Probabilistic Principle Component Analysis on Time Lapse images

Yue Gao

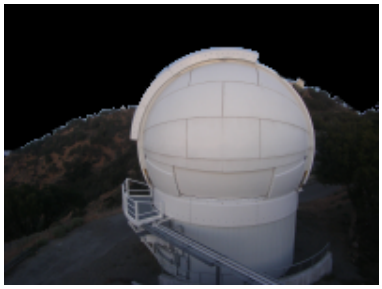
December 13, 2010

1 Introduction

Time-lapse photography, in which images are captured at a lower rate than that at which they will ultimately be played back. Classic time-lapse photography subjects are scenes. Today, most of those image sequences are collected by the thousands of Internet cameras. They typically provide outdoor views of cities, construction sites, traffic, the weather, or natural phenomena. However, Time-lapse photography can create an overwhelming amount of data. Image compression reduces the storage requirements, but the resulting data has compression artifacts and is not very useful for further analysis [9]. In addition, it is currently difficult to edit the images in a time-lapse sequence .

A key challenge in dealing with time-lapse data is to provide a representation that efficiently reduces storage requirements while allowing useful scene analysis and advanced image editing. In our project, given a sequence of images, we want to be able to model the scene with few parameters. We will focus on time-lapse image sequences of outdoor scenes under clear-sky conditions[6]. The camera viewpoint is fixed and the scene is mostly stationary, hence the predominant changes in the sequence are changes in illumination. As a preprocessing procedure, We would use a sky mask to preprocess the images. With clear sky assumption, we would extract the sky from every image show in Figure 1(b) using GIMP.

The road map for the report is that we first start with some background clarification about the properties of outdoor scenes. Then I will explain how to use Maximum Likelihood Estimation(MLE) to compute intrinsic image and illumination images. Furthermore, we will use those information to find the shadow map of every single image and if a point is in shadow, it is considered as a missing value. Therefore, using Expectation Maximization to solve Probabilistic PCA with missing data will give us a compacted representation of the dataset. I will also demonstrate how I can use Markov Random Field to smooth out the shadow region at the end.



(a) A single image from our dataset

(b) A sky mask model from one of our dataset

Figure 1: Sky mask

2 Background and Comparison Model

Intrinsic images are a useful mid-level description of scenes proposed by Barrow and Tenenbaum [1]. An image is decomposed into two images: a reflectance image and an illumination image as shown in figure 2. Finding such a decomposition remains a difficult problem in computer vision. Here we focus on a

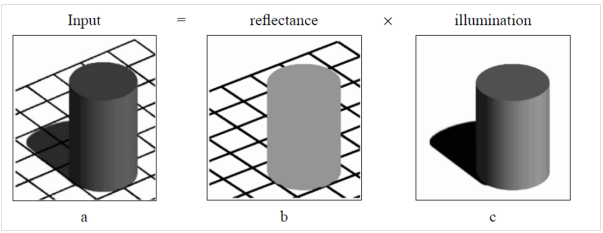


Figure 2: An image is decomposed into two images: a reflectance image and an illumination image

slightly easier problem: given a sequence of images where the reflectance is constant and the illumination changes, we cam recover illumination images and a single reflectance image. We will denote the input by $I(x, y)$ the input image and use $R(x, y)$ the reflectance image and $L(x, y)$ the illumination image and $I(x, y) = L(x, y)R(x, y)$

We would compare our result with the model being proposed in [9] and [8]. Their method is to convert time-lapse photography captured with outdoor cameras into Factored Time-Lapse Video (FTLV): a video in which time appears to move faster and where data at each pixel has been factored into shadow, illumination, and reflectance components. The factorization allows a user to easily relight the scene, recover a portion of the scene geometry (normals), and to perform advanced image editing operations. The key formulation for their approach is the following

$$\begin{aligned} F(t) &= I_{sky}(t) + S_{sun}(t) * I_{sun}(t) \\ I_{sky}(t) &= W_{sky} H_{sky}(t) \\ I_{sun}(t) &= W_{sun} H_{sun}(t + \theta) \end{aligned}$$

where H represent a basis curve scaled by per pixel weight W. the term $S_{sun}(t)$ is a term indicating shadows. They used basis curves describing the changes of intensity over time, together with per-pixel offsets and scales of these basis curves, which capture spatial variation of reflectance and geometry

