Chapter 13
Robust Photometric Stereo

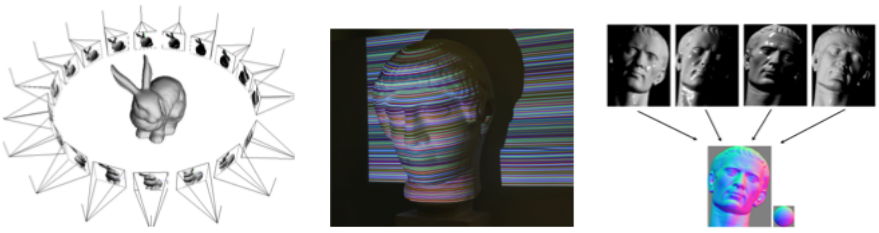
13.1 Introduction

One of the most fundamental problems in computer vision is to capture the 3D shape of an object or a scene. Most popular 3D shape capturing techniques fall into one of the two categories:

1. The so-called *structure from motion* approach reconstructs the 3D geometry by taking multiple images of an object or a scene from different viewpoints [Hartley and Zisserman, 2000, Ma et al., 2004]. See Figure 13.1(a) for illustration. The images are usually taken under the same or a similar lighting condition since such methods rely on establishing correspondence of common feature points across all the images.

2. The *active light* approach captures the 3D shape by taking multiple images of the object or a scene under different illumination conditions or patterns, but usually at a fixed viewpoint. Methods such as structured lights, photometric stereo, and shape from shading all belong to this category. See Figure 13.1(b) and (c) for illustration.

One can tell from the setup that these two approaches are kind of complementary to each other: one varies the camera viewpoints whereas the other varies the lighting conditions. Their results are also complementary to each other: structure from motion techniques typically recover a sparse set of distinguished feature points in the scene that have rich local textures; whereas active lighting techniques usually recover a dense per-pixel geom-



(a) structure from motion (b) structured lights (c) photometric stereo

Figure 13.1. Some techniques for capturing 3D shapes: (a) structure-from-motion takes multiple images of an object from different viewpoints to triangulate its shape; (b) structured light methods cast different light patterns onto an object surface to reveal its 3D geometry; (c) photometric stereo illuminates the object with multiple directional lights to recover its surface normal.

etry (depth or surface normal) of the scene preferably for non-textured regions.

Both approaches have been developed in computer vision and related fields with a long and rich history, and there has been a vast body of literature associated with each method within both categories. In hindsight, though, it would be illuminating to understand now, from the perspective of high-dimensional data analysis, how 3D geometric information of the scene is encoded in the vast data measured in both approaches,[1] and why one can efficiently and accurately recover such information from the data.

According to the settings of both approaches, all the images are capturing a common object or a scene. Then, under some reasonable assumptions such as the scene being mostly static and most surfaces having well-conditioned photometric properties, these imagery data should be highly correlated. It has been well-studied and understood that in the structure-from-motion setting, no matter how many corresponding feature points are captured in arbitrarily many views, they form a large measurement matrix, the so-called *multiple-view matrix*, whose rank will always be bounded below two. Such a low-rank matrix precisely encodes all the camera poses and the depths of all the feature points. Essentially all structure from motion algorithms harness the same low-rank structures to recover the camera poses and feature depths. We refer interested readers to [Ma et al., 2004] for a full account.

The situation is similar in the active light approach. In this chapter, we will use photometric stereo as an example to show that how low-dimensional structures naturally arise from the physical model of the data generation process and how to harness such low-dimensional structures (using tools

[1] Typically hundreds or thousands of feature points in structure-from-motion and the millions of pixels for the active light methods.

from this book) to deal with imperfections in the measurement process so as to accurately recover the object's 3D geometry.

13.2 Photometric Stereo via Low-Rank Matrix Recovery

Photometric stereo [Woodham, 1980, Silver, 1980] has been a very popular method for 3D shape capture. It estimates surface orientations of the scene from images taken from a fixed viewpoint under multiple directional lights. As we will soon see, photometric stereo can produce a dense field of surface normals at the level of detail that cannot be achieved by any other feature-based approaches such as structure-from-motion.

13.2.1 Lambertian Surface under Directional Lights

In the setting for photometric stereo, the relative position of the camera and object is usually fixed. The intrinsic parameters of the camera is usually pre-calibrated and known. We do not need to know the camera pose (the extrinsic parameters) as all geometric quantities can be expressed with respect to the camera frame.

For simplicity, we assume a static object is illuminated by a single point light source at infinity.[2] The direction of the light source can be represented as a vector $\boldsymbol{l} \in \mathbb{R}^3$ (with respect to the camera frame). If we take multiple, say n, images under n different lighting directions, we denote the directions as vectors $\boldsymbol{l}_1, \ldots, \boldsymbol{l}_n \in \mathbb{R}^3$. The magnitude of the vector $\boldsymbol{l}$ is assigned to be proportional to the power of the light source.

Next, we need to know that under the illumination, how much light is reflected from the surface and then measured by the sensor of the camera. Notice that this could be a very complicated process. For every point on the surface, we need to describe by how much the incoming light energy, known as irradiance in radiometry, in any direction is absorbed and emitted in any other outgoing direction, known as radiance. This relationship fully characterizes the photometric properties of the surface and is formally known as the *bidirectional reflectance distribution function* (BRDF). In general, the BRDFs for different material surfaces can be very different. For example, metal, plastic, and cloth look very different under the same light.

Nevertheless, for the majority of the objects and scenes we encounter in the real world, their surface photometric property can be approximately modeled by a simple reflectance function known as the *Lambertian model*. For an ideal Lambertian surface, when illuminated by a light source, the surface diffuses and reflects the light equally in all directions. The fraction

[2] In practice, we only need the light source to be relatively far from the object.

