Application of Dictionary Methods in Denoising of Uncalibrated Photometric Stereo

Zhe Du
Electrical and Computer Engineering
University of Michigan
Ann Arbor, Michigan
Email: zhedu@umich.edu

Fengxing Zhu
Electrical and Computer Engineering
University of Michigan
Ann Arbor, Michigan
Email: fengxing@umich.edu

***Abstract*—Uncalibrated photometric stereo is an ill-posed inverse problem and usually difficult to solve. Many literatures have shown some nice results of solving uncalibrated photometric stereo using some assumptions. However, through our experiments, if the images are perturbed by some randomness like noise and missing values, the performance will degrade greatly. Our project focuses on the implementation of applying dictionary learning denoising method (SOUP-DIL) to uncalibrated photometric stereo to handle perturbations. By comparing with several other denoising methods, SOUP-DIL has shown satisfactory results even under noise with rather high variance.**

I. Introduction

The basic idea of photometric stereo is that if we take a sequence of photos of an object from a fixed viewpoint and the light directions and intensities are known, we can reconstruct its surface. The number of photos should be greater than three with non-coplanar light directions. To achieve the best result, shadows and specularity highlights should be avoided. When light directions are unknown, the reconstruction becomes an ill-posed problem. Fortunately, with integrability constraint, we can solve this problem up to a linear transform with three unknown parameters, which is also called GBR ambiguity. With some further assumption, like minimum total variation (TV) [1], minimum albedo entropy [2], availability to diffuse maxima [3], etc., GBR ambiguity can be resolved by solving some optimization problem. And experiments showed they all can give satisfactory results [1].

Using the TV-minimization method in [1], we find that if the photos are perturbed by Gaussian noise or missing values, the surface cannot be reconstructed correctly. Specifically, missing values in images tend to make surface uneven and rugged and surface can become totally unrecognizable with as few as 10% missing pixel values. When noise has a large variance, some spikes would appear the surface with height even hundred times as much as original height.

To deal with perturbations, one intuition is to apply image denoising method to photos and then reapply uncalibrated photometric stereo method to recovered images. Natural signals and images have sparse representation in some transformed domain, which could be exploited in several applications such as compression, denoising and inverse problem. Recently, dictionary learning model based on data has shown some advantages in many applications compared with fixed or analytic models. So, in our project we investigated three methods: SVD, fixed dictionary (Discrete Cosine Transform) denoising and adaptive dictionary denoising (SOUP-DIL). Rather than using metric PSNR commonly used in image denoising, we used mean angular error (MAE) of normal vectors of surface to evaluate their performances. Our experiment results show that dictionary methods outperform non-dictionary method SVD and adaptive dictionary outperforms fixed dictionary.

The rest of report is organized as follows: in Section II, we introduce classical calibrated photometric stereo and TV-minimization uncalibrated photometric stereo. In Section III, we introduce a dictionary learning method (SOUP-DIL). Section IV shows how perturbation can influence surface reconstruction. Section V discusses the implementation of three image denoising methods in photometric stereo. Experiments evaluating the performances of denoising schemes and results are in Section VI.

II. Uncalibrated Photometric Stereo

A. Calibrated Photometric Stereo

Classical calibrated photometric stereo [4] is based on Lambert Cosine Law. The intensity at a surface point is proportional to the inner product between light vector and normal vector of surface at that point. If we take n photos each with m pixels from a fixed view point, we have

$$I_{ij} = \alpha_i N_i^T L_j \tag{1}$$

of which I_{ij} is the intensity at the ith pixel in jth photo (or under jth lighting condition), α_i and N_i are the albedo and normal vector at pixel i (here an image is represented as a $n \times 1$ vector) and L_j is the jth light vector, the norm of which is the light intensity. One thing to note is that we assume the light source is far away enough so that the directions of light shining on every pixel can be considered as the same.

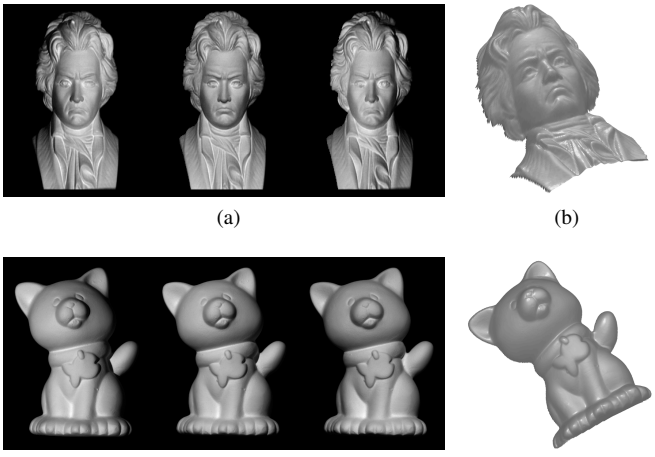
Let $M_i = \alpha_i N_i$, we could have the matrix form of Lamber Law

$$I = M^T L \tag{2}$$

of which $I \in \mathbb{R}^{m \times n}$, $M \in \mathbb{R}^{3 \times m}$ and $L \in \mathbb{R}^{3 \times n}$. If the lighting conditions L is known, we could solve M using least squares. Then, N_i is obtained by normalizing each M_i

$$N_i = \frac{M_i}{\| M_i \|} \tag{3}$$

After we obtain all the normal vector N_i, we could obtain the height map $f(x, y)$ at pixel (x, y) using some integration method. Fig 1 shows two examples of applying photometric stereo to Beethoven and Cat datasets. We can see that using photometric stereo, the surfaces can be perfectly reconstructed.