3 Reflectance and Illumination Image Extraction

Given a sequence of T images, we want to solve for $L(x, y, t)$ and $R(x, y, t)$. However, we need to add some more constrains. Since we have a sequence of images, we would constrained the problem of solving one reflectance image as constant over time and only the illumination image changes. Another constrain is that we use a prior that assumes that illumination images will give rise to sparse filter outputs applied to L will tend to be sparse. We derive the ML estimator under this assumption and show that it gives a simple algorithm for recovering reflectance. Figure 3 illustrates this fact: the image of the outdoor scene has a histogram distribution that are peaked at zero and fall off much faster than a Gaussian. This property is robust enough that it continues to hold if we apply a pixel wise log function to each image. These prototypical histograms can be well fit by a Laplacian distribution as shown in Figure 3 as $P(x) = \frac{1}{Z} \exp^{-a|x|}$.

We will therefore assume that when derivative filters are applied to $L(x, y, t)$ the resulting filter outputs are sparse: more exactly, we will assume the filter outputs are independent over space and time and have a Laplacian density. Assume we have N filters $\{f_n\}$ we denote the filter outputs by $o_n(x, y, t) = I * f_n$. then $r_n = R * f_n$ denotes the reflectance image filtered by the nth filter. Assume filter outputs applied to $L(x, y, t)$ are Laplacian distributed and independent over space and time. Then the ML estimate of the filtered reflectance image $\hat{r_n}$ are given by $\hat{r_n}(x, y) = median_t o_n(x, y, t)$. This is derived by assuming Laplacian densities and independence which gives

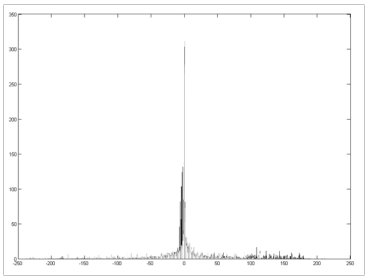
$$\begin{aligned} P(o_n|r_n) &= \frac{1}{Z} \prod_{x,y,t} e^{-\beta |o_n(x,y,t) - r_n(x,y,t)} \\ &= \frac{1}{Z} e^{-\beta \sum_{x,y,t} |o_n(x,y,t) - r_n(x,y,t)} \end{aligned}$$

Maximizing the likelihood is equivalent to minimizing the sum of absolute deviations from $o_n(x, y, t)$. Therefore, applying ML, we achieved the filtered reflectance image $\hat{r_n}$. To recover r, the estimated reflectance function, I will use the formulation derived in [10]. Once we computed the estimated $R(x, y)$, then in log space, $\log(L(x, y, t)) = \log(I(x, y, t)) - \log(R(x, y))$

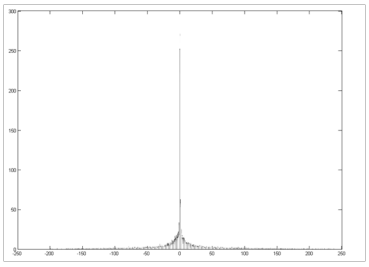
4 Probabilistic PCA with Missing Values

Once we achieved the $L(x, y, , t)$, I used a threshold value t to determine if a pixel is in shadow or not. If the intensity of $L(x, y, t) < t$ then I say this pixel is in shadow otherwise it is not. Then on my origi-