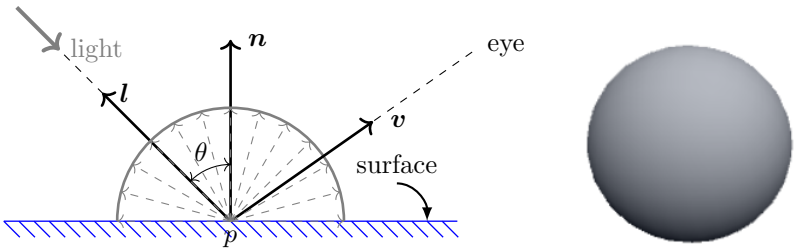


Figure 13.2. Illustration of an ideal Lambertian surface model: incoming light is diffused equally to all directions and the amount of diffused light is proportional to the angle θ between the light direction $\boldsymbol{l}$ and the surface normal $\boldsymbol{n}$. Right: an image of a Lambertian (diffusive) sphere.

of light reflected only depends on the angle between the incoming light direction and the surface normal. More precisely, for a point p on a Lambertian surface illuminated under a light in direction $\boldsymbol{l}$, if the surface normal vector at p is $\boldsymbol{n} \in \mathbb{R}^3$, then the amount of light radiated from point p in all direction is given by (the radiance R):

$$R = \rho \langle \boldsymbol{n}, \boldsymbol{l} \rangle = \rho \boldsymbol{n}^T \boldsymbol{l} = \rho \cos(\theta) \|\boldsymbol{l}\|, \tag{13.2.1}$$

where ρ is the diffuse albedo that models the percentage of light gets reflected by the surface at point p, $\langle \cdot, \cdot \rangle$ is the inner product, and θ is the angle between the light direction $\boldsymbol{l}$ and surface $\boldsymbol{n}$. See Figure 13.2 for a basic idea. It is easy to see from the model that the brightness of the point p does not depend on the view direction $\boldsymbol{v}$.

The albedo ρ of a purely black surface would be zero, hence photometric stereo does not apply to black surfaces or surfaces with very small albedo. Note that the above expression is only valid when $\boldsymbol{n}$ and $\boldsymbol{l}$ form an acute angle ($\theta < 90°$) since radiance R is nonnegative. That is, the surface needs to face the light source. In the case when the surface is facing away from the light source (i.e. $\theta > 90°$), it receives no irradiance hence $R = 0$. We say such area is in the shadow. See the bottom of the image of the sphere in Figure 13.2.

We further assume that there is no inter-reflection,[3] which is often the case if the object is convex or approximately convex. So the corresponding pixel on imaging sensor receives only radiance R contributed from a single point p. If the imaging sensor responses linearly to the radiance, the value of the pixel (x, y) (at the image of the point p) would simply be

$$I(x, y) = R = \rho \boldsymbol{n}^T \boldsymbol{l}. \tag{13.2.2}$$

[3]Inter-reflection is a phenomenon where lights bound off surfaces multiple times before reaching the sensor.

Let the region of interest be composed of a total of m pixels in each image.[4] We order the pixels with a single index k, and let $I_j(k)$ denote the observed intensity at pixel k in image I_j. With this notation, we have the following relation about the observation $I_j(k)$:

$$I_j(k) = \rho_k \, \boldsymbol{n}_k^T \boldsymbol{l}_j, \tag{13.2.3}$$

where ρ_k is the albedo of the scene at pixel k, $\boldsymbol{n}_k \in \mathbb{R}^3$ is the (unit) surface normal of the scene at pixel k, and $\boldsymbol{l}_j \in \mathbb{R}^3$ represents the light direction vector corresponding to image I_j.[5]

Consider the matrix $\boldsymbol{D} \in \mathbb{R}^{m \times n}$ constructed by stacking all the vectorized images $\mathrm{vec}(I)$ as

$$\boldsymbol{D} \doteq [\mathrm{vec}(I_1) \mid \cdots \mid \mathrm{vec}(I_n)], \tag{13.2.4}$$

where $\mathrm{vec}(I_j) = [I_j(1), \ldots, I_j(m)]^T$ for $j = 1, \ldots, n$. It follows from (13.2.3) that $\boldsymbol{D}$ can be factorized as follows:

$$\boldsymbol{D} = \boldsymbol{N} \cdot \boldsymbol{L}, \tag{13.2.5}$$

where $\boldsymbol{N} \doteq [\rho_1 \boldsymbol{n}_1 \mid \cdots \mid \rho_m \boldsymbol{n}_m]^T \in \mathbb{R}^{m \times 3}$, and $\boldsymbol{L} \doteq [\, \boldsymbol{l}_1 \mid \cdots \mid \boldsymbol{l}_n \,] \in \mathbb{R}^{3 \times n}$. Suppose that the number of images $n \geq 3$. Then, irrespective of the number of pixels m and the number of images n, the rank of the matrix $\boldsymbol{D}$ is at most 3.

13.2.2 Modeling Shadows and Specularities

The low-rank structure of the observation matrix $\boldsymbol{D}$ (13.2.5) is seldom observed with real images. This is due to the presence of shadows and specularities in real images.

Shadows

Shadows arise in real images in two possible ways. As we have discussed before in the Lambertian model, some areas on the object will be entirely dark in the image because they face away from the light source. Such dark pixels in the image are referred to as *attached shadows* [Knill et al., 1997]. See the image of a sphere in Figure 13.2 as an example, where the bottom of the sphere is dark as that part of surface is facing away from the light source. In deriving the low-rank model (13.2.5) from (13.2.3), we have implicitly assumed that all pixels of the object are illuminated by the light source in every image. However, that is impossible to achieve in reality: for a generic object (other than a flat surface), almost in every image, there will

[4] Typically, m is much larger than the number of images n.

[5] The convention here is that the lighting direction vectors point from the surface of the object to the light source.

always be some pixels facing away from the light source and in the shadows. Mathematically, this implies that (13.2.3) should be modified as follows:

$$I_j(k) = \max\left\{\rho_k \, \boldsymbol{n}_k^T \boldsymbol{l}_j, 0\right\}. \tag{13.2.6}$$

Shadows can also occur in images when the shape of the object's surface is not entirely convex: parts of the surface can be occluded from the light source by other parts. Even though the normal vectors at such occluded pixels may form an acute angle with the lighting direction, these pixels appear entirely dark. We refer to such dark pixels as *cast shadows*. See the image of Caesar in Figure 13.5 as an example, where, unlike the sphere, the face is not exactly convex and the sporadic shadows around the left side of Caesar's face are cast shadows due to occlusions.

