

(CV PA1) Robust Photometric Stereo

20225092 Huigyoeng Son

Abstract

본 프로젝트에서는 다중 이미지로부터 물체의 3차원 표면 정보를 정밀하게 추정하기 위해 Robust Photometric Stereo 파이프라인을 구현하였다. 먼저, 물체가 Lambertian 반사 특성을 갖는다고 가정하고, Least Squares 기반의 전통적인 Photometric Stereo 기법을 구현하였다. 이후, 실제 환경에서 발생할 수 있는 shadow와 highlight와 같은 Non-Lambertian outlier 요소들을 효과적으로 제거하기 위해, RPCA(Robust Principle Components Analysis)를 활용한 **Robust Photometric Stereo**를 구현하였다. 실험은 고정된 위치와 서로 다른 11개의 조명 조건 하에서 촬영된 네 가지 물체(moai, nike, choonsik, toothless)에 대해 수행되었으며, 각 기법을 통해 추정한 albedo map과 surface normal을 기반으로 새로운 조명 조건에서의 relighting을 수행하여 결과를 정량적·정성적으로 검증하였다. 추가로 우리는 relighting 성능을 향상시키기 위해, 생성된 이미지에 **Photometric Factor Classification**를 수행한 후, 각 factor별로 weight를 다르게 설정하여 밝기를 조정하는 방식을 도입하였다. Least Squares 기반의 전통적인 방법에 비해 Robust Photometric Stereo를 적용하였을 때 surface normals가 부드럽게 나오며 albedo map의 이상치가 감소하는 효과를 확인할 수 있었고, Photometric factor에 따른 보정으로 relighting 오차를 효과적으로 줄일 수 있었다. 해당 프로젝트에 사용한 코드는 다음의 github에서 확인할 수 있다: <https://github.com/gyoenge/cvpa1-robust-photometric-stereo>.

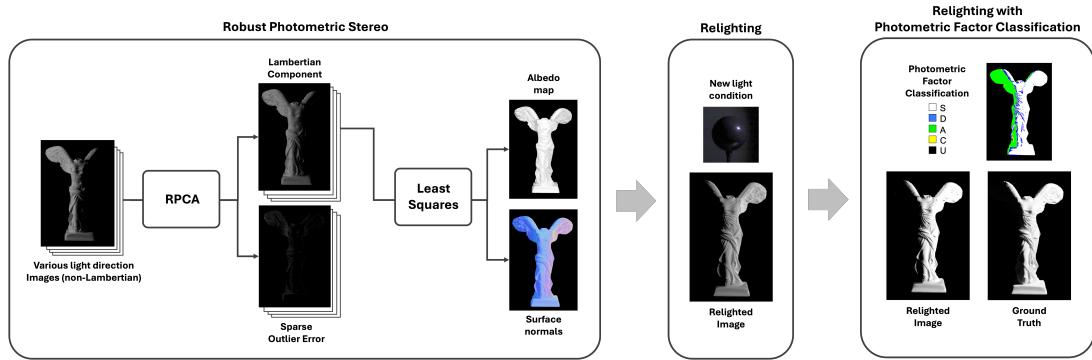


Figure 1: Overview framework of Robust Photometric Stereo and Relighting

1 Introduction

Photometric stereo 문제는 같은 물체를 고정된 위치에서 촬영하되, 조명만 다르게 설정한 여러 이미지를 사용해서, 그 물체의 정확한 surface normals을 계산하고, 나아가 3D 형상(shape)을 복원하는 기법이다 [1]. 기존의 전통적인 stereo 기법은 여러 방향에서 촬영된 물체 이미지에서 대응점을 찾아 triangulation으로 물체까지의 거리를 계산하는데, 이 대응 점을 구하기 까다롭다는 단점이 있다. Photometric stereo는 물체를 고정된 위치에서 촬영하여 imaging geometry가 변하지 않으므로, 대응점을 매칭하는 과정을 거치지 않아도 된다. 대신에 조명 조건을 촬영마다 달리하는데, 이 달라지는 조명 정보와 이에 따른 물체의 빛 반사 정보를 이용해, 물체의 surface normals를 계산할 수 있다. 이렇게 얻은 물체의 surface normals은 3D shape recovery, depth refinement, image relighting 등에 활용할 수 있다.

일반적으로 Lambertian 물체에 대한 Photometric stereo는 Least Squares 방법으로 해결할 수 있다. Lambertian 물체는 각도와 관계없이 일정한 밝기로 보이는 표면을 갖는 물체이다. 즉, 빛이 물체 표면에서 반사될 때, 이상적인 확산 반사에 의해 모든 방향으로 빛이 균일하게 퍼지는 것이다. 예시로는 종이, 콘크리트 등과 같이 대개 표면이 거칠고 광택이 없는 재질을 가지는 물체들이 있다. 이상적인 확산 반사를 가정하기 때문에, 물체를 촬영한 이미지에서의 물체 각 지점의 밝기는 해당 표면에서 반사된 복사회도라고 생각할 수 있다. 따라서, 관찰된 복사회도는 다음과 같이 물체에 입사하는 빛에 대한 식으로 곧바로 표현할 수 있다:

$$L_r = \rho L_i \cos \theta_i = \rho L_i (L \cdot N) \quad (1)$$

여기서 L_r 은 표면에서 관찰자 방향으로 반사되는 복사회도 (reflected radiance), L_i 는 광원으로부터 입사하는 복사회도 (incident radiance), θ 는 입사각, L 은 광원이 향하는 단위 벡터 (빛의 방향 벡터), N 은 표면의 단위 법선 벡터 (surface normals), ρ 는 표면의 반사율 또는 확산 반사 계수 (albedo)이다. 여기서 albedo는 Lambertian의 BRDF(Bidirectional