(a) (b) (c) (d)

Fig. 1. Photos and Reconstructed Surface Using Photometric Stereo: (a)(b) Beethoven Dataset; (c)(d) Cat Dataset

B. Uncalibrated Photometric Stereo

If lighting conditions are unknown, photometric stereo becomes an ill-posed problem. Given one set of solution $\{M_0, L_0\}$ satisfying $I = M_0^T L_0$, any invertible 3×3 matrix A gives another solution $\{M', L'\} = \{A^{-T} M_0, A L_0\}$ such that $I = {M'}^T L'$.

1) GBR Ambiguity: Although A has 9 degree freedom, with the assumption that surface is twice differentiable we can reduce the freedom to 3 [8]. From notation in Section II-A, we have

$$\frac{M_x}{M_z} = \frac{N_x}{N_z} = -\frac{\partial f(x,y)}{\partial x} \quad \frac{M_y}{M_z} = \frac{N_y}{N_z} = -\frac{\partial f(x,y)}{\partial y} \tag{4}$$

And twice differentiability $\frac{\partial}{\partial y}(\frac{\partial f(x,y)}{\partial x}) = \frac{\partial}{\partial x}(\frac{\partial f(x,y)}{\partial y})$ gives

$$\frac{\partial}{\partial y}\left(\frac{M_x}{M_z}\right) = \frac{\partial}{\partial x}\left(\frac{M_y}{M_z}\right) \tag{5}$$

The twice differentiable assumption is also called integrability constraint.

Under this constraint, if $\{M_0, L_0\}$ satisfies $I = M_0^T L_0$, then another set of solution must have the form $\{G^{-T} M_0, G L_0\}$ of which G is parameterized by $\{\mu, \nu, \lambda\}$:

$$G(\mu, \nu, \lambda) = \begin{bmatrix} 1 & & \\ & 1 & \\ \mu & \nu & \lambda \end{bmatrix} \tag{6}$$

The ambiguity represented by G is called *generalized bas-relief* (GBR) ambiguity. Let M^* denote the true M, $\{G^{-T} M^*$ corresponds to a deformed surface that has a different height scaling and skewness with respect to the true surface. The idea behind GBR ambiguity is that a deformed surface $\{G^{-T} M^*$ can even generate correct image I_j, if the lighting is $G L_j$. The name GBR ambiguity originated from the fact what we see from bas-relief looks the same as real world objects under some lighting conditions even though they are carved in a flattened way.

2) TV-minimization Uncalibrated Photometric Stereo: [1] has empirically shown that given the solution space $\{G^{-T} M_0, G L_0\}$, only the true solution $\{M^*, L^*\}$ has the smallest total variation of M. Yvain [1] proposed the following problem

$$\begin{cases} M^* = \arg\min\limits_{M} \ TV(M) \\ s.t. \begin{cases} I = M^T L \\ \frac{\partial}{\partial y}\left(\frac{M_x}{M_z}\right) = \frac{\partial}{\partial x}\left(\frac{M_y}{M_z}\right) \quad \forall x, y \end{cases} \end{cases} \tag{7}$$

Use the method in [8], we could easily find an initial solution $\{M_0, L_0\}$ satisfying the two constraints in (7), so problem (7) reduces to

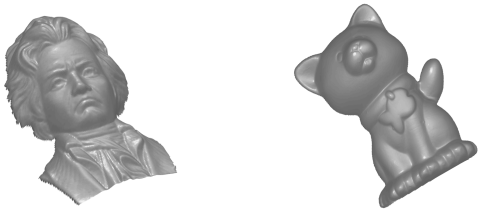
$$\begin{cases} \{\mu^*, \nu^*, \lambda^*\} = \arg\min\limits_{\{\mu,\nu,\lambda\}} \ TV(G(\mu, \nu, \lambda)^{-T} M_o) \\ s.t. \ \lambda > 0 \end{cases} \tag{8}$$

Since the minimum total variation constraint can only uniquely reconstruct the surface up to the sign of λ, which is also convex/concave ambiguity, so we restrict the sign of λ to be positive. Problem (8) is a convex problem with respect to $\{\mu, \nu, \lambda\}$, so we could solve for $\{\mu^*, \nu^*, \lambda^*\}$ using gradient search. After we obtain $\{\mu^*, \nu^*, \lambda^*\}$, M^* is given by

$$M^* = G(\mu^*, \nu^*, \lambda^*)^{-T} M_0 \tag{9}$$

Then the normal vectors and height map could be reconstructed easily.

Fig. 2 shows the surfaces of Beethoven and cat reconstructed with TV-minimization uncalibrated photometric stereo. They look almost the same as surfaces in Fig. 1. One small distinction is that surface of Beethoven in Fig. 2 is a little flattened compared with Fig. 1.



(a) (b)

Fig. 2. Surfaces Reconstructed with TV-minimization Uncalibrated Photometric Stereo: (a) Beethoven Dataset; (b) Cat Dataset

III. Dictionary Learning—SOUP-DIL

The sum of outer product dictionary learning (SOUP-DIL) method is mainly based on [5]. According to a classic synthesis model,a signal $y \in \mathbb{R}^n$ is approximately a linear combination of a small subset of atoms or columns of a

dictionary $D \in \mathbb{R}^{n\times J}$, i.e. $y = Dx+\epsilon$ with sparse $x \in \mathbb{R}^J$, and ϵ is assumed to be a small modeling error or approximation error in the signal domain.