We may pre-detect all the dark shadowed pixels in each images by testing if

$$I_j(k) \approx 0.$$

These pixels are associated with a set entries $\{(k, j)\}$ in the data matrix $\boldsymbol{D}$ where $\boldsymbol{D}(k, j) \approx 0$. We denote the support of these shadowed entries as Ω^c and all the other valid entries as its complement as Ω. With this notation, the valid measurements of the (low-rank) data matrix $\boldsymbol{D}$ are given by

$$\mathcal{P}_\Omega(\boldsymbol{D}) = \mathcal{P}_\Omega(\boldsymbol{N} \cdot \boldsymbol{L}). \tag{13.2.7}$$

For the remaining pixels not in the shadows, we have assumed that each pixel measures the radiance directly from each point on the surface. For a non-convex object like a human face, that is not entirely the case. Lights can bound back and forth between different part of the surfaces and create the so-called *inter-reflection*. The radiance that some pixels receive might be compounded by such inter-reflection. Nevertheless, studies have shown that if the object is approximately convex, pixels that are affected by inter-reflection will be relatively few [**?**]. We may model such effect as a sparse error $\boldsymbol{E}_1$ in the data matrix:

$$\mathcal{P}_\Omega(\boldsymbol{D}) = \mathcal{P}_\Omega(\boldsymbol{N} \cdot \boldsymbol{L} + \boldsymbol{E}_1). \tag{13.2.8}$$

Specularities

Specular reflection arises when the object of interest is not perfectly diffusive, i.e., when the surface luminance is not purely isotropic. Mirror is an extreme case which reflects the light with the same angle as the in coming light on the opposite side of the surface normal:

$$\boldsymbol{r} = 2(\boldsymbol{n}^T \boldsymbol{l})\boldsymbol{n} - \boldsymbol{l}.$$

Many real surfaces have both diffusive and reflective characteristics and their reflectance model is a combination of a Lambertian component and a reflective component. The so-called *Phong model* [Phong, 1975] is a

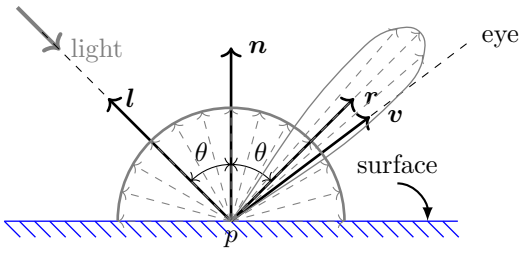


Figure 13.3. A Phong reflectance model: incoming light is diffused equally to all directions and an extra amount of light is reflected close to the direction of reflection $\boldsymbol{r}$, known as a specular lobe.

correction to the pure Lambertian model with such a reflective component:

$$R = \rho \boldsymbol{n}^T \boldsymbol{l} + k(\boldsymbol{r}^T \boldsymbol{v})^\alpha, \tag{13.2.9}$$

where $\boldsymbol{v}$ is the viewing direction (to the sensor), k is a weight parameter, and α is an exponent parameter. Figure 13.3 illustrates such a reflectance model. In the computer vision and graphics literature, people have also used other functions to model the reflective component, such as the Cook-Torrance reflectance model [Cook and Torrance, 1981].

In general, for a surface of Phong model (or of Cook-Torrance model), the intensity of radiance depends on the viewing direction: part of the light is reflected in a mirror-like fashion that generates a specular lobe when the viewing direction $\boldsymbol{v}$ is close to the reflecting direction $\boldsymbol{r}$. This gives rise to some bright spots or shiny patches on the surface of the object, known as *specularities*. Figure 13.3 illustrates this concept and Figure 13.4 compares the Phong model to the Lambertian model with the images of a sphere.

Figure 13.4. Comparison of a Lambertian (diffusive) sphere and a Phong (specular) sphere under the same (directional) lighting condition.

For most real surfaces, the reflective components are usually benign in the sense that the value of the reflective term is significant only when the

view direction is very close to the reflecting direction.[6] The specular lobe is usually very small, and from any given viewing angle, only a small fraction of the surface has the specular effect. See Figure 13.5 for some examples of object surfaces with specular effect.

(a) Sphere (b) Caesar (c) Elephant (d) Caesar (with texture)

Figure 13.5. Synthetic image samples used for experiments.

As the surface normals and the viewing angles are not known a priori, we cannot determine which part of the surface is specular. Nevertheless, knowing that specularities are few and sporadic, we may model them as an additional sparse error $\boldsymbol{E}_2$ to the measured data matrix $\boldsymbol{D}$:

$$\boldsymbol{D} = \boldsymbol{N} \cdot \boldsymbol{L} + \boldsymbol{E}_2, \tag{13.2.10}$$

Now, if we combine the sparse errors $\boldsymbol{E}_1$ due to inter-reflections and $\boldsymbol{E}_2$ due to specularities and let $\boldsymbol{E} = \boldsymbol{E}_1 + \boldsymbol{E}_2$, then, instead of the ideal low-rank model: (13.2.5), a more realistic model for the image measurements should be:

$$\mathcal{P}_\Omega(\boldsymbol{D}) = \mathcal{P}_\Omega(\boldsymbol{N} \cdot \boldsymbol{L} + \boldsymbol{E}), \tag{13.2.11}$$

where Ω marks out pixels in the shadows and the sparse matrix $\boldsymbol{E}$ accounts for corruptions by inter-reflections or specularities.