Reflectance Distribution Function) — 특정한 입사 방향에서 들어온 빛이 특정한 반사 방향으로 얼마나 많이 반사되는가를 나타내는 함수 — 의 한 종류로, 방향에 무관하게 항상 일정한 값을 가짐을 표현하고 있다. Photometric stereo는 빛에 대한 정보를 모두 알고 있다고 가정하기 때문에, 위 식에서 (L_i, L, L_r) 조합들로 선형 시스템을 구성할 수 있다. 따라서 우리는 위 식을 Least Squares 방법을 이용하여 풀 수 있고, 결과적으로 N 과 ρ 를 구할 수 있다.

그러나, non-Lambertian 물체는 그림자(shadow)나 광택(specular highlight)와 같은 관측값 상의 이상치(outlier)가 존재하기 때문에, 바로 Least Squares 방법을 적용하면 해당 outlier들에 의해 부정확한 결과가 나온다. 여기서 non-Lambertian 물체는 Lambertian과는 다르게 표면의 반사 특성이 입사각과 반사각에 따라 달라지며, 유리나 금속처럼 방향성 있는 반사를 보이는 물체를 말한다. non-Lambertian 문제를 해결하기 위해, RPCA(Robust Principle Components Analysis)기법을 도입할 수 있고, 이 방법을 ‘Robust Photometric Stereo’라고 한다 [2, 3, 6]. RPCA는 관측된 radiance map을 low-rank 성분(Lambertian component)과 sparse 성분(outlier error)로 분리하는 방식이다:

$$D = NL \text{ (low-rank)} + E \text{ (sparse)} \quad (2)$$

즉, specular highlight나 shadow와 같은 이상치는 sparse error로 모델링되고, 나머지 부분은 low-rank 구조를 갖는 Lambertian 성분으로 간주된다. 이상치를 제거한 후의 Lambertian diffuse 성분에 대해서는 기존의 Least Squares 기법을 안정적으로 적용할 수 있으며, 이를 통해 non-Lambertian 물체에 대해서 보다 정확하고 강건한 Photometric Stereo 결과를 얻을 수 있다. RPCA 문제를 효율적으로 풀기 위해 IALM(Inexact Augmented Lagrange Multiplier) 알고리즘을 사용하였다 [4, 5]. IALM은 RPCA의 각 반복 단계에서 보조 문제를 정확하게 풀지 않고(inexact) 근사 해를 사용함으로써 계산 비용을 줄이면서도 빠른 수렴 속도를 달성하는 최적화 기법이다. 결과적으로, RPCA 기반의 Robust Photometric Stereo는 여러 비이상적인 표면 특성을 갖는 물체에 대해서도 안정적인 3차원 표면 추정을 가능하게 한다.

이렇게 추정된 albedo와 surface normal을 기반으로, 입력 이미지에 존재하지 않는 조명 조건을 가정한 relighting을 수행하였다. 이를 통해 복원된 3차원 정보의 품질을 정량적·정성적으로 평가하고, 단순 Least Squares 및 Robust Photometric Stereo 기법 간의 성능 차이를 비교하였다. relighting은 식 (1)을 통해 주어진 L, L_i, ρ, N 을 이용해 L_r 을 구하는 과정으로 수행될 수 있다. 정량적인 평가 지표로 MSE(Mean Squares Error)를 사용하였고, 정성적으로는 relighted 이미지와 ground truth 이미지 간의 시각적 유사도를 비교하였다. 더 나아가, relighting 성능을 높이기 위해 추가로 보정하는 방식을 도입하였는데, relighted 이미지에 Photometric Factor Classification[3]을 수행한 후, 각 factor별로 weight 을 다르게 설정하여 밝기를 조정하는 것이다. highlight 부분은 더욱 밝게, shadow 부분은 더욱 어둡게 처리하여, RPCA 과정에서 제거되었던 non-Lambertian에 대한 outlier 정보를 추가해주는 과정으로 생각될 수 있다.