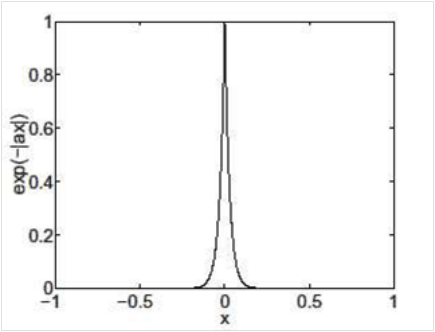
(a) Original Image



(b) histograms of horizontal derivative filter outputs



(c) histograms of vertical derivative filter outputs



(d) A Laplacian distribution

Figure 3: We use a prior motivated by the statistics of natural scenes

nal data set I, pixel $I(x, y, t) = NaN$ which means I considered this pixel is missing value. According to the [5] Shashua proved that three images are sufficient to span the full range of images of a Lambertian scene rendered under distant lighting and a fixed viewpoints. Now I want to compute the top principle component of the sequence of images $I(x, y, t$ with missing values, I will use Probabilistic PCA to solve it [3],[7],[4].

Let $y \in R^D$ denote a data vector where $D = mxn$, m and n are the size of the $I(x, y)$ input image. $x \in R^d$ denotes the vector of the principle component coordinates. Here, we will only use the top 3 component, so $d = 3$. We let

$$p(x) = \mathcal{N}(x; 0, I)$$
$$p(y|x) = \mathcal{N}(y; C^T x, \sigma^2 I)$$

Where C is a dxD matrix with the projection vectors form the principal componeet coordinates to the data coordinates. The conditional on x given y is then given by

$$p(x|y) = \mathcal{N}(x; \mu, \Sigma)$$
$$\mu = \sigma^{-2}\Sigma C y$$
$$\Sigma^{-1} = I = \sigma^(- 2)CC^T$$

When only a subset of the coordinates of y is observed, we replace C C above with the C_o which has only the columns corresponding to the observed values, and similar for y which is replaced by the observed part y_o.

Our goal is now to find the parameters C and that maximize the likelihood of some observed data: vectors y that are fully or partially observed. To do so, we use an EM algorithm that estimates in the E-step the missing values: the vectors x and the missing parts of the y which we denote by y_h. In the M-step we fix these estimates, and maximize the expected joint log-likelihood of x and y. For simplicity we assume that the distribution over x and y_h factors so that we write a lower-bound on the data log-likelihood as

$$\begin{aligned}\log p(y_o) &= \log p(y_o) - D(q(x)q(y_h)||p(x, y_h|y_o)) \\ &= H(q(x) + H(q(y_h)) + \mathbf{E}_q[\log p(x) + \log p(y|x)]\end{aligned}$$

where $H = \frac{1}{2}\log|\Sigma|$. Now we can maximize this bound. In the E-step with respect to the distributions q, and in the M-step with respect to the parameters.

In the E-step

$$q(y_h) = \exp \int q(x) \log p(z_h|x) = \mathcal{N}(y_h; C_h^T \overline{x}, \sigma^2 I)$$

$$q(x) = p(x|y_o) exp \int q(y_h) log p(y_h|x) = \mathcal{N}(x; \sigma^{-2} \Sigma C \overline{y}, \Sigma)$$

where $\bar{y}$ is the mean of $q(y_h)$ for the missing values and y_o for the observed part, and $\bar{x}$ is the mean of $q(x)$.

In the M-step The expectation can be summed over N data, we can write it as

$$\mathbf{E}_q[\log p(x) + \log p(y|x)] =$$

$$-\frac{ND}{2} \log \sigma^2 - \frac{1}{2\sigma^2} (\sum_n ||\bar{y_n} - C^T \bar{x_n}||^2 - Tr\{C^T \Sigma C\})$$

$$-\frac{D_h}{2\sigma^2} \sigma_{old}^2 - \frac{1}{2} \sum_n ||\bar{n_n}||^2 - \frac{N}{2} Tr\{\Sigma\}$$

where D_h denotes the total number of missing values, and σ_{old} is the current value of σ that was used in the E-step to compute the q. Maximizing this over C and σ we get

$$C = (N\Sigma + \bar{X}\bar{X}^T)^- 1 \bar{X} \bar{Y}^T$$

$$\sigma^2 = \frac{1}{ND} (XTr\{C^T \Sigma C\} + \sum_n ||\bar{y_n} - C^T \bar{x_n}|| + D_h \sigma_{old}^2)$$

where $\bar{X}$ and $\bar{Y}$ denote matrices that collect all $\bar{x}$ and $\bar{y}$ columns.