In order to find out the light directions $\boldsymbol{L}$ and the surface normals $\boldsymbol{N}$, we need to recover the complete matrix $\boldsymbol{A} = \boldsymbol{N} \cdot \boldsymbol{L}$. Since $\boldsymbol{A}$ is of rank at most 3, the problem becomes a low-rank matrix completion problem subject to sparse errors $\boldsymbol{E}$. That is we need to solve the following optimization problem:

$$\min_{\boldsymbol{A},\boldsymbol{E}} \operatorname{rank}(\boldsymbol{A}) + \gamma \|\boldsymbol{E}\|_0 \quad \text{s.t.} \quad \mathcal{P}_\Omega(\boldsymbol{D}) = \mathcal{P}_\Omega(\boldsymbol{A} + \boldsymbol{E}), \tag{13.2.12}$$

where $\|\cdot\|_0$ denotes the ℓ_0-norm (number of non-zero entries in the matrix), and $\gamma > 0$ is a parameter that trades off the rank of the solution $\boldsymbol{A}$ versus the sparsity of the error $\boldsymbol{E}$.

[6] In the Phong model, that corresponds to choosing a large exponent α.

Let $(\hat{\boldsymbol{A}}, \hat{\boldsymbol{E}})$ be the optimal solution to (13.2.12). If the lighting directions $\boldsymbol{L}$ are given, we can easily recover the matrix $\boldsymbol{N}$ of surface normals from $\hat{\boldsymbol{A}}$ as:

$$\boldsymbol{N} = \hat{\boldsymbol{A}}\boldsymbol{L}^{\dagger}, \tag{13.2.13}$$

where $\boldsymbol{L}^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of $\boldsymbol{L}$. The surface normals $\boldsymbol{n}_1, \ldots, \boldsymbol{n}_m$ can then be estimated by normalizing each row of $\boldsymbol{N}$ to have unit norm.

13.3 Robust Matrix Completion Algorithm

While (13.2.12) follows from our formulation, it is not tractable since both rank and ℓ_0-norm are non-convex and discontinuous functions. As we have learned from earlier chapters, we can try to solve the convex version of this program:

$$\min_{\boldsymbol{A},\boldsymbol{E}} \|\boldsymbol{A}\|_* + \lambda \|\boldsymbol{E}\|_1 \quad \text{s.t.} \quad \mathcal{P}_{\Omega}(\boldsymbol{D}) = \mathcal{P}_{\Omega}(\boldsymbol{A} + \boldsymbol{E}). \tag{13.3.1}$$

The above problem is almost identical to the PCP program studied in Chapter 5, except that the linear equality constraint is now applied only on the subset Ω of pixels that are not in the shadows. In the rest of this section, we show how the Augmented Lagrange Multiplier (ALM) method, earlier introduced for matrix completion or matrix recovery in Chapter 5, can be adapted to efficiently solve the problem (13.3.1) that requires simultaneously completing and correcting a low-rank matrix.

Recall the basic idea of the ALM method is to minimize the augmented Lagrangian function instead of the original constrained optimization problem. For our problem (13.3.1), the augmented Lagrangian is given by

$$\mathcal{L}_{\mu}(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{Y}) = \|\boldsymbol{A}\|_* + \lambda\|\boldsymbol{E}\|_1 + \langle \boldsymbol{Y}, \mathcal{P}_{\Omega}(\boldsymbol{D} - \boldsymbol{A} - \boldsymbol{E})\rangle + \frac{\mu}{2}\|\mathcal{P}_{\Omega}(\boldsymbol{D} - \boldsymbol{A} - \boldsymbol{E})\|_F^2, \tag{13.3.2}$$

where $\boldsymbol{Y} \in \mathbb{R}^{m \times n}$ is a Lagrange multiplier matrix, μ is a positive constant, $\langle \cdot, \cdot \rangle$ denotes the matrix inner product,[7] and $\|\cdot\|_F$ denotes the Frobenius norm. For appropriate choice of the Lagrange multiplier matrix $\boldsymbol{Y}$ and sufficiently large constant μ, it can be shown that the augmented Lagrangian function has the same minimizer as the original constrained optimization problem. The ALM algorithm iteratively estimates both the Lagrange multiplier and the optimal solution. The basic ALM iteration is given by

$$\begin{cases} (\boldsymbol{A}_{k+1}, \boldsymbol{E}_{k+1}) &= \operatorname{argmin}_{\boldsymbol{A},\boldsymbol{E}} \mathcal{L}_{\mu_k}(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{Y}_k), \\ \boldsymbol{Y}_{k+1} &= \boldsymbol{Y}_k + \mu_k \mathcal{P}_{\Omega}(\boldsymbol{D} - \boldsymbol{A}_{k+1} - \boldsymbol{E}_{k+1}), \\ \mu_{k+1} &= \rho \cdot \mu_k, \end{cases} \tag{13.3.3}$$

[7] $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle \doteq \operatorname{trace}(\boldsymbol{X}^T\boldsymbol{Y})$.

where $\{\mu_k\}$ is a monotonically increasing positive sequence ($\rho > 1$).

We now focus our attention on solving the non-trivial first step of the above iteration. Since it is difficult to minimize $\mathcal{L}_{\mu_k}(\cdot)$ with respect to both $\boldsymbol{A}$ and $\boldsymbol{E}$ simultaneously, we adopt an alternating minimization strategy as follows:

$$\left\{ \begin{array}{rcl} \boldsymbol{E}_{j+1} & = & \text{argmin}_{\boldsymbol{E}}\ \lambda\|\boldsymbol{E}\|_1 - \langle \boldsymbol{Y}_k, \mathcal{P}_\Omega(\boldsymbol{E})\rangle + \frac{\mu_k}{2}\|\mathcal{P}_\Omega(\boldsymbol{D} - \boldsymbol{A}_j - \boldsymbol{E})\|_F^2, \\ \boldsymbol{A}_{j+1} & = & \text{argmin}_{\boldsymbol{A}}\ \|\boldsymbol{A}\|_* - \langle \boldsymbol{Y}_k, \mathcal{P}_\Omega(\boldsymbol{A})\rangle + \frac{\mu_k}{2}\|\mathcal{P}_\Omega(\boldsymbol{D} - \boldsymbol{A} - \boldsymbol{E}_{j+1})\|_F^2. \end{array} \right. \tag{13.3.4}$$

Without loss of generality, we assume that the $\boldsymbol{Y}_k$'s and the $\boldsymbol{E}_k$'s (and hence, $\boldsymbol{Y}$ and $\boldsymbol{E}$, respectively) have their support in Ω^c. Then, the above minimization problems in (13.3.4) can be solved as described below.