For a given signal y and dictionary D, the following problem is to be solved in order to find a sparse representation x.

$$\arg\min_x \|y - Dx\|_2^2 \tag{10}$$

where $\|x\|_0 \leq s$ and s is set sparsity level.

More generally, given a collection of training signals $\{y_i\}_{i=1}^N$ that are represented as columns of the matrix $Y \in \mathbb{R}^{n\times N}$the dictionary learning problem is formulated as follows.

$$\text{(P0)} \quad \arg\min_{D,X} \|Y - DX\|_F^2 \tag{11}$$

where $\|x_i\|_0 \leq s$ and $\|d_i\|_2 = 1$.Here, d_j and x_i denote the columns of the dictionary $D \in \mathbb{R}^{n\times J}$ and sparse code matrix $X \in \mathbb{R}^{J\times N}$ respectively. s denotes the maximum sparsity level allowed for each training signal.

However, problem (P0) is highly non-convex and most dictionary learning approaches may not guarantee convergence. The learning problem with an l_0 sparsity penalty is investigated.

A. Problem Formulation

The sparsity constraints in problem (P0) is replaced with l_0 penalty $\sum_{i=1}^N \|x_i\|_0$. Next with $C = X^T$, we can express $DX = DC^T$ as a sum of (sparse) rank-one matrices or outer products $\sum_{j=1}^J d_j c_j^T$, where c_j is the jth column of C. With these modifications, the new dictionary learning problem can be formulated as follows.

$$\text{(P1)} \quad \arg\min_{\{d_j,c_j\}} \|Y - \sum_{j=1}^J d_j c_j^T\|_F^2 + \lambda^2 \sum_{j=1}^J \|c_j\|_0 \tag{12}$$

where $\|d_j\| = 1, \|c_j\|_\infty \leq L$.

Problem (P1) is designed to learn the factor $\{d_j\}_{j=1}^J$ and $\{c_j\}_{j=1}^J$ that give the best SOUP sparse representation of Y. Learning sparse approximations of data can be an effective way to denoise the data.

B. Algorithm

A block coordinate descent method to estimate the unknowns in Problem (P1) is presented. For each $1 \leq j \leq J$, first solve (P1) with respect to c_j keeping all other variables fixed (sparse coding step). Once c_j is updated, solve the problem(P1) with respect to d_j keeping all other variables fixed (dictionary update step).

1) Sparse Coding Step: Minimizing (P1) with respect to c_j results in the following problem, where $E_j = Y - \sum_{k\neq j} d_k c_k^T$ is a fixed matrix based on the most recent values of all other atoms and coefficients:

$$\arg\min_{\{c_j\}} \|E_j - d_j c_j^T\|_F^2 + \lambda^2 \|c_j\|_0 \tag{13}$$

where $\|c_j\|_\infty \leq L$.

Given $E_j \in \mathbb{R}^{n\times N}$ and $d_j \in \mathbb{R}^n$,and assuming $L > \lambda$, a global minimizer of the sparse coding problem is

$$\hat{c_j} = \min(|H_\lambda(E_j^T d_j)|, L1_N) \odot sign(H_\lambda(E_j^T d_j)) \tag{14}$$

of which

$$(H_\lambda(b))_i = \begin{cases} 0 & |b_i| < \lambda \\ b_i & |b_i| \geq \lambda \end{cases} \tag{15}$$

2) Dictionary Update Step: Minimizing (P1) with respect to d_j leads to the following problem:

$$\arg\min_{\{c_j\}} \|E_j - d_j c_j^T\|_F^2 \tag{16}$$

where $\|d_j\|_2 = 1$.

Given $E_j \in \mathbb{R}^{n\times N}$ and $c_j \in \mathbb{R}^n$, a global minimizer of the the dictionary update problem is

$$\hat{d_j} = \begin{cases} \frac{E_j c_j}{\|E_j c_j\|} & c_j \neq 0 \\ v_1 & c_j = 0 \end{cases} \tag{17}$$

where v_1 is the first column of the $n \times n$ identity matrix.

C. Denoising Examples

In image denoising, the goal is to recover an estimate of an image $x \in \mathbb{R}^M$ from its corrupted measurements $y = x + \epsilon$, where 2D image is represented as vectors x and y. In order to do image denoising by using (P1), firstly all the overlapping patches of the noisy image y are extracted and then training matrix $Y \in \mathbb{R}^{n\times N}$ whose columns are those noisy patches is constructed. To obtain the denoised image estimate, the following least squares problem has to be solved.

$$\arg\min_x \sum_{j=1}^N \|P_j x - \hat{D}\hat{\alpha_j}\|_2^2 + v\|x - y\|_2^2 \tag{18}$$

where $\hat{D}$ and $\hat{\alpha}_j$ denote the learned dictionary and patch sparse codes obtained from the noisy patched, and P_j is an operator that extracts a patch as a vector.