5 Result

I have two datasets, one is a scene with a telescope and the other is a far view of University of Arizona. Both dataset satisfy the requirement of time lapse images where the camera is fixed still and images are take every 10 minutes.

In figure 4 and figure 5, we can see a sequence of six outdoor images. The left upper corner image is the original image, the right upper image is the reflectance image which does not change over time. The left lower image is the illumination image and the right corner image is the reconstructed image by multiplying $R(x, y)xL(x, , t)$ The result illumination image is much more helpful in determine the shadow image.

In figure 6, we can see the same six outdoor scene images. The upper image is the original image, the

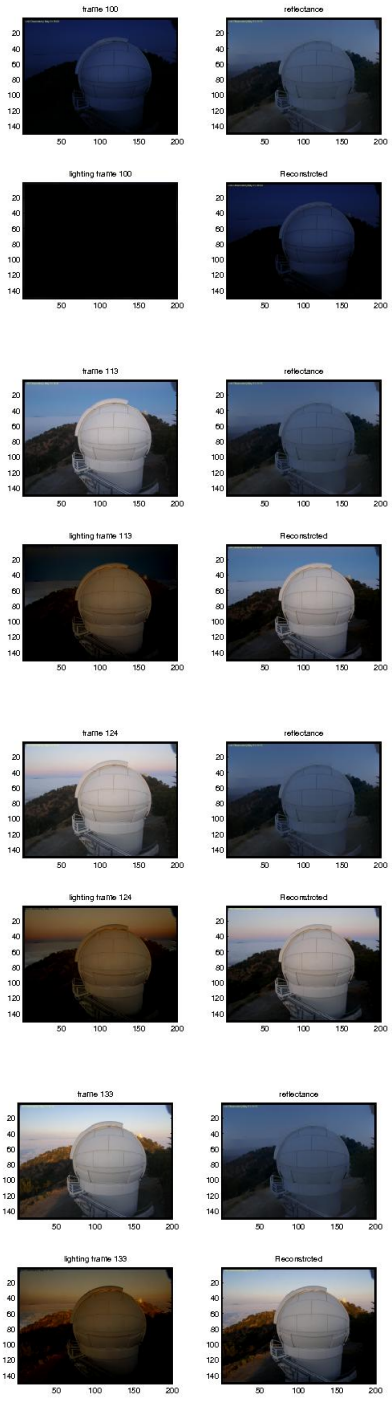


Figure 4: Four Images from the telescope dataset

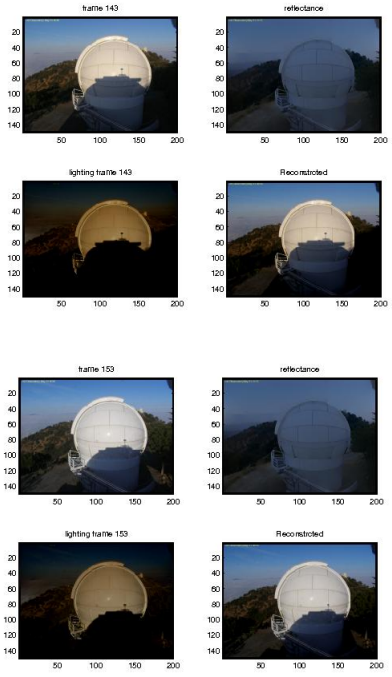


Figure 5: Two more images from the telescope dataset.

black and write image is the shadow map where black meaning it is in shadow and white meaning it is not in shadow. (The result here shown has not applied by the sky mask yet) Then the third image is the reconstructed using the top three principle components. We we can see, the region that is in the shadow will become "NaN" and they are filled in with appropriate images. In order to reconstruct the RGB color image, I computed each channel separately.

