprets the roles of the light and reflectance. We know it is wrong because the real world lighting does not behave like a movie projector, and green light is quite unlikely[10]. This leads us to the discussion of the admissible global lighting distribution $p(l)$, defined in the domain of lighting $\mathcal{L} \subset \mathcal{R}^3$.

Suppose there are n independent light sources in a scene, each represented by $l_i, i = 1, 2, \dots, n$. Ideally the effects of these light sources are additive. The admissible lighting distribution is defined by the set $\{l | l = \sum_{i=1}^n \alpha_i l_i, \alpha_i \geq 0\}$. For instance, if there are two light sources, the possible lighting distribution will be on a line.

However, there are several factors that can result in deviations from this ideal linear model. The most noticeable effects are due to nonlinearity introduced by an imaging device, such as a digital camera. When common nonlinear processes such as gamma-correction, clipping and blooming [15] are introduced, chromaticity of a colored surface will change as its intensity varies. Secondly, interreflection between scene objects can change the lighting distribution in an unpredictable way. The situation is modeled in the radiosity and global illumination model[16]. For our case, however, only one picture of a scene is provided. It is not possible to estimate the interreflection using techniques known to us. In summary, in an uncontrolled environment the admissible lighting distribution will not be in a linear subspace. Instead of trying to accurately predict the lighting distribution, we represent and approximate them in a sampling and resampling framework[14, 9].

Now we describe the framework we adopted. We estimate lighting samples \hat{l} as will be introduced in Section 4. We keep them as 3D points in the lighting space \mathcal{L} . See Figure 1 for an illustration. When a query about $p(l)$ is requested, we estimate the sample density around l and report it. To avoid the computational burden of estimating the density at every query, we divide the lighting space \mathcal{L} into N^3 bins. Sample density in each bin can be precomputed. This technique is known as the cached statistics[13] and can achieve very high efficiency with little sacrifice in accuracy.

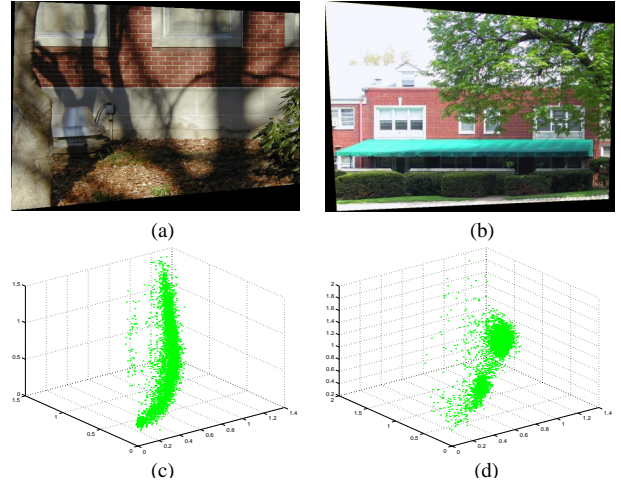


Figure 1: *Estimated lighting distributions of two scenes taken by two different cameras. (a) A scene with many soft shadows on the textured region. (b) An image whose textured regions have little soft shadows. (c) Lighting samples estimated from (a). They are distributed along a curved structure. (d) Lighting samples are concentrated on two clouds in the lighting space.*

A second distribution describes the lighting variation as a smooth process, in the sense the lighting at a pixel must be very close to at least one of its neighbors. Formally we represent this local constraints using a Markov random field (MRF) model [6]. We define the clique energy at pixel x due to lighting change as

$$c_l(x) = \min_{y \in \mathcal{N}(x)} \frac{\|l(x) - l(y)\|^2}{\sigma_l^2} \quad (2)$$

with $\mathcal{N}(x)$ describing all the four neighbors of pixel x . The probability of observing lighting l given its neighborhood is given by the Gibbs distribution $g(l|\mathcal{N}) = ke^{-c_l(x)/T}$, with k being a normalization factor and T being the current temperature. This model enables a continuous trend of lighting pattern in arbitrary directions, such as the thin shadow cast by twigs. But it discourages isolated sudden changes.

3. The Texture Model

A texture model describes how a texture pattern is produced. It is usually defined in the form of a MRF model, $p(\rho(x)|\rho_{x_1}, \rho_{x_2}, \dots, \rho_{x_n})$, with $x_i \in \mathcal{N}(x)$. It represents the possibility of observing a color vector ρ given observations of its neighbors.

Analytical form of the texture model $p(\rho(x)|\rho_{x_1}, \rho_{x_2}, \dots, \rho_{x_n})$ is not available in our problem. And it is also not possible to enumerate the space $\Omega = \mathcal{C}_{x_1} \times \mathcal{C}_{x_2} \dots \times \mathcal{C}_{x_n}$, with \mathcal{C}_{x_i} representing the

domain of colors observed at pixel x_i . A practical way to approximate this model is by texture pattern samples.