We first define the *shrinkage* (or soft-thresholding) operator for scalars as follows:

$$\text{shrink}(x, \alpha) = \text{sgn}(x) \cdot \max\{|x| - \alpha, 0\}, \tag{13.3.5}$$

where $\alpha > 0$.[8] When applied to vectors or matrices, the shrinkage operator acts element-wise. Then, the first step in (13.3.4) has a closed-form solution given by

$$\boldsymbol{E}_{j+1} = \text{shrink}\left(\mathcal{P}_\Omega(\boldsymbol{D}) + \frac{1}{\mu_k}\boldsymbol{Y}_k - \mathcal{P}_\Omega(\boldsymbol{A}_j), \frac{\lambda}{\mu_k}\right). \tag{13.3.6}$$

Since it is not possible to express the solution to the second step in (13.3.4) in closed-form, we adopt an iterative strategy based on the Accelerated Proximal Gradient (APG) algorithm discussed in Chapter 8.3 to solve it. The iterative procedure is given as:

$$\left\{ \begin{array}{rcl} (\boldsymbol{U}_i, \boldsymbol{\Sigma}_i, \boldsymbol{V}_i) & = & \text{svd}\left(\frac{1}{\mu_k}\boldsymbol{Y}_k + \mathcal{P}_\Omega(\boldsymbol{D}) - \boldsymbol{E}_{j+1} + \mathcal{P}_{\Omega^c}(\boldsymbol{Z}_i)\right), \\ \boldsymbol{A}_{i+1} & = & \boldsymbol{U}_i\ \text{shrink}\left(\boldsymbol{\Sigma}_i, \frac{1}{\mu_k}\right)\boldsymbol{V}_i^T, \\ \boldsymbol{Z}_{i+1} & = & \boldsymbol{A}_{i+1} + \frac{t_i - 1}{t_{i+1}}(\boldsymbol{A}_{i+1} - \boldsymbol{A}_i), \end{array} \right. \tag{13.3.7}$$

where svd$(\cdot)$ denotes the singular value decomposition operator, and $\{t_i\}$ is a positive sequence satisfying $t_1 = 1$ and $t_{i+1} = 0.5\left(1 + \sqrt{1 + 4t_i^2}\right)$. The entire algorithm to solve (13.3.1) has been summarized as Algorithm 13.1.

13.4 Experimental Evaluation

In this section, we verify the effectiveness of the proposed method using both synthetic and real-world images. We compare results of the above robust matrix completion (RMC) method with a simple Least Squares (LS)

[8] If $\alpha = 0$, then the shrinkage operator reduces to the identity operator.

Algorithm 13.1 (Matrix Completion and Recovery via ALM).

INPUT: $\boldsymbol{D} \in \mathbb{R}^{m\times n}$, $\Omega \subset \{1,\ldots,m\} \times \{1,\ldots,n\}$, $\lambda > 0$.
Initialize $\boldsymbol{A}_1 \leftarrow 0$, $\boldsymbol{E}_1 \leftarrow 0$, $\boldsymbol{Y}_1 \leftarrow 0$.
while not converged $(k = 1, 2, \ldots)$ **do**
  $\boldsymbol{A}_{k,1} = \boldsymbol{A}_k$, $\boldsymbol{E}_{k,1} = \boldsymbol{E}_k$;
  while not converged $(j = 1, 2, \ldots)$ **do**
    $\boldsymbol{E}_{k,j+1} = \text{shrink}\left(\mathcal{P}_\Omega(\boldsymbol{D}) + \frac{1}{\mu_k}\boldsymbol{Y}_k - \mathcal{P}_\Omega(\boldsymbol{A}_{k,j}), \frac{\lambda}{\mu_k}\right)$;
    $t_1 = 1$; $\boldsymbol{Z}_1 = \boldsymbol{A}_{k,j}$; $\boldsymbol{A}_{k,j,1} = \boldsymbol{A}_{k,j}$;
    while not converged $(i = 1, 2, \ldots)$ **do**
      $(\boldsymbol{U}_i, \boldsymbol{\Sigma}_i, \boldsymbol{V}_i) = \text{svd}\left(\frac{1}{\mu_k}\boldsymbol{Y}_k + \mathcal{P}_\Omega(\boldsymbol{D}) - \boldsymbol{E}_{k,j+1} + \mathcal{P}_{\Omega^c}(\boldsymbol{Z}_i)\right)$;
      $\boldsymbol{A}_{k,j,i+1} = \boldsymbol{U}_i\, \text{shrink}\left(\boldsymbol{\Sigma}_i, \frac{1}{\mu_k}\right)\, \boldsymbol{V}_i^T$, $t_{i+1} = 0.5\left(1 + \sqrt{1 + 4t_i^2}\right)$;
      $\boldsymbol{Z}_{i+1} = \boldsymbol{A}_{k,j,i+1} + \frac{t_i - 1}{t_{i+1}}(\boldsymbol{A}_{k,j,i+1} - \boldsymbol{A}_{k,j,i})$, $\boldsymbol{A}_{k,j+1} = \boldsymbol{A}_{k,j,i+1}$;
    end while
    $\boldsymbol{A}_{k+1} = \boldsymbol{A}_{k,j+1}$; $\boldsymbol{E}_{k+1} = \boldsymbol{E}_{k,j+1}$;
  end while
  $\boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k + \mu_k\, \mathcal{P}_\Omega(\boldsymbol{D} - \boldsymbol{A}_{k+1} - \boldsymbol{E}_{k+1})$, $\mu_{k+1} = \rho \cdot \mu_k$;
end while
OUTPUT: $(\hat{\boldsymbol{A}}, \hat{\boldsymbol{E}}) = (\boldsymbol{A}_k, \boldsymbol{E}_k)$.