6 Discussion

The result seems to be on the right track. However there are some unsolved issues.

1. The reconstructed color image has some weird color effects. For example, there are some very green region or very red region. This might due to the fact that I computed the RGB channel independent of each other. I might need to model the correlation between the three channels to avoid noise like that.

2. Just using threshold might not be a smart idea.

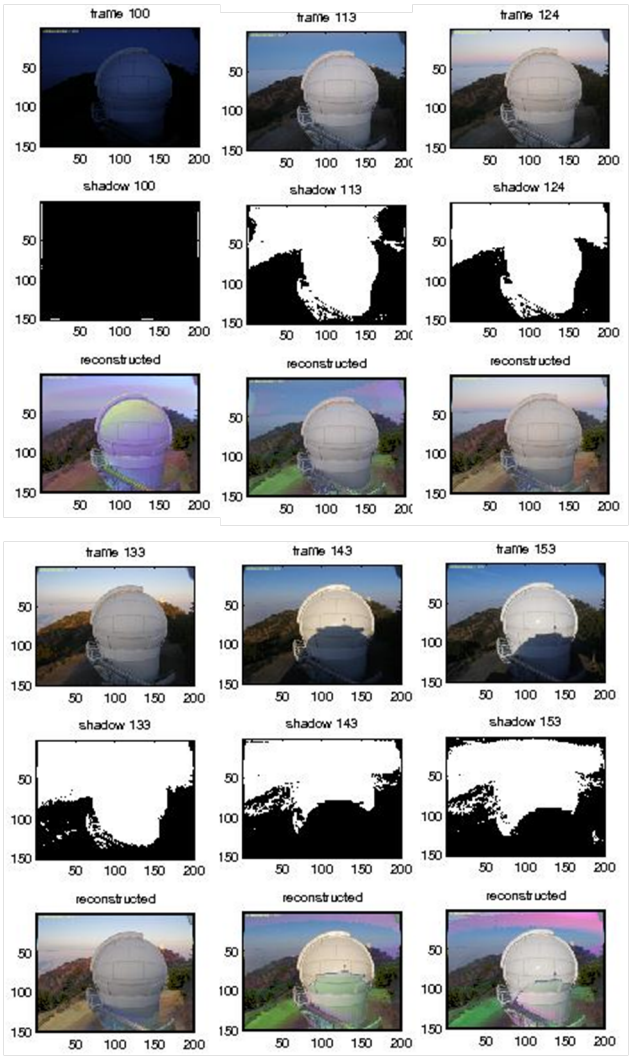


Figure 6: We use a prior motivated by the statistics of natural scenes

In figure 7, As we can see, that when the telescope body is mostly in shadow, the shadow map does not indicate so. Therefore, I need some better decision to determine if a point is in shadow or not.

3. The last issue i noticed is that the edge on the shadow boundary is not smooth enough. That means I need some edge smoothing procedure to transition the change for the filled in region.

4. The boundary of the shadow is still not smooth enough

7 Markov Random Field

To solve the problem discussed above, we can naturally construct our model to a Markov Random Field model. MRF has been used widely in the computer vision field [1]. I will first propose the model based on comparison model [2] where a scene is consist of $F(t) = I_{sky}(t) + S_{sun}(t) * I_{sun}(t)$. Let $I = \{I_{ij}\}$ denote the observed image, with $I_{ij} \in \{0, 255\}$ representing the pixel at row i and column j in image I. Assume the image has dimensions N x M, so that $1 \leq i \leq N$ and $1 \leq j \leq M$. In addition, we also have an extra parameter t indicating the temporal change. Therefore, to represent a pixel , we use the notation $I_{ij,t}$. We have a set of latent variables $X = \{x_{ij,t}\}$ represents the true image, with $x_{ij,t} \in \{0, 255\}$ indicating the value of $I_{ij,t}$ with our re-construction. , we also know that a single image contains environment and sun lighting. Therefore, our label x can be represented as

$$X(t) = I_{sky} + \begin{pmatrix} c_1 \\ c_2 \\ c_3 i \end{pmatrix} \begin{pmatrix} PC_1 \\ PC_2 \\ PC_3 \end{pmatrix} * S_{sun}(t)$$