The texture model of [3] can be considered as a lookup table $f : \Omega \mapsto \mathcal{C}(x)$. Colors in the neighborhood of a pixel x define a query entry for the lookup table, indexing colors at x . Given a training image, all query entries in it is drawn and stored. This completes the training step. When evaluating $p(\rho(x)|\mathcal{N}(x))$ of a new texture pattern, we form a query key using the color in the neighborhood of x . A search is performed by comparing the query key with all the query entries. The set of closest query entries provide a set of color vectors representing color distribution at x . The search process can be very costly depending on the size of the neighborhood and the size of the texture samples. Wei and Levoy [18] improves the search speed by storing the query entry in a tree structure. In this research we study a special set of texture patterns, the near regular patterns. We show with this model a simple texture model can be defined.

Each tile of a near regular texture pattern has very similar appearance as that of a standard tile, but it is allowed to have fluctuations in color and structure. Examples of such patterns include bricks, carpet texture and they represent a large set of texture models in interior and architectural design.

The advantage of adopting a near regular texture pattern model is that we can locate each tile and conveniently define a local coordinate system for it[11]. Once the origin of a tile is determined, the color distribution of a pixel depends on the relative location u alone. Similar to the previous approaches, we define the texture model by samples. The query entry in our case is just the relative pixel location u .

In the texture replacement problem, we define lighting that illuminates the sample patch as the canonical lighting. The cached color samples thus represent the color distribution at a location u under the canonical lighting, denoted as $p^c(\rho|u)$. Assuming the color at pixel u is generated by the same texture model t that generated the sample image, the probability of observing color vector $\hat{\rho}$ under lighting l is given by

$$p(\rho|l, t, u) = p^c(\hat{\rho} ./ l | u) \quad (3)$$

Here operator $./$ is the inverse of $.*$, representing element-wise division. On the other hand, if we know color observed at u is not generated by t , the color has a uniform distribution.

$$p(\hat{\rho}|\bar{t}, u) = c \quad (4)$$

Here t and \bar{t} serve to flag the segmentation of textures.

4. Learning Statistical Models

In this section we present details of learning the texture model (3) and the admissible lighting distribution $p(l)$.

Our program asks a user to select a uniformly illuminated region. This region indicates the intention of the user, and serves as a training sample for our algorithm. A standard tile can be computed automatically using [11], or it can be selected manually. A typical tile size is between several hundred to one thousand pixels. Number of tiles in the selected region is around several dozen. To locate the rest of the tiles in a sample image, we correlate the standard tile with the whole image. Local maxima of the correlation output provides candidates for tile centers. Due to noises and sub-structures in a texture pattern, spurious peaks may occur. To prune these spurious local maxima we apply the region of dominance algorithm (ROD)[11], and it is proven by our experiences to be very robust. This procedure provides us with a set of registered sample tiles, from which the color samples at each pixel can be drawn and cached. As a result the texture model (3) is defined.

Our next goal is to estimate the admissible lighting distribution $p(l)$. As will become clear soon these statistical models provide a powerful way to identify both the texture segmentation and the lighting map.

The first step in estimating lighting samples is to register the standard tile with tiles under different lighting, such as those in the shadow. To register tiles robustly in the presence of nonlinear camera effects such as clipping and gamma-correction, we propose to use mutual information based registration technique [17]. Besides its robustness to nonlinear lighting changes, mutual information gives us a quantitative measure of similarities between two tiles, and in theory this measurement is independent of the nonlinear camera transformations. Mutual information can be considered as a criterion to match structures of two tiles. The registration is performed much like that by using correlation. We compute mutual information of the standard tile with all the image patches. After the computation local mutual information peaks are detected using a 3×3 window. Peaks with high mutual information scores signal a strong candidate for estimating light samples. Peaks with mutual information scores higher than a moderate threshold is used to initialize our segmentation and lighting estimation algorithm, which will be introduced in Section 5.

In the second step we estimate lighting samples l by comparing the standard tile with a registered tile. The estimation is given by $\hat{l} = \hat{\rho}(u) ./ \rho^c(u)$. It is valid given that we are working on near regular patterns. We preclude possible outliers by robust estimation techniques[12]. For example, we estimate a median and a scale factor for each channel (R,G, or B) of the l samples. Samples outside certain distance to the median is considered as outliers and is not used to estimate the lighting distribution.

Finally, the samples l is used to estimate $p(l)$, represented in cached statistics as described in Section 2.