우리는 총 4가지 물체(moai, nike, choonsik, toothless)를 Robust Photometric Stereo 실험에 사용하였다:

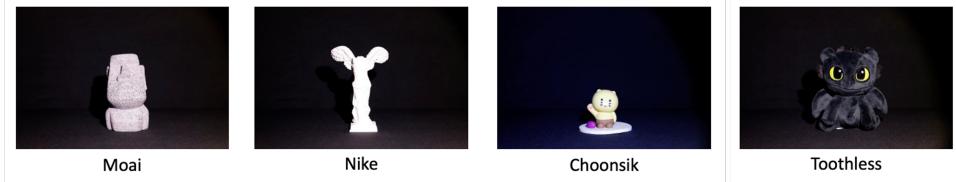


Figure 2: 4 Objects — Moai, Nike, Choonsik, Toothless

각 물체에 대해 고정된 위치와 11개의 조명 조건 하에서 촬영된 사진으로부터 albedo map과 surface normals를 구할 수 있었고, 이렇게 얻은 3차원 표면 정보를 이용해 새로운 빛에 대한 relighting을 수행하여 결과를 확인하였다.

2 Method

2.1 Recover Light Direction

Photometric stereo는 물체에 비춰지는 조명 빛의 정보를 알고 있다고 가정한다. 우리 실험에서는 11개의 조명 조건에서의 빛 방향 벡터를 구하기 위해 chromeball의 복사획도 이미지를 사용하였다 (Fig3(a)).

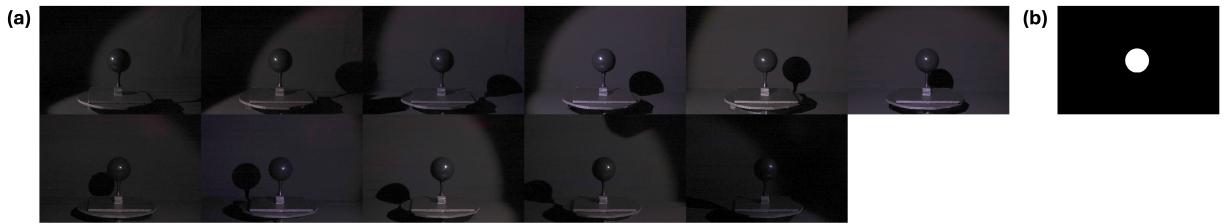


Figure 3: (a) 11 different light conditions on chromeballs (구체 식별을 위해 밝게 조정하여 첨부), (b) chromeball mask

각각의 사진에 대해 Fig(b)의 chromeball mask를 적용하면, ball의 영역에 해당하는 복사획도 맵을 구할 수 있다. 이 맵에서 가장 밝은 지점을 찾는 것이 핵심이다. 가장 밝은 지점은 복사획도 맵에서 x, y 각 축에 대해 최대 복사획도 값을 갖는 점들을 찾은 후 해당 점들 좌표의 평균값을 계산하여 구할 수 있었다. chromeball은 완벽한 구체이므로, Figure 4 와 같이 간단한 light calibration 모형으로 빛 방향 벡터를 구할 수 있다. 필요한 정보는 이미지 픽셀 좌표계 기준으로 chromball의 반지름 r , 중심 좌표 C , 그리고 가장 밝은 지점의 좌표 P 이다. 반지름은 chromball mask에서 x, y 각 축에 대해 (최대좌표값 - 최소좌표값)/2로 구할 수 있고, 중심 좌표는 각 축에 대해 (좌표값의 평균)으로 구할 수 있었다. 이제 3

차원 좌표계로 옮겨 방향 벡터들의 관계를 생각할 수 있다. 먼저, 가장 밝은 지점의 방향 벡터 N 을 다음과 같이 구할 수 있다: $N_x = (P_x - C_x)/r$, $N_y = (C_y - P_y)/r$ (이미지 픽셀상의 y축과 방향이 반대인 것을 고려), $N_z = (\sqrt{r^2 - N_x^2 - N_y^2})/r$. 광선이 구 표면에 닿아 반사되는 방향 (reflection direction) R 은 카메라 정면 방향이라고 가정하여 $[0, 0, 1]$ 로 둘 수 있다. 그러면 입사광의 방향을 반사 법칙에서 유도된 다음 수식을 통해 구할 수 있다: $L = 2(N \cdot R)N - R$.

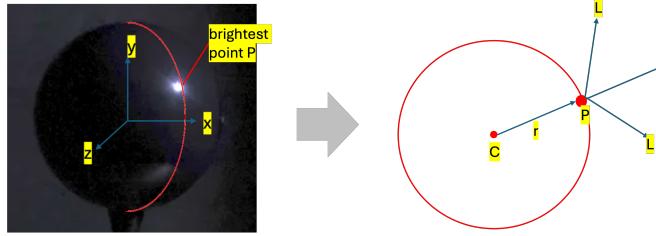


Figure 4: Chromeball Light Calibration

2.2 Lambertian Least Squares

Lambertian 물체임을 가정하였을 때 Photometric stereo 문제는 단순 Least Squares 방법으로 해결될 수 있다. 식(1)에서 $L_i = 1$ 이라 하고 L_r 을 I 로 두어 다시 식을 다음과 같이 나타낼 수 있다:

$$I = \rho(L \cdot N) = L \cdot (\rho N) = L \cdot G \quad (\text{where } G = \rho N) \quad (3)$$