For every pixel at a given time t

$$x_{i,j,t} = I_{sky}(i,j,t) + \begin{pmatrix} c_1 \\ c_2 \\ c_3 i \end{pmatrix} \begin{pmatrix} PC_1(i,j) \\ PC_2(i,j) \\ PC_3(i,j) \end{pmatrix} * S_{sun}(i,j,t)$$

As shown in Figure in 7, each (internal) $x_{ij,t}$ is linked with four immediate neighbors in a single image and also linked with the same pixel before the frame and the pixel after the frame, denoted as $x_{i-1,j,t}, x_{i+1,j,t}, x_{i,j-1,t}, x_{i,j+1,t} x_{i,j,t-1}, x_{i,j,t+1}$, which together are denoted $x_{N(i,j,t)}$. Pixels at the borders of the image (with $i \in 1, N$ or $j \in 1, M$) also

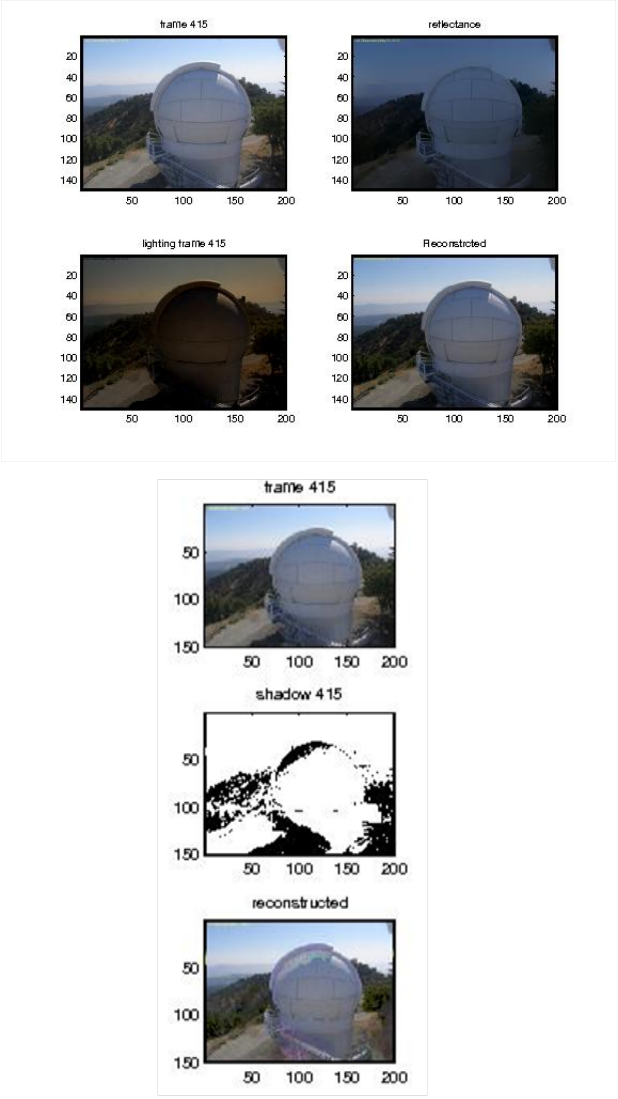


Figure 7: We use a prior motivated by the statistics of natural scenes

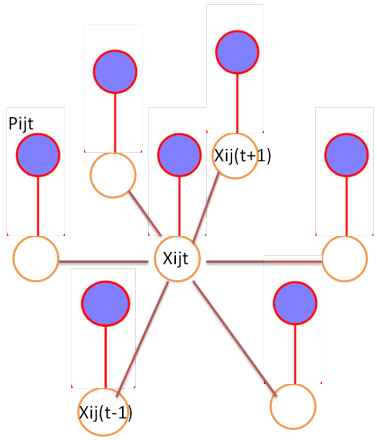


Figure 8: MRF model for a pixel $x_{ij,t}$ and its Neighbors

have neighbors denoted $x_{N(i,j,t)}$, but these sets are reduced in certain way.

This construction have to types of cliques. First type is associate with $\{x_{i,j,t}, I_{i,j,t}\}$. We choose an energy function to model this term:
For a given $x_{i,j,t}$

$$E_{X,I} = \sum_{i,j,t} ||x_{i,j,t} - I_{i,j,t}||^2$$

$$E_{x_{i,j,t}} = ||I_{sky}(i,j,t) + \begin{pmatrix} c_1 \\ c_2 \\ c_3 i \end{pmatrix} \begin{pmatrix} PC_1(i,j) \\ PC_2(i,j) \\ PC_3(i,j) \end{pmatrix} * S_{sun}(i,j,t) - I_{i,j,t}||$$

$$P(X,I) = \frac{1}{Z} \exp\left(-E_{X,I}\right)$$

The second type of clique is $\{x_i, x_j\}$ where i and j are the indices of neighboring pixels in $x_{N(i,j,t)}$, therefore, the energy function means adding smoothness to the labeling which can be represented as

$$E_{x_{N(i,j,t)}} = \sum_{p,q \in x_{N(i,j,t)}} w_{pq} ||S_{sun}(p) - S_{sun}(q)||^2$$

Therefore, In our energy function, we have to solve for the following terms:

$$\begin{aligned} I_{sky} &= \text{A basis image for the sky term} \\ PC_{1,2,3} &= \text{The first,second and third principle component for the sun term} \\ c_{1,2,3} &= \text{The first three coefficient principle component corresponding to it PC} \\ S_{sun}(t) &= \text{The shadow matrix with respect to the change in time t)} \end{aligned}$$