5. A Bayesian Framework and the Stochastic Annealing Solution

Mutual information is a region based method and it can only give us very coarse texture segmentation. Texture replacement, however, requires pixel wise segmentation and lighting estimation. It is necessary to design an algorithm based on the color observation at each pixel.

We incorporate the learned knowledge about the admissible lighting distribution $p(l)$ and the texture model (3), into a Bayesian framework. Formally we are interested in solving the *maximum a posteriori* (MAP) problem,

$$[\hat{t}, \hat{l}] = \underset{t, l}{\operatorname{argmax}} p(t, l | \hat{\rho}, u) \quad (5)$$

Applying Bayes rule,

$$\begin{aligned} & p(t, l | \hat{\rho}, u) \\ \propto & p(\hat{\rho} | t, l, u) p(t, l | u) \\ = & p(\hat{\rho} | t, l, u) p(l | u) p(t | u) \end{aligned} \quad (6)$$

Here we assume independence of t and l .

The lighting distribution $p(l | u)$ at position u is determined by two factors, namely, the global admissible lighting distribution, and the smoothness constraint imposed by the Markov random field model(2). We define $p(l | u) = p(l)g(l | \mathcal{N})$.

To encourage connectivity of both the textured and non-textured regions, we impose the final constraints via $p(t | u)$. Denote n_1 as the number of neighbors that has the same labeling as u , and denote the total numbers of neighbors as n_2 , we define $p(t | u) = a \frac{n_1}{n_2} + b$, with $a > b > 0$ and $a + b < 1$.

We solve the MAP problem (5) using the Markov Chain Monte Carlo (MCMC) method[14]. To apply MCMC efficiently a good initialization is important. In our case the computed mutual information is reused for initialization. Recall we detected all local peaks using a small window. By setting an appropriate threshold, some of the peaks can be discarded. The remaining peaks define a set of tile centers. We define one candidate tile on each of the peak locations. All the pixels covered by a candidate tile is labeled as t , and others are labeled as \bar{t} . Thus a very coarse segmentation is obtained. Lighting at location u is estimated as $\hat{\rho}(u) / \rho^c(u)$, with $\rho^c(u)$ denoting the color of the standard tile at relative position u . In case multiple candidate tiles cover a single pixel u , the lighting estimation given by the tile with the highest mutual information score is accepted.

The thresholds, including the high mutual information score (for estimating $p(l)$), moderate mutual information score (to initialize the MCMC algorithm) and the constant probability $p(\hat{\rho} | \bar{t})$ are manually set. Given this is a graphics

application, we relax the requirement for theoretical analysis of such thresholds and obtain them through user interaction.

Starting from the initialization, the algorithm updates each candidate tile in turn. The algorithm makes a proposal $[t, l]$ or \bar{t} . The *a posteriori* probability of the new proposal is computed using (6). Denote the *a posteriori* probability of the new proposal as p_n , and that of the current estimation as p_o . We accept the new proposal if $\log(p_n) - \log(p_o) < T \log(r)$, where r is a random number drawn from a uniform distribution in $(0, 1]$, and T the current “temperature”. Initially the temperature is high and is decreased at each step. The sampling process is illustrated by a small patch as shown in Figure 2. At the first several steps the initialization error is alleviated. And as temperature cools down only the ideal segmentation is kept.

6. Texture Replacement Results

A sample output of our algorithm is shown in Figure 4. (Please visit http://www.cs.cmu.edu/~ytsin/research/texture_replacement/ for more results). The result is visually pleasing. Our algorithm generated a correct texture segmentation and a realistic lighting map. When replaced with new texture patterns, occlusion and shadowing effects are preserved. Notice the pink pipe on the left side of the image. Our algorithm detected the region as a non-textured pattern even though it has a color close to that of the original brick pattern. This confirms the accuracy of our statistical models $p(l)$ around that area. However, we notice there exist artifacts in the output. Part of the pink pipe in the right side of the image is misclassified as texture region. This is due to the low signal to noise ratio in the shadow. Furthermore, there are misdetections (Figure 3a) and false alarms (Figure3(b)). The misdetection is due to the fact we didn’t choose a candidate tile at the heavily occluded region. And the false alarm is largely due to a fake candidate tile and color similarity between pixels in the image and the standard tile. So we need a mechanism to determine birth of a new candidate tile and death of a current tile, much in the spirit of the stochastic annealing framework. We leave this as a future research. Despite all these artifacts, which mainly occur in dark areas and highly cluttered areas, we consider the replacement result in Figure 4 satisfactory.