각각의 행렬의 차원은 $I \in \mathbb{R}^{n \times m}$, $L \in \mathbb{R}^{n \times 3}$, $N \in \mathbb{R}^{3 \times m}$ 이다. 식 (3)에서 I 와 L 이 priori로 주어지기 때문에 $G = L^{-T}I$ 를 구할 수 있다. Least Squares을 이용해, 다음과 같이 목적함수 $\|I - L^T G\|^2$ 를 최소화하는 문제로 대응시킬 수 있다:

$$\min_G \|I - L^T G\|^2 = \min_G I^T I + G^T L L^T G - 2G^T L I \quad (4)$$

이후 전개된 목적함수에서 G 에 대한 미분값을 0으로 두어 최종 해를 구할 수 있다:

$$2L L^T G - 2L I = 0, \quad G = (L L^T)^{-1} L I \quad (5)$$

마지막으로 G 를 다시 ρ 와 N 으로 분해하면, $\rho = \|G\|$, $N = \frac{G}{\|G\|}$ 으로 albedo와 surface normal을 구할 수 있다. $n \geq 3$ 의 서로 다른 lighting direction에 대한 정보가 주어진다면 해당 값들을 추정할 수 있는 것이다. 여기서 $G = (L L^T)^{-1} L I$ 를 실제로 구현할 때는 G, L, I 의 차원에 맞게 적절히 수정하여 사용하였다.

2.3 Robust PCA with IALM

non-Lambertian 물체에도 잘 적용되는 Photometric stereo를 구현하기 위해 RPCA(Robust Principle Components Analysis)를 도입할 수 있다. RPCA의 핵심은 주어진 이미지 D 를 Low-rank A 와 sparse error E 의 두 부분으로 나누는 것이다. 일반적인 PCA(Principal Components Analysis)는 데이터 행렬을 저차원의 선형 공간으로 근사하는 방법으로, 일반적으로 데이터 내의 주요 구조를 유지하면서 노이즈를 제거하는 데 유용하다. 그러나 일반 PCA는 이상치(outlier)에 민감하다는 단점이 있다. 즉, shadow나 specular highlight와 같은 non-Lambertian 성분이 존재하는 경우, 이들이 전체 모델에 큰 영향을 미쳐 정확한 low-rank 근사를 방해하게 된다. 이에 반해 RPCA는 데이터 행렬 D 를 $A + E$ 로 분해함으로써, 이러한 이상치 E 를 효과적으로 분리한다. 여기서 A 는 rank가 낮은 행렬로, Lambertian한 물체의 일반적인 반사 특성을 나타내며, E 는 희소(sparse) 행렬로, 주로 비정상적인 non-Lambertian 요소들(그림자, 반사, 노이즈 등)을 포함한다.

$$D = A \text{ (low-rank lambertian properties)} + E \text{ (spase error non-lambertian outliers)} \quad (6)$$

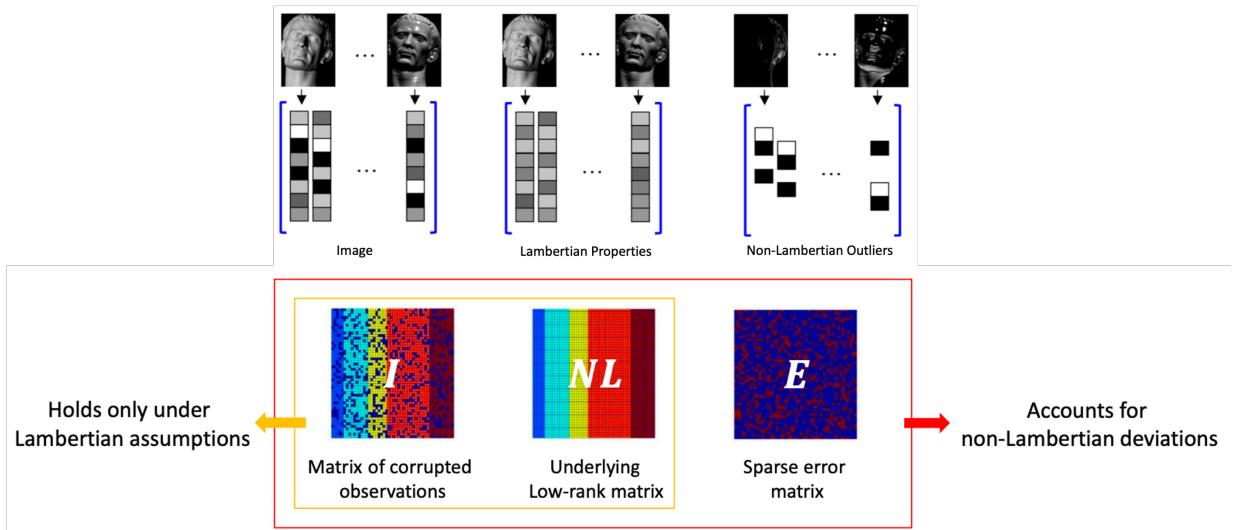


Figure 5: Robust PCA

위의 식(6)의 RPCA는 lagrangian formulation을 통해, 다음과 같은 convex optimization 문제로 나타내어질 수 있다:

$$\min_{A, E} \text{rank}(A) + \gamma \|E\|_0 \quad \text{subject to } D = A + E \quad (7)$$

여기서 $\|\cdot\|_0$ 는 ℓ_0 -norm(number of non-zero entries in the matrix)를 의미하고, $\gamma > 0$ 는 A 의 rank와 E 의 sparsity 사이의 trade off를 조절하는 parameter이다. 그러나 이 식(7)에 해당하는 최적화 문제는 풀기에 매우 어려운 NP-hard 문제이다. 따라서 이를 적절히 풀기 좋은 형태로 변형해야 할 필요가 있다. 앞선 연구들에 따르면[3, 4, 5, 6], 위 식(7)의 cost function 을 이의 convex surrogate로 대체하면 해결될 수 있다. $\text{rank}(\cdot)$ 은 nuclear norm(sum of the singular values of the matrix)로 대체하고, ℓ_0 -norm은 ℓ_1 -norm(sum of the absolute values of all entries of the matrix)로 바꾸는 것이다. 그러면 다음과 같은 식의 최적화 문제로 변형된다:

$$\min_{A, E} \|A\|_* + \lambda \|E\|_1 \quad \text{subject to } D = A + E \quad (8)$$

여기서 $\|A\|_*$ 는 nuclear norm으로 low-rank 성분을 유도하고, $\|E\|_1$ 은 element-wise ℓ_1 norm으로 sparsity를 유도한다. λ 는 weighting parameter이며, 이론적으로 $C/\sqrt{\max\{m, n\}}$ (C : constant)의 형태로 나타내져야 한다. 주목할 점은 식(7) 최적화 문제와 식(8) 최적화 문제 간의 equivalence는 A 해의 singular values의 magnitude와 E 해의 non-zero entries의 magnitude에 영향을 받지 않고 유지된다는 것이다.

이 최적화 문제를 효율적으로 풀기 위해 본 프로젝트에서는 IALM(Inexact Augmented Lagrange Multiplier) 알고리즘을 사용하였다. 먼저, 식(8) 최적화 문제에 ALM(Augmented Lagrangian Multiplier) method를 적용하면 다음과 같다:

$$\mathcal{L}_\mu(A, E, Y) = \|A\|_* + \lambda \|E\|_1 + \langle Y, D - A - E \rangle + \frac{\mu}{2} \|D - A - E\|_F^2 \quad (9)$$

여기서 $Y \in \mathbb{R}^{m \times n}$ 은 Lagrange multiplier matrix이고, μ 는 positive constant이며, $\langle \cdot, \cdot \rangle$ 은 matrix inner product이고, $\|\cdot\|_F$ 는 Frobenius norm이다. Y 의 적절한 선택과 충분히 큰 μ 에 대해, 이 최적화 문제는 원래의 constrained 최적화 문제와 같아진다. 이 ALM method는 Lagrange multiplier와 optimal solution을 추정하기 위해 interative한 접근으로 해결할 수 있다. 기본적인 ALM iteration은 다음과 같다:

$$\begin{cases} (A_{k+1}, E_{k+1}) = \operatorname{argmin}_{A, E} \mathcal{L}_{\mu_k}(A, E, Y_k) \\ Y_{k+1} = Y_k + \mu_k(D - A_{k+1} - E_{k+1}) \\ \mu_{k+1} = \rho \cdot \mu_k \end{cases} \quad (10)$$

여기서 $\{\mu_k\}$ 는 monotonically increasing positive sequence ($\rho > 1$)이다. 식(10)의 첫 번째 단계는 다음과 같이 alternating minimization strategy를 적용할 수 있다:

$$\begin{cases} E_{j+1} = \operatorname{argmin}_E \lambda \|E\|_1 - \langle Y_k, E \rangle + \frac{\mu_k}{2} \|D - A_j - E\|_F^2 \\ A_{j+1} = \operatorname{argmin}_A \|A\|_* - \langle Y_k, A \rangle + \frac{\mu_k}{2} \|D - A - E_{j+1}\|_F^2 \end{cases} \quad (11)$$

먼저 식(11)의 첫 번째 step을 풀기 위해 shrinkage (or soft-thresholding) operator for scalars를 정의해줄 수 있다:

$$\operatorname{shrink}(x, \alpha) = \operatorname{sign}(x) \cdot \max\{|x| - \alpha, 0\} \quad (12)$$

여기서 $\alpha \geq 0$ 이며, shrink operator가 vector나 matrix에 적용될 때는 각 원소별로 수행될 수 있다. 그러면, 첫 번째 step 은 다음과 같이 closed-form solution을 구할 수 있다:

$$E_{j+1} = \operatorname{shrink}(D + \frac{1}{\mu_k} Y_k - A_j, \frac{\lambda}{\mu_k}) \quad (13)$$

두 번째 step은 closed-form solution을 구하는 것이 불가능하며, APG(Accelatated Proximal Gradient) method를 이용할 수 있다:

$$\begin{cases} (U_i, \Sigma_i, V_i) = \operatorname{svd}(\frac{1}{\mu_k} Y_k + D - E_{j+1} + Z_i) \\ A_{i+1} = U_i \operatorname{shrink}(\Sigma_i, \frac{1}{\mu_k}) V_i^T \\ Z_{i+1} = A_{i+1} + \frac{t_i - 1}{t_i + 1} (A_{i+1} - A_i) \end{cases} \quad (14)$$

여기서 $\operatorname{svd}(\cdot)$ 은 singular value decomposition operator를 의미하며, $\{t_i\}$ 는 $t_1 = 1$, $t_{i+1} = 0.5(1 + \sqrt{1 + 4t_i^2})$ 을 만족 하는 positive sequence이다. 위 식(14)에 기반한 전체 절차는 Algorithm 1에 정리되어 있다. 해당 방식은 다음과 같은 sub-problem을 내부 nested-loop으로 해결하는 Exact ALM(EALM) 방식이다:

$$(A_{k+1}^*, E_{k+1}^*) = \operatorname{argmin}_{A, E} L(A, E, Y_k^*, \mu_k) \quad (15)$$

그러나 이는 nested-loop로 인해 수렴 속도가 느린다. 심지어는 수학적으로 접근하였을 때 몇몇 정리에 의하여[4, 5], 해당 sub-problem을 풀 필요가 없다라는 것이 밝혀졌다. sub-problem으로 푸는 것 대신에, A_k 와 E_k 를 한 번씩만 update하여도

A_k, E_k 는 RPCA 문제의 optimal solution으로 수렴할 수 있다. 이것은 Inexact ALM(IALM)이라고 불리며, Algorithm 2에 전체 절차가 정리되어 있다.

Algorithm 1 RPCA via Exact ALM (EALM) method

Require: Observation matrix $D \in \mathbb{R}^{m \times n}$, $\lambda > 0$

- 1: Initialize $A_1 \leftarrow 0$, $E_1 \leftarrow 0$, $Y_1 \leftarrow 0$
- 2: **for** $k = 1, 2, \dots$ until convergence **do**
- 3: $A_{k,1} \leftarrow A_k$, $E_{k,1} \leftarrow E_k$
- 4: **for** $j = 1, 2, \dots$ until convergence **do**
- 5: $E_{k,j+1} \leftarrow \text{shrink}\left(D + \frac{1}{\mu_k} Y_k - A_{k,j}, \frac{\lambda}{\mu_k}\right)$
- 6: $t_1 \leftarrow 1$, $Z_1 \leftarrow A_{k,j}$, $A_{k,j,1} \leftarrow A_{k,j}$
- 7: **for** $i = 1, 2, \dots$ until convergence **do**
- 8: $(U_i, \Sigma_i, V_i) \leftarrow \text{svd}\left(\frac{1}{\mu_k} Y_k + D - E_{k,j+1} + Z_i\right)$
- 9: $A_{k,j,i+1} \leftarrow U_i \cdot \text{shrink}\left(\Sigma_i, \frac{1}{\mu_k}\right) \cdot V_i^T$
- 10: $t_{i+1} \leftarrow 0.5 \left(1 + \sqrt{1 + 4t_i^2}\right)$
- 11: $Z_{i+1} \leftarrow A_{k,j,i+1} + \frac{t_i - 1}{t_{i+1}}(A_{k,j,i+1} - A_{k,j,i})$
- 12: $A_{k,j+1} \leftarrow A_{k,j,i+1}$
- 13: **end for**
- 14: $A_{k+1} \leftarrow A_{k,j+1}$, $E_{k+1} \leftarrow E_{k,j+1}$
- 15: **end for**
- 16: $Y_{k+1} \leftarrow Y_k + \mu_k(D - A_{k+1} - E_{k+1})$
- 17: $\mu_{k+1} \leftarrow \rho \cdot \mu_k$
- 18: **end for**

Ensure: $\hat{A} = A_k$, $\hat{E} = E_k$

Algorithm 2 RPCA via Inexact ALM (IALM) Method

Require: Observation matrix $D \in \mathbb{R}^{m \times n}$, $\lambda > 0$

- 1: $Y_0 \leftarrow D/J(D)$; $E_0 \leftarrow 0$; $\mu_0 > 0$; $\rho > 1$; $k \leftarrow 0$
- 2: **while** not converged **do**
- 3: // Lines 4-5 solve $A_{k+1} = \arg \min_A L(A, E_k, Y_k, \mu_k)$
- 4: $(U, S, V) \leftarrow \text{svd}(D - E_k + \mu_k^{-1} Y_k)$
- 5: $A_{k+1} \leftarrow U \cdot \mathcal{S}_{\mu_k^{-1}}[S] \cdot V^T$
- 6: // Line 7 solves $E_{k+1} = \arg \min_E L(A_{k+1}, E, Y_k, \mu_k)$
- 7: $E_{k+1} \leftarrow \mathcal{S}_{\lambda \mu_k^{-1}}[D - A_{k+1} + \mu_k^{-1} Y_k]$
- 8: $Y_{k+1} \leftarrow Y_k + \mu_k(D - A_{k+1} - E_{k+1})$
- 9: Update μ_k to μ_{k+1}
- 10: $k \leftarrow k + 1$

11: end while

Ensure: $\hat{A} = A_k$, $\hat{E} = E_k$

Figure 6: Algorithm 1. EALM method, Algorithm 2. IALM method

IALM은 각 반복 단계에서 식(16)에 해당하는 sub-problem를 정확하게 풀지 않고(inexact), 근사 해를 사용함으로써 계산 효율성을 크게 향상시키는 장점이 있다. IALM의 Algorithm 2에 대해, Y_0 의 initialization으로 $D/J(D)$ 을 사용하는데, 여기서 $J(D)$ 는 $J(D) = \max(\|D\|_2, \lambda^{-1}\|D\|_\infty)$ 이다. hyperparameter μ_0 , ρ , λ 의 initialization은 [5, 6]에서 사용한 것을 따라, 각각 $1.25/\|D\|_2$, 1.6 , $1/\sqrt{m}$ 으로 설정하였다. 수렴 확인 조건 또한 [5]에서 사용한 다음 수식을 사용하였다 :

$$\begin{cases} \|D - A_k - E_k\|_F / \|D\|_F < \varepsilon_1 \\ \text{dist}(\partial\|A_k\|_*, \partial(\|\lambda E_k\|_1)) / \|D\|_F \approx \mu_{k-1} \cdot \|E_k - E_{k-1}\|_F / \|D\|_F < \varepsilon_2 \end{cases} \quad (16)$$

여기서 [5]에 따라 ε_1 으로 10^{-7} 을 사용하였고, ε_2 로 10^{-5} 를 사용하였다.

최종적으로 IALM method로 구한 RPCA의 해 (\hat{A}, \hat{E}) 에 대해, \hat{A} 으로 section 2.2에서 다른 Least Squares 방법을 $N = \hat{A}\hat{A}^\dagger$ 와 같이 사용할 수 있다. 이를 통해 non-Lambertian 환경에서도 강건한 Photometric stereo 수행이 가능해진다.

2.4 Applying Relighting and Evaluation

section 2.2와 2.3의 결과로 물체의 albedo map과 surface normals를 구할 수 있었다. 이제 이들을 이용하면, 새로운 빛 환경에 대해 relighting된 물체 이미지를 얻을 수 있다. 식(3)에 L , ρ , N 을 넣어 I 를 구하는 것이다. 이때 조명의 빛 방향 벡터 L 은 주어진다고 가정한다.

이후 relighting한 결과를 정량적, 정성적으로 평가하였다. 정량적인 평가 지표로는 MSE(Mean Squared Error)를 사용하였다:

$$MSE = \sum_{i=0}^m (\text{GroundTruth}[i] - I[i])^2 \quad , \text{ where } i \text{ is masked object pixel point} \quad (17)$$

정성적인 평가로는 relighted 이미지와 ground truth 이미지 간의 시각적인 유사도를 비교하였다.

2.5 Relighting with Photometric Factor Classification

relighting 성능을 높이기 위해 추가로 보정하는 방식을 도입하였는데, relighted 이미지에 Photometric Factor Classification[3]을 수행한 후, 각 factor별로 weight을 다르게 설정하여 밝기를 조정하는 것이다. highlight 부분은 더욱 밝게, shadow 부분은 더욱 어둡게 처리하여, RPCA 과정에서 제거되었던 non-Lambertian에 대한 outlier 정보를 추가해주는 과정으로 생각될 수 있다.

먼저 PFC(Photometric Factor Classification)을 살펴보자. PFC는 물체 이미지 영역을 총 4개로 나눈다: C (cast shadow), A (attached shadow), D (diffuse reflection), S (specular reflection). 물체 이미지 영역이 아닌 곳은 따로 U (undefined regions)로 분류한다. 이러한 영역 분류를 나누는 기준은 [3]에 따라 다음과 같이 나타낼 수 있다:

$$\begin{cases} C &= \{i | F_i > 0, O_i \leq T_1\} \\ A &= \{i | F_i \leq 0, O_i \leq T_1\} \\ D &= \{i | |F_i - O_i| < T_2 \cdot O_i, O_i > T_1\} \\ S &= \{i | O_i - F_i > T_2 \cdot O_i, O_i > T_1\} \end{cases} \quad (18)$$

여기서 thresholds에 해당하는 $T_1, T_2 \in [0, 1]$ 는 물체에 따라 경험적으로 설정할 수 있다. 우리는 toothless에 대해 $[0.05, 0.3]$ 를, 나머지는 $[0.1, 0.15]$ 로 설정했다. 아래 Figure7은 PFC를 수행하여 분류된 영역을 시각화한 예시 결과이다.