$$E(I,X,C,N) = \eta \sum_{X,I} E_{X,I} + \sum_{i,j,t} \beta E_{x_{N(i,j,t)}}$$

However, I was not able to finish this part of the project for now. However, I still think this is a novel proposal and worth trying.

8 Conclusion

Time lapse Images are a new source of data and to use those data to understand the scene is an interesting problem . My goal is to look for a compact, intuitive and factored representation for time lapse sequences that separate a scene into its reflectance, illumination, and geometry factors. Therefore, I think this preliminary approach enable a number of more general outdoor scene modeling such as different weather condition. Also this problem correlated with applications such as shadow removal, relighting, advanced image editing, and painterly rendering.

References

[1] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of markov random fields for segmentation of 3d scan data. 2005.

[2] E.P. Bennett and L. McMillan. Computational time-lapse video. In *ACM SIGGRAPH 2007 papers*, page 102. ACM, 2007.

[3] M. Brand. Incremental singular value decomposition of uncertain data with missing values. *Computer VisionECCV 2002*, pages 707–720, 2002.

[4] F. De la Torre and M.J. Black. Robust principal component analysis for computer vision.

In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 362–369. IEEE, 2002.

[5] R. Garg, H. Du, S.M. Seitz, and N. Snavely. The dimensionality of scene appearance. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1917–1924. IEEE, 2010.

[6] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007.

[7] H.Y. Shum, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(9):854–867, 2002.

[8] K. Sunkavalli, F. Romeiro, W. Matusik, T. Zickler, and H. Pfister. What do color changes reveal about an outdoor scene? In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[9] Kalyan Sunkavalli, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. Factored time-lapse video. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), August 2007.

[10] Y. Weiss. Deriving intrinsic images from image sequences. 2001.