7. Conclusions

We defined the real image texture replacement problem for near regular planar patterns, and proposed a statistical algorithm to solve it. Statistical information on lighting distribution, and the texture model, is systematically learned from a sample image and tiles in the entire image. A Markov Chain Monte Carlo algorithm is adopted to solve the texture seg-

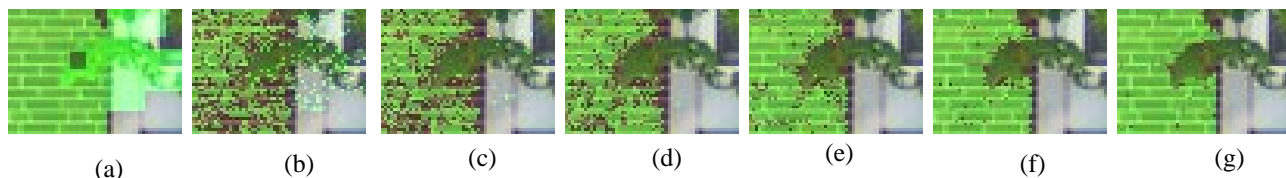


Figure 2: The process of stochastic annealing. Temperatures from left to right are 2.000, 1.843, 1.699, 1.443, 1.041, 0.542, 0.147, respectively. Detected texture pixels are displayed by superimposing a light green mask.

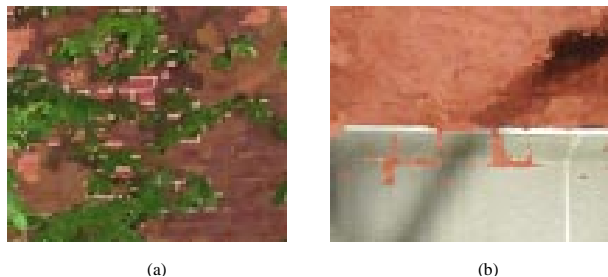


Figure 3: Known artifacts with current approach. (a) Mis-detection. (b) False alarm

mentation problem and the color constancy problem. Visually satisfactory results have been achieved.

Acknowledgments

This research is partially supported by an NSF award #IIS-0099597. We thank Professor Steven Seitz for his initial suggestion of the problem. And we thank Jonathan Rowlett for his initial effort on the project.

References

- [1] P. Andrey and P. Tarroux. Unsupervised segmentation of markov random field modeled textured images using selectionist relaxation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(3):252–262, March 1998.
- [2] D.H. Brainard and W.T. Freeman. Bayesian color constancy. *J. Opt. Soc. Amer.-A*, 14(7):1393–1411, July 1997.
- [3] A. Efros and Leung T. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision*, pages 1033–1038, 1999.
- [4] G. Finlayson, M. Drew, and Funt B. Diagonal transforms suffice for color constancy. In *IEEE International Conference on Computer Vision*, pages 163–171, 1993.
- [5] D.A. Forsyth. Sampling, resampling and colour constancy. In *IEEE Computer Vision and Pattern Recognition*, pages 1:300–305, 1999.
- [6] D Geman and Geman S. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
- [7] G. Healey and L. Wang. The illumination-invariant recognition of color texture. In *IEEE International Conference on Computer Vision*, pages 128–133, 1995.
- [8] T. Hofmann, J. Puzicha, and J. Bulmann. Unsupervised texture segmentation in a deterministic annealing framework. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(8):803–818, August 1998.
- [9] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Intern. J. Comp. Vis.*, 29:5–28, 1998.
- [10] D. B. Judd, D. L. MacAdam, and G. Wyszecki. Spectral distribution of typical daylight as a function of correlated color temperature. *J. Opt. Soc. Amer.*, 54:1031–1040, 1964.
- [11] L. Liu and Collins R. A computational model for repeated pattern perception using frieze and wallpaper groups. In *IEEE Computer Vision and Pattern Recognition*, 2000.
- [12] P. Meer. Robust techniques for computer vision (tutorial). In *IEEE Computer Vision and Pattern Recognition*, June 1997.
- [13] A. Moore and M. Lee. Cached sufficient statistics for efficient machine learning with large datasets. *Journal of Artificial Intelligence Research*, 8:67–91, 1998.
- [14] R.M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical Report CRGTR-93-1, Computer Science, Uni. Toronto, 1993.
- [15] C.L. Novak, S.A. Shafer, and R.G. Wilson. Obtaining accurate color images for machine vision research. In *Proc. SPIE Conf. Perceiving, Measuring and Using Color*, 1990.
- [16] F. Sillion and C. Puech. *Radiosity and Global Illumination*. Morgan Kaufmann Publishers, 1994.
- [17] P. Viola. Alignment by maximization of mutual information. In *IEEE International Conference on Computer Vision*, pages 16–23, 1995.
- [18] L. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH 2000, Computer Graphics Proceedings*, pages 479–488, 2000.



(a)



(b)

Figure 4: The final replacement results. (a) The original images. (b) The texture replaced image.