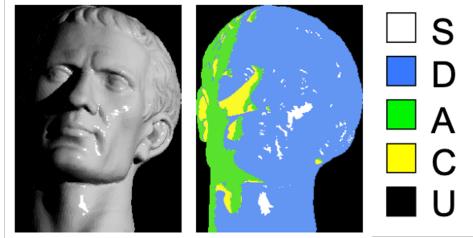


Figure 7: example PFC(Photometric Factor Classification) result

PFC를 적용한 다음, section 2.3과 2.4를 거쳐 relighting된 이미지 I 를 다시 영역별로 적절한 weight을 줘서 보정할 수 있다. 이를 식으로 나타내면 다음과 같다 :

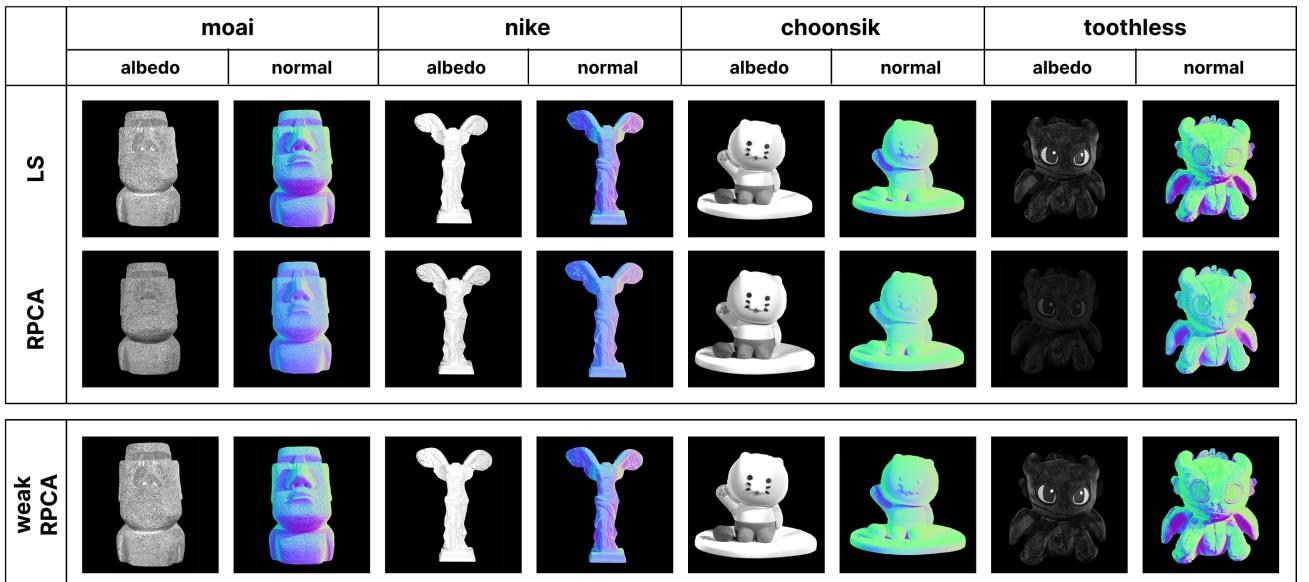
$$I \leftarrow I * w_{area \in [\mathcal{C}, \mathcal{A}, \mathcal{D}, \mathcal{S}]} \quad (19)$$

각 영역별 weight는 경험적으로 $w_C = 0.1, w_A = 0.2, w_D = 1.1, w_S = 1.3$ 로 설정할 수 있었다. 결과적으로, cast shadow와 attached shadow에 해당하는 영역은 더 어둡게 보정되고, diffuse reflection에 해당하는 영역은 조금 밝게, specular reflection에 해당하는 영역은 더 밝게 보정되는 효과를 얻을 수 있다. 그리고 이는 relighting MSE 오차를 줄이는 데에 기여할 수 있다.

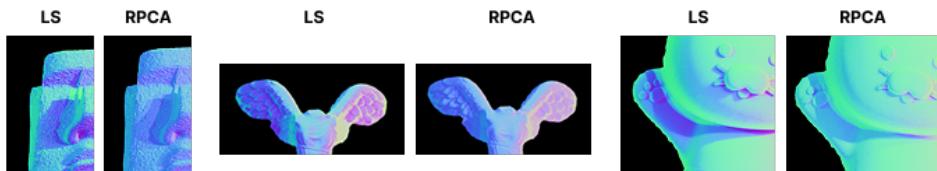
3 Result and Discussion

3.1 Least squares vs. Robust PCA

LS(Least Squares)만을 사용한 Photometric stereo와 RPCA(Robust PCA)를 도입한 Photometric stereo의 결과를 비교하기 위해, Figure 8에 결과를 시각화하여 정리하였다. 여기서 RPCA는 앞선 method section에서 다루었던 hyperparameter와 구현을 그대로 따른 것이고, weak RPCA는 hyperparameter μ_0, ρ 에 대해 각각 $\mu_0 = 1.0, \rho = 1.1$ 로 두고 RPCA를 훈련 버전이다. weak RPCA는 원본 RPCA보다 $\{\mu_k\}$ 가 증