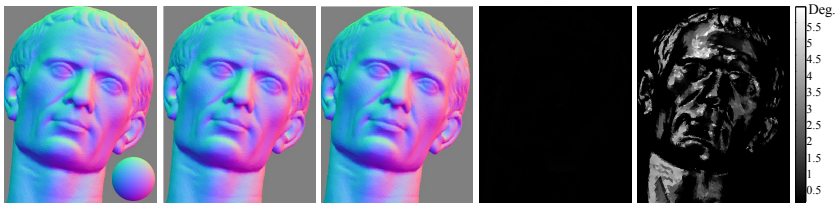
approach, which assumes the ideal diffusive model given by (13.2.5). However, we do not use those pixels that were classified as shadows (the set Ω). Thus, the LS method can be summarized by the following optimization problem:

$$\min_{\boldsymbol{N}} \|\mathcal{P}_\Omega(\boldsymbol{D} - \boldsymbol{N} \cdot \boldsymbol{L})\|_F. \tag{13.4.1}$$

We first test the algorithms using synthetic images whose ground-truth normal maps are known. In these experiments, we quantitatively verify the correctness of the algorithms by computing the angular errors between the estimated normal map and the ground-truth. We then test the algorithms on more challenging real images. Throughout this section, we denote by m the number of pixels in the region of interest in each image, and by n the number of input images (typically, $m \gg n$).

13.4.1 Quantitative Evaluation with Synthetic Images

In this section, we use synthetic images of three different objects (see Fig. 13.5(a)-(c)) under different scenarios to evaluate the performance of the algorithms. Since these images are free of any noise, we use a pixel threshold value of zero to detect shadows in the images. Unless otherwise stated, we set $\lambda = 1/\sqrt{m}$ in (13.3.1).



(a) Ground truth (b) Our method (c) Least Squares (d) Error map (our method) (e) Error map (LS)

Figure 13.6. **Specular scene.** 40 different images of Caesar were generated using the Cook-Torrance model for specularities. (a) Ground truth normal map with reference sphere. (b) and (c) show the surface normals recovered by the robust matrix completion (RMC) method and LS, respectively. (d) and (e) show the pixel-wise angular error w.r.t. the ground truth.

a. Specular objects.

In this experiment, we generate images of an object under 40 different lighting conditions, where the lighting directions are chosen at random from a hemisphere with the object placed at the center. The images are generated with some specular reflection. For all experiments, we use the Cook-Torrance reflectance model [Cook and Torrance, 1981] to generate images with specularities. Thus, there are two sources of corruption in the images – attached shadows and specularities.

A quantitative evaluation of our method and the Least Squares approach is presented in Table 13.1. The estimated normal maps are shown in

Object	Mean error		Max. error		% of corrupted pixels	
	LS	RMC	LS	RMC	Shadow	Specularity
Sphere	0.99	$\mathbf{5.1 \times 10^{-3}}$	8.1	**0.20**	18.4	16.1
Caesar	0.96	$\mathbf{1.4 \times 10^{-2}}$	8.0	**0.22**	20.7	13.6
Elephant	0.96	$\mathbf{8.7 \times 10^{-3}}$	8.0	**0.29**	18.1	16.5

Table 13.1. **Specular scene.** Statistics of angle error (in degrees) in the normals for different objects. In each case, 40 images were used. In the rightmost column, we indicate the average percentage of pixels corrupted by attached shadows and specularities in each image.

Fig. 13.6(b),(c). We use the RGB channel to encode the 3 spatial components (XYZ) of the normal map for display purposes. The error is measured in terms of the angular difference between the ground truth normal and the estimated normal at each pixel location. The pixel-wise error maps are shown in Fig. 13.6(d),(e). From the mean and the maximum angular error (in degrees) in Table 13.1, we see that the RMC method is much more accurate than the LS approach. This is because specularities introduce large

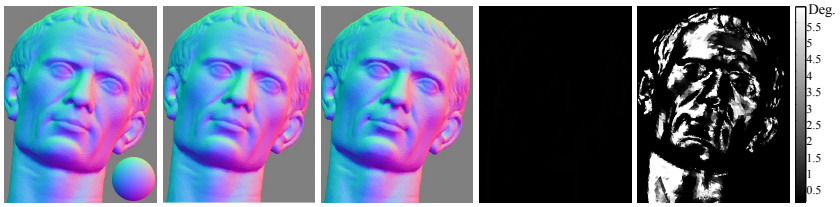


Figure 13.7. **Textured scene with specularity.** 40 different images of Caesar were generated with texture and using the Cook-Torrance model for specularities. (a) Ground truth normal map with reference sphere. (b) and (c) show the surface normals recovered by RMC and LS, respectively. (d) and (e) show the pixel-wise angular error wrt ground truth.

magnitude errors to a small fraction of pixels in each image whose locations are unknown. The LS algorithm is not robust to such corruptions while RMC can correct these errors and recover the underlying rank-3 structure of the matrix. The column on the extreme right of Table 13.1 indicates the average percentage of pixels in each image (averaged over all images) that were corrupted by shadows and specularities, respectively. We note that even when more than 30% of the pixels are corrupted by shadows and specularities, RMC can efficiently retrieve the surface normals.

b. Textured objects.

We also test the RMC method using a textured scene. Like the traditional photometric stereo approach, the RMC method does not have a dependency on the albedo distribution and works well on such scenes.

We use 40 images of Caesar for this experiment with each image generated under a different lighting condition (see Fig. 13.5(d) for example input image). The estimated normal maps as well as the pixel-wise error maps are shown in Fig. 13.7. We provide a quantitative comparison in Table 13.2 with respect to the ground-truth normal map. From the mean and maximum angular errors, it is evident that the RMC performs much better than the LS approach in this scenario.