Let $A = \sum_{j=1}^N \|P_j x - \hat{D}\hat{\alpha_j}\|_2^2 + v\|x - y\|_2^2$. Expanding A, we have $A = \sum_{j=1}^N [(P_j x)^T (P_j x) - 2(P_j x)^T \hat{D}\hat{\alpha_j} + (\hat{D}\hat{\alpha_j})^T \hat{D}\hat{\alpha_j}] + vx^T x - 2vx^T y + vy^T y$. By letting $\frac{dA}{dx} = 0$, we could solve for $\hat{x}$:

$$\hat{x} = (\sum_{j=1}^N (P_j^T P_j) + vI)^{-1} (\sum_{j=1}^N P_j^T \hat{D}\hat{\alpha_j} + vy) \tag{19}$$

Let us further explore Eq (19). Assuming the image has size 512×512 and patch size is 8×8, then the dimension of y is $512^2 \times 1$. P_j is an operator that extracts a patch as a vector. Then P_j should be a matrix with dimension 64×512^2. More importantly, $P_j^T P_j$ is a diagonal matrix with dimension $512^2 \times 512^2$.

Therefore, $\sum_{j=1}^N (P_j^T P_j) + vI$ is a diagonal matrix with dimension $512^2 \times 512^2$. Since $\sum_{j=1}^N (P_j^T P_j) + vI$ is a diagonal matrix, the calculation of the inverse of B is very easy. $\sum_{j=1}^N P_j^T \hat{D}\hat{\alpha_j} + vy$ is a long column vector with dimension $512^2 \times 1$. Thus,$\hat{x} = (\sum_{j=1}^N P_j^T \hat{D}\hat{\alpha_j} +$

$vy)./diag(\sum_{j=1}^{N}(P_j^T P_j)+vI)$ where ./ denotes element-wise division and $diag$ denotes the extraction of diagonal elements and forming a long column vector.

The matrix P_j puts patches back and $P_j^T P_j$ is corresponding to which indices were used for the jth patch. $[diag(P_j^T P_j)]_i = 1$ means ith pixel was used by jth patch, otherwise $[diag(P_j^T P_j)]_i = 0$.

Since $\hat{x}$ is a matrix with dimension $512^2 \times 1$, by reshaping it we could get recovered image matrix with dimension 512×512.

After the analysis above, the equation for $\hat{x}$ can be implemented in Matlab. According to the paper, SOUP-DIL denoising uses patches of size 8×8, a 64×256 over-complete DCT dictionary, $\lambda = 5\sigma$, $v = 20/\sigma$ and 10 iterations of learning algorithm. Fig. 3 is the learned dictionary for the image Lena without any noise.

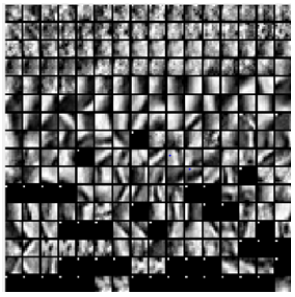


Fig. 3. Learned Dictionary for Lena

The results of applying SOUP-DIL to Lena and Boat images are in Fig. 4 and Fig. 5. The results are very satisfactory.

In order to have a better denoising performance, it is important to subtract the mean of each columns of the training data Y before applying the SOUP-DIL algorithm. Then after applying the SOUP-DIL, the mean has to be added back.The main reason of subtracting the mean and adding it back is that the SOUP-DIL works well for the sparse signals and after subtract the mean, the training data is more sparse in some sense.

IV. Perturbation in Photometric Stereo

From Section II, we see that even though lighting conditions are unknown, the surface can still be reconstructed accurately. We then designed tests to evaluate the robustness of TV-minimization scheme. Given image matrix I, we corrupted I using Gaussian noise and random mask.

$$I_{noise} = I \odot Mask + N \tag{20}$$

$Mask$ is a boolean matrix of which $Mask(i,j) = 0$ with probability p and $Mask(i,j) = 1$ with probability $1-p$. $\odot$ is element-wise product, so p also denotes the probability of missing value in I. $N \sim \mathcal{N}(0, \sigma^2 I)$ denotes Gaussian noise. After we obtain I_{noise}, we apply TV-minimization scheme to it to reconstruct the surface. To quantitatively evaluate the performance, we compute the mean angular error (degree) of normal vectors of perturbed surface with respect to ground truth normal vectors obtained using calibrated photometric stereo.

Fig. 4. Lena Denoised with SOUP-DIL (Left: Corrupted; Middle: Denoised; Right: Original): (a) $\sigma = 5$; (b) $\sigma = 10$; (c) $\sigma = 20$

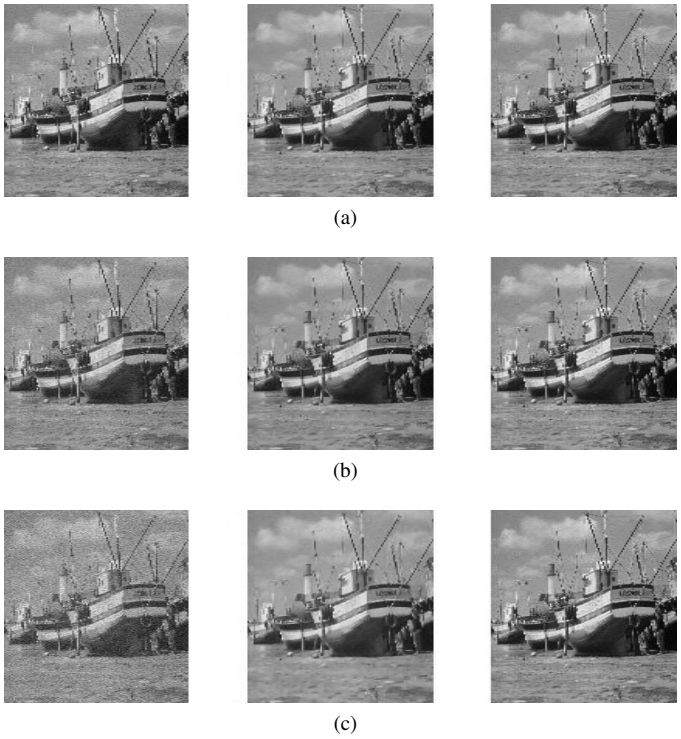
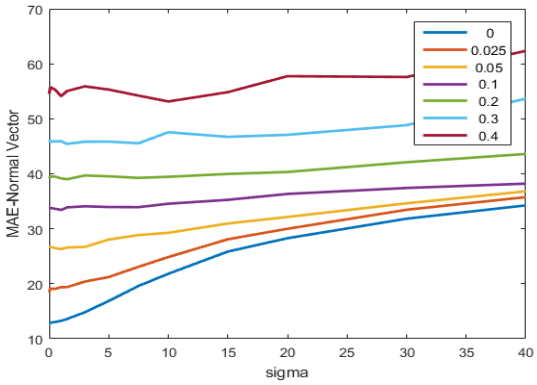


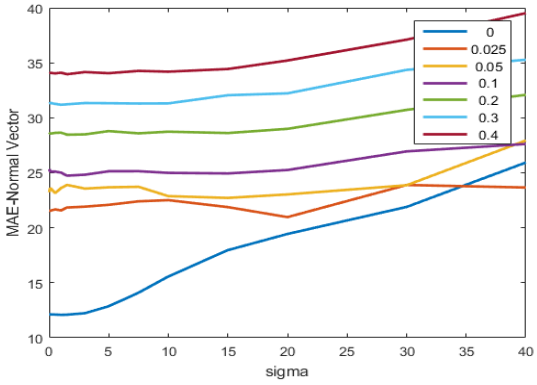
Fig. 5. Lena Denoised with SOUP-DIL (Left: Corrupted; Middle: Denoised; Right: Original): (a) $\sigma = 5$; (b) $\sigma = 10$; (c) $\sigma = 20$

In our test, we scale I so that $max(I) = 255$. σ takes val-

ues from $\{(0, 0.1, 0.5, 1, 1.5, 3, 5, 7.5, 10, 15, 20, 30, 40\}$ and p takes value $\{0, 0.025, 0.05, 0.1, 0.2, 0.3, 0.4\}$. The datasets we use are still the Beethoven and cat datasets. The plots of MAE are in Fig. 6.



(a)



(b)

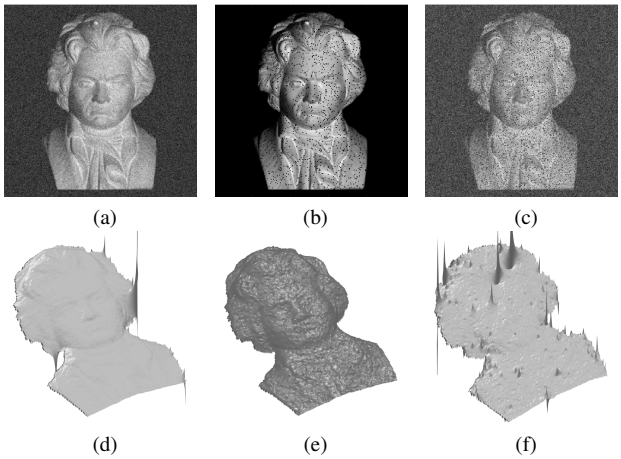
Fig. 6. Mean Angular Error of Normal Vectors of Perturbed Surface for Different p and σ: (a) Beethoven Dataset (b) Cat Dataset

From the plots, we could see that MAE increase as we increase p and σ. And when p is large enough (larger than 0.1), the influence of increasing noise variance will diminish.

The perturbed images and reconstructed surfaces are shown in Fig. 7 and Fig. 8. we could see that noise tend to generate spikes on the surfaces and missing values could make the surface granular. In addition, the performance of cat dataset is much better than Beethoven dataset, which is also demonstrated in Fig. 6. This is because there are only 3 photos in Beethoven dataset while there are 12 photos in cat dataset, which makes it more robust to perturbations. From these tests we could conclude that when there exits perturbation like noise or missing values, the performance of uncalibrated photometric stereo can degrade greatly.

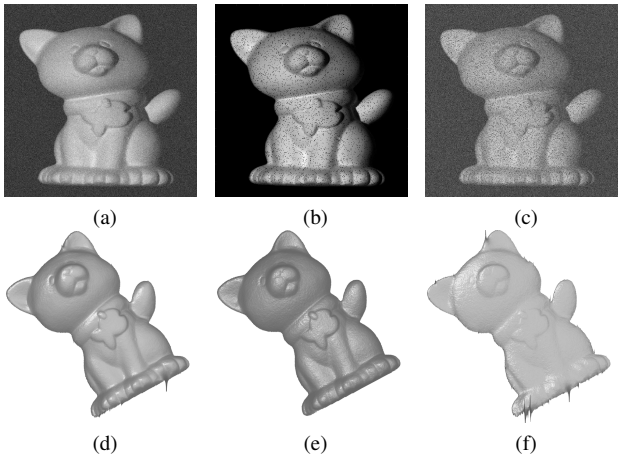
V. Discussion of Denoising Methods

One intuitive way to deal with these perturbation is to denoise the original images and then apply uncalibrated photometric stereo to them. For the following paragraphs, we will discuss three schemes: non-dictionary scheme SVD,



(a) (b) (c) (d) (e) (f)

Fig. 7. Perturbed Images and Corresponding Reconstructed Surfaces of Beethoven Dataset: (a) $p = 0$ and $\sigma = 20$; (b) $p = 0.05$ and $\sigma = 0$; (b) $p = 0.1$ and $\sigma = 30$



(a) (b) (c) (d) (e) (f)

Fig. 8. Perturbed Images and Corresponding Reconstructed Surfaces of Cat Dataset: (a) $p = 0$ and $\sigma = 20$; (b) $p = 0.05$ and $\sigma = 0$; (b) $p = 0.1$ and $\sigma = 30$

fixed dictionary scheme using Discrete Cosine Transform (DCT) dictionary and adaptive dictionary scheme SOUP-DIL described in Section III.

A. SVD Denoising

From Eq (2), we know the rank of I should be 3. One thing to note is rank 3 is the ideal case and no cast shadows or attached shadows exist. Cast shadows form when part of the object block the light rays keeping some other part from being lighted. And attached shadows form when the angle between normal vector of some position and light direction is obtuse thus light cannot get to that position. In our project, we assume the two kinds of shadows are negligible so that the rank of I is three.

Under some perturbation, the randomness will make I full-rank, therefore we could denoise I by taking the singular value decomposition of I and preserving the first three principal component, which is

$$U\Sigma V^T = I_{noise} \qquad I_{rec} = U_{1:3}\Sigma_{1:3}V_{1:3}^T \tag{21}$$

In this case, the recovered I_{rec} strictly has rank 3.

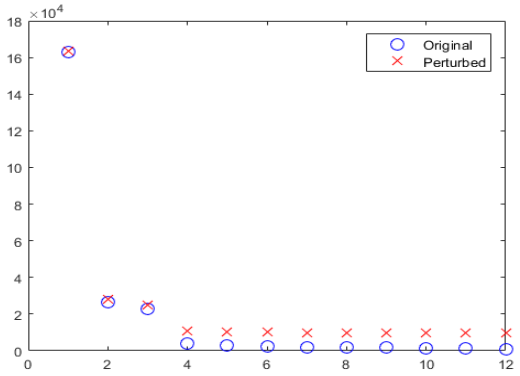
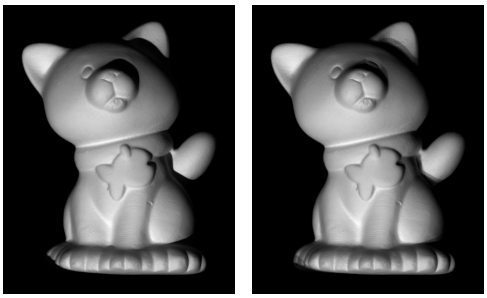


Fig. 9. Singular Values of Original and Perturbed Images (σ=20) of Cat Dataset

Fig. 9 shows that the rank of original I is almost 3 and after perturbation remaining singular values fly up while the first three remain almost unchanged. This plot has justified the SVD denoising method as well as the choice of singular value threshold.

An interesting fact is that SVD denoising method can remove shadows. Fig 10 shows the comparison of original image and SVD-denoised image. We could see the shadows around the ear, nose, left arm and left corner has been reduced. One amazing thing which is not shown in the graph is that for some parts that have attached shadow, i.e. existing obtuse angle between normal vector and light direction, the recovered pixels value even have negative values. This can be explained by the Eq (2), because when the angle is obtuse, the theoretical intensity value which is proportional to their inner product should be negative value rather than the 0 value in real world. What SVD denoising does here is to push I to the direction in which Lamber Cosine Law Eq (2) holds better.



(a) (b)

Fig. 10. Shadow-Removing Effect of SVD Denoising: (a) Original Image; (b) SVD-Denoised Image

B. DCT Denoising

If we take the discrete cosine transform of a natural image, a sparse set of coefficients could be obtained. Under perturbation, the perturbed images are no longer 'natural' thus DCT coefficients are no longer sparse. If we only pick first several important coefficients, we could recover the sparsity and thus recover the images. In our project, we use a 64 $\times$ 256 over-complete DCT dictionary so that the coefficient could be sparser. And we use orthogonal matching pursuit [6] to obtain the coefficients with respect to atoms in DCT dictionary. Fig. 11 shows the over-complete DCT dictionary. After we obtain the sparse coefficients, we could recover the

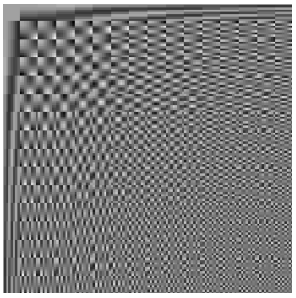


Fig. 11. 64 $\times$ 256 Over-complete Dictionary

images with coefficients and atoms in dictionary.

Since the atom has size 64, we can only denoise an image with size 8 $\times$ 8 at one time. So for a image with larger size, we first convert the whole image into patches with size 8 $\times$ 8 using sliding window, then denoise each patch separately and finally recombine the patches into image. For overlapping pixels appeared in different patches, we choose the mean as the recovered pixel value.

C. SOUP-DIL Denoising

To apply SOUP-DIL denoising to perturbed images, similar to DCT denoising method, we first convert image into 8 $\times$8 size patches, then use dictionary learning to learn the dictionary for all patches as well as corresponding codeword (coefficients). When the learned dictionary and codeword converges after several iterations, we use Eq (19) to obtain the denoised images.

Like Fig. 1, photos used in photometric stereo all look the same, so their learned dictionary and codeword for each frame should also look similar. So one strategy to save computation time of denoising photos is to use learned dictionary and codeword for previous frame as the initial guess for next frame. In this way, the learning process can converge faster.

Fig 12 shows the convergence property for this strategy. The relative errors of recovered image with respect to original image after each iteration are evaluated. We could see as we train more frames, the results converge faster. It could be confusing that after 15 iterations, the second frame has the smallest error and the third frame has the largest error. The is because the second could be best represented using dictionary. But what we are supposed to fucos here is the convergence rate, i.e. the slope of the curves, rather than the value they converge to.

Fig. 13 shows the learned dictionaries for Beethoven and cat dataset. We could see many of the atoms in Fig 13 look the same as those in DCT Dictionary in Fig. 11, demonstrating

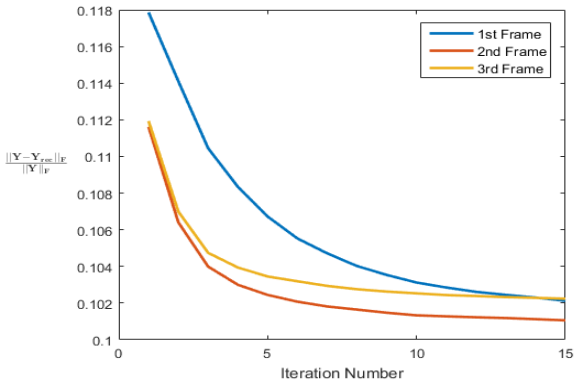
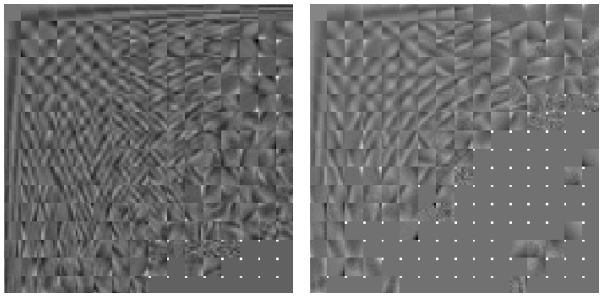


Fig. 12. Convergence Plot of SOUP-DIL Applied to Images in Photometric Stereo



(a) (b)

Fig. 13. Learned Dictionary Using SOUP-DIL: (a) Beethoven Dataset; (b) Cat Dataset

why we could denoise using DCT dictionary. In dictionary for Beethoven dataset, some atoms look like the texture around the hair part of Beethoven and the atoms for cat dataset don't exhibit complicated patterns since there are no textures in the cat photos. The atoms with a white dots on the upper left corner is from the second case Eq (17), we enter the second case since $c_j = 0$. This indicates that to represent the image, we don't have to train a dictionary with 256 atoms, training a smaller dictionary is enough and could save more time. Or on the other hand, we increase sparsity regularization coefficient λ in Eq (12), so that we could make full use of the dictionary and achieve a better denoising performance. This is left to future work.

VI. Experiment and Result

We compared the performances of denoising methods mentioned in Section V on Beethoven and cat datasets. In addition, we also test the performance of combined methods which are "SVD + SOUP-DIL" and "SOUP-DIL + SVD". These methods just mean we deal with perturbed images with two methods applied sequentially. The image matrix I is perturbed as Eq (20) with different noise variance σ^2 (I is scaled so that the largest element in I is 255) and missing value probability p. Then different denoising methods are used to denoise I and finally uncalibrated photometric stereo is applied to reconstruct the surface. To evaluate the performance, we evaluate the mean angular error (MAE) of normal vectors of reconstructed surfaces with respect to normal vectors of ground truth surface reconstructed by calibrated photometric stereo.

A. Experiment Result

Fig. 14 shows the performance comparison of different methods on cat dataset. From the plot, we could see SVD denoising almost has no denoising ability at all. In dictionary denoising methods, adaptive dictionary methods have obvious advantage over DCT denoising. Of the three methods involving SOUP-DIL, "SOUP-DIL + SVD" performs a little worse and the other two. SOUP-DIL and "SVD + SOUP-DIL" have the best performances overall. Since combined methods do not show an obvious improvement compared with SOUP-DIL denoising, so using SOUP-DIL solely is enough.

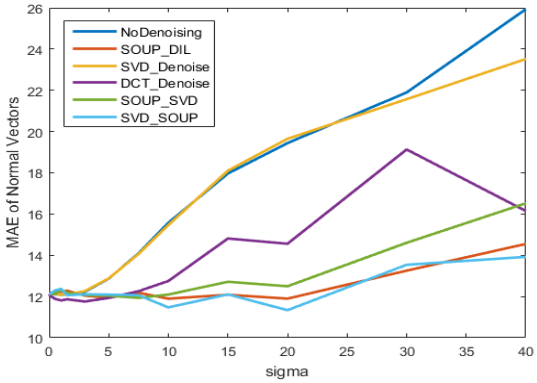


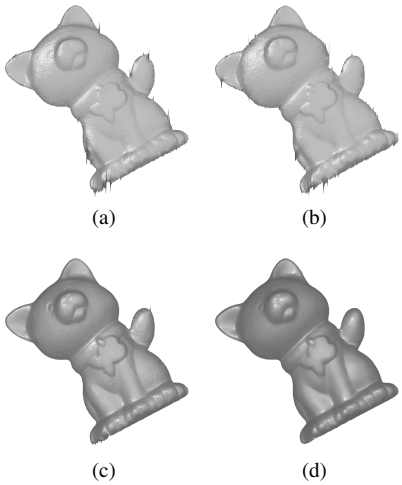
Fig. 14. Performance of Different Denoising Methods on Cat Dataset

Fig. 15 and Fig. 16 give some examples of reconstructed surfaces. We especially choose the worst case $p = 0$, $\sigma = 40$ so that the comparison is more obvious. We could see the surfaces just match the plots shown in Fig .14. Since Beethoven dataset only has three images so we cannot apply SVD denoising. Under same perturbation conditions the cat dataset has better performances than Beethoven dataset indicating again that larger number of photos can increase the robustness.

One thing to note is the results when missing values exist are not provided above. This is because through experiments, we find none of these denoising methods can handle missing values quite well. Fig. 17 give two examples showing that the failure on missing values. For SOUP-DIL denoising, when there are only missing values and no noise, since $\nu = \frac{20}{\sigma}$, according to Eq (19), what is returned is simply the original image perturbed with missing values. To let SOUP-DIL handle missing value, some modification of the model in SOUP-DIL needs to be done, and this is left to future work.

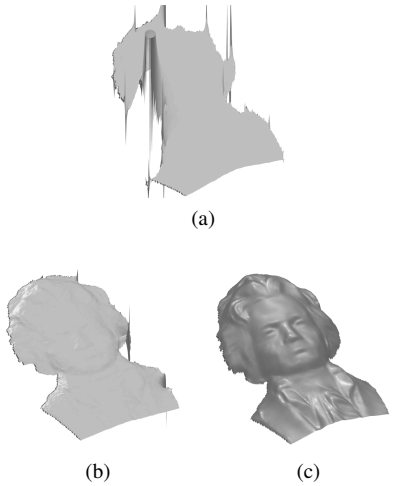
VII. Conclusion

In this project, we implemented different denoising methods to deal with perturbation in uncalibrated photometric stereo. We compared their performances on Beethoven and cat datasets under different perturbation conditions. Results show SOUP-DIL achieves the best performances. We have several



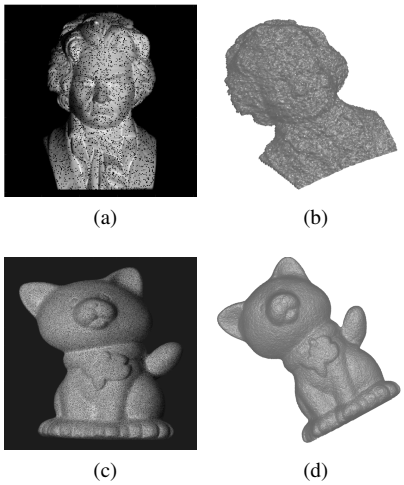
(a) (b) (c) (d)

Fig. 15. Reconstructed Surfaces of Cat Dataset when $p = 0$, $\sigma = 40$: (a) No Denoising; (b) SVD Denoising; (c) DCT Denoising; (d) SOUP-DIL Denoising



(a) (b) (c)

Fig. 16. Reconstructed Surfaces of Beethoven Dataset when $p = 0$, $\sigma = 40$: (a) No Denoising; (b) DCT Denoising; (c) SOUP-DIL Denoising



(a) (b) (c) (d)

Fig. 17. Recovered Images and Reconstructed Surfaces when Missing Values Exist: (a)(b) $p = 0.1$, $\sigma = 0$, DCT-Denoising; (c)(d) $p = 0.2$, $\sigma = 0$, SVD-Denoising

interesting findings like SVD can eliminate the shadows. During the process, several future directions are uncovered, like: (1) SOUP-DIL method can be further modified to deal with missing values; (2) when atoms in dictionary are not all used, we could reduce the dictionary size to save computation time or increase sparsity constraint to increase denoising performance; (3) since the images in a dataset all look similar, we may improve dictionary learning process to save computation time.

REFERENCES

[1] Quau, Yvain, Franois Lauze, and Jean-Denis Durou. "Solving uncalibrated photometric stereo using total variation." Journal of Mathematical Imaging and Vision 52.1 (2015): 87-107.

[2] Alldrin, Neil G., Satya P. Mallick, and David J. Kriegman. "Resolving the generalized bas-relief ambiguity by entropy minimization." Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, 2007.

[3] Favaro, Paolo, and Thoma Papadhimitri. "A closed-form solution to uncalibrated photometric stereo via diffuse maxima." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[4] Woodham, Robert J. "Photometric method for determining surface orientation from multiple images." Optical engineering 19.1 (1980): 191139-191139.

[5] Ravishankar, Saiprasad, Raj Rao Nadakuditi, and Jeffrey A. Fessler. "Efficient Sum of Sparse Outer Products Dictionary Learning (SOUP-DIL)." arXiv preprint arXiv:1511.06333 (2015).

[6] Pati, Yagyensh Chandra, Ramin Rezaiifar, and P. S. Krishnaprasad. "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition." Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on. IEEE, 1993.

[7] Elad, Michael, and Michal Aharon. "Image denoising via sparse and redundant representations over learned dictionaries." Image Processing, IEEE Transactions on 15.12 (2006): 3736-3745.

[8] Yuille, Alan, and Daniel Snow. "Shape and albedo from multiple images using integrability." Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on. IEEE, 1997.