Object	Mean error (in degrees)		Max error (in degrees)	
	LS	RMC	LS	RMC
Caesar	2.4	**0.016**	32.2	**0.24**

Table 13.2. **Textured scene with specularity:** Statistics of angle errors. We use 40 images under different illuminations.

c. Effect of the number of input images.

In the above experiments, we have used images of the object under 40 different illuminations. In this experiment, we study the effect of the number of illuminations used. In particular, we would like to find out empirically the minimum number of images required for the RMC method to be effective. For this experiment, we generate images of Caesar using the Cook-Torrance reflectance model, where the lighting directions are generated at random. The mean percentage of specular pixels in the input images is maintained approximately constant at 10%. The angular difference between the estimated normal map and the ground truth is used as a measure of accuracy of the estimate.

Num of images		10	20	30	40
Mean error (in degrees)	LS	0.52	0.53	0.59	0.57
	RMC	**0.23**	**0.026**	**0.019**	**0.013**
Max. error (in degrees)	LS	**34.5**	9.0	7.6	7.0
	RMC	56.6	**5.8**	**0.48**	**0.37**

Table 13.3. **Effect of number of input images.** We use synthetic images of Caesar under different lighting conditions. The number of illuminations is varied from 10 to 40. The angle error is measured with respect to the ground truth normal map. The illuminations are chosen at random, and the error has been averaged over 20 different sets of illumination.

We present the experimental results in Table 13.3. We observe that with less than 10 input illuminations, estimates of both algorithms are very inaccurate but RMC is worse than LS. However, when the number of illuminations is larger than 10, we observe that the mean error in the LS estimate becomes higher than that RMC. Upon increasing the number of images further, the proposed method consistently outperforms the LS approach. If the number of input images is less than 20, then the maximum error in the LS estimate is smaller than that of RMC. However, RMC performs much better when more than 30 different illuminations are available. Thus, the proposed technique performs significantly better as the number of input images increases.

d. Varying amount of specularities.

From the above experiments, it is clear that the proposed technique is quite robust to specularities in the input images when compared to the LS method. In this experiment, we empirically determine the maximum amount of specularity that can be handled by RMC. We use the Caesar scene under 40 randomly chosen illumination conditions for this experiment. On an average, about 20% of the pixels in each image is corrupted by attached shadows. We vary the size of the specular lobe in the input

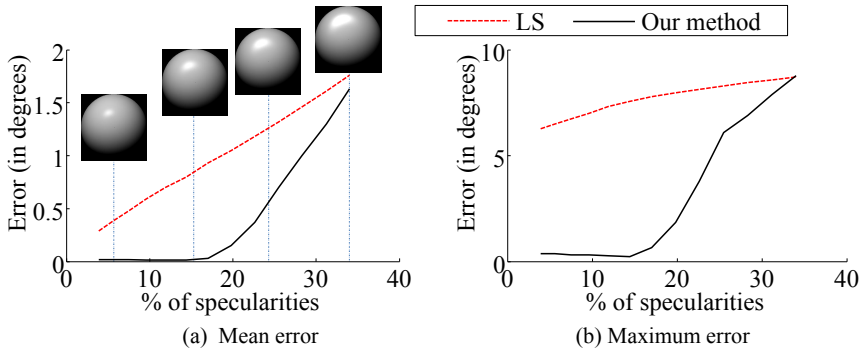


Figure 13.8. **Effect of increasing size of specular lobes.** We use synthetic images of Caesar under 40 randomly chosen lighting conditions. (a) Mean angular error, (b) Maximum angular error w.r.t. the ground truth. The illuminations are chosen at random, and the error has been averaged over 10 different sets of illumination. (a) contains illustrations of increasing size of specular lobe.

images (as illustrated in Fig. 13.8(a)), thereby varying the number of corrupted pixels. We compare the accuracy of RMC against the LS technique using the angular error of the estimates with respect to the ground-truth.

The experimental results are illustrated in Fig. 13.8. We observe that RMC is very robust when up to 16% of all pixels in the input images are corrupted by specularities. The LS method, on the other hand, is extremely sensitive to even small amounts of specularities in the input images. The angular error in the estimates of both methods rises as the size of the specular lobe increases.

e. Enhancing performance by better choice of λ.

We recall that λ is a weighting parameter in our formulation given by (13.3.1). In all the above experiments, we have fixed the value of the parameter $\lambda = 1/\sqrt{m}$, as suggested by the theory in Chapter 5. While this choice promises a certain degree of error correction, it may be possible to correct larger amounts of corruption by choosing λ appropriately, as demonstrated in [Ganesh et al., 2010] for instance. Unfortunately, the best choice of λ depends on the input images, and cannot be determined analytically.

We demonstrate the effect of the weighting parameter λ on a set of 40 images of Caesar used in the previous experiments. In this set of images, approximately 20% of the pixels are corrupted by attached shadows and about 28% by specularities. We choose $\lambda = C/\sqrt{m}$, and vary the value of C. We evaluate the results using angular error with respect to the ground-truth normal map. We observe from Table 13.4 that the choice of C influences the accuracy of the estimated normal map. For real-world applications, where

the data is typically noisy, the choice of λ could play an important role in the efficacy of RMC.

C	1.0	0.8	0.6	0.4
Mean error (in degrees)	1.42	0.78	0.19	0.029
Max. error (in degrees)	8.78	8.15	1.86	0.91

Table 13.4. **Handling more specularities by appropriately choosing λ.** We use 40 images of Caesar under different lighting conditions with about 28% specularities and 20% shadows, and set $\lambda = C/\sqrt{m}$.

f. Computation.

The core computation of RMC is solving a convex program (13.3.1). For the specular Caesar data (Fig. 13.5(b)) with 40 images of 450×350 resolution, a single-core MATLAB implementation of RMC takes about 7 minutes on a Macbook Pro with a 2.8 GHz Core 2 Duo processor and 4 GB memory, as against 42 seconds taken by the LS approach. While RMC is slower than the LS approach, it is much more accurate in a wide variety of scenarios and is more efficient than other methods (e.g. [Miyazaki et al., 2010]).

13.4.2 Qualitative Evaluation with Real Images

We now test the algorithms on real images. We use a set of 40 images of a toy Doraemon and Two-face taken under different lighting conditions (see Fig. 13.9(a), (d)). A glossy sphere was placed in the scene for light source calibration when capturing the data. We used a Canon 5D camera with the RAW image mode.[9] These images present new challenges to RMC. In addition to shadows and specularities, there is potentially additional noise inherent to the acquisition process as well as possible deviations from the idealistic Lambertian model illuminated by distant lights. In this experiment, we use a threshold of 0.01 to detect shadows in images.[10] We also found experimentally that setting $\lambda = 0.3/\sqrt{m}$ works well for these datasets.

Since the ground truth normal map is not available for these scenes, we compare the RMC method and the LS approach by visual inspection of the output normal maps shown in Fig. 13.9(b),(c),(e),(f). We observe that the normal map estimated by RMC appears smoother and hence, more realistic. This can be observed particularly around the necklace area in

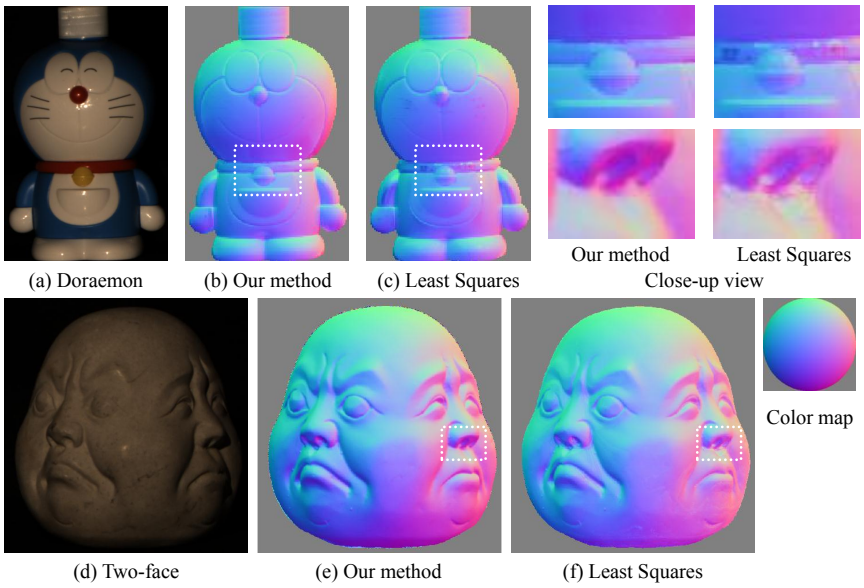


Figure 13.9. **Qualitative comparison on real data.** We use images of Doraemon and Two-face taken under 40 different lighting conditions to qualitatively evaluate the performance of the RMC method against the LS approach. (a),(d) Sample input images. (b),(e) Normal map estimated by RMC. (c),(f) Normal map estimated by Least Squares. Close-up views of the dotted rectangular areas are shown on the top-right.

Doraemon and nose area in Two-face (see Fig. 13.9) where the LS estimate exhibits some discontinuity in the normal map.

13.5 Notes and References

It is well understood that when a Lambertian surface is illuminated by at least three known lighting directions, the surface orientation at each visible point can be uniquely determined from its intensities. From different perspectives, it has long been shown that if there are no shadows, the appearance of a convex Lambertian scene illuminated from different lighting directions span a three-dimensional subspace [Shashua, 1992] or an illumination cone [Belhumeur and Kriegman, 1996]. Basri and Jacobs [Basri and Jacobs, 2003b] and Georghiades *et al.* [Georghiades et al., 2001b] have further shown that the images of a convex-shaped object with cast shad-

[9] We did not apply Gamma correction.
[10] All pixels are normalized to have intensity between 0 and 1.

ows can also be well-approximated by a low-dimensional linear subspace. The aforementioned works indicate that there exists a degenerate structure in the appearance of Lambertian surfaces under variation in illumination. This is the key property that all photometric stereo methods harness to determine the surface normals.

Previously, photometric stereo algorithms for Lambertian surfaces generally find surface normals as the *Least Squares* solution to a set of linear equations that relate the observations and known lighting directions, or equivalently, try to identify the low-dimensional subspace using conventional Principal Component Analysis (PCA) [Jolliffe, 1986]. Such a solution is known to be optimal if the measurements are corrupted by only *i.i.d.* Gaussian noise of small magnitude. Unfortunately, in reality, photometric measurements rarely obey such a simplistic noisy linear model: the intensity values at some pixels can be severely affected by specular reflections (deviation from the basic Lambertian assumption), sensor saturations, or shadowing effects. As a result, the Least Squares solution normally ends up with incorrect estimates of surface orientations in practice.

To overcome this problem, researchers have explored various heuristic approaches to eliminate such deviations by treating the corrupted measurements as outliers, e.g., using a RANSAC scheme [Fischler and Bolles, 1981, C. Hernández and Cipolla, 2008], or a median-based approach [Miyazaki et al., 2010]. To identify the different types of corruptions in images more carefully, Mukaigawa *et al.* [Mukaigawa et al., 2001,Mukaigawa et al., 2007] have proposed a method for classifying diffuse, specular, attached, and cast shadow pixels based on RANSAC and outlier elimination.

The method presented in this chapter was first introduced through the work [Wu et al., 2010]. In contrast to previous robust approaches, this method is computationally more efficient and provides theoretical guarantees for robustness to large errors. More importantly, the method is able to use all the available information simultaneously for obtaining the optimal result, instead of pre-processing measurements which might discard useful information, e.g., by either selecting the best set of illumination directions [C. Hernández and Cipolla, 2008] or using the median estimator [Miyazaki et al., 2010].

The method can also be used to improve virtually any existing photometric stereo method, including uncalibrated photometric stereo [Hayakawa, 1994], where traditionally, corruption in the data (e.g., by specularities) is either neglected or ineffectively dealt with conventional heuristic robust estimation methods.

Exercises

13.1 (New Exercise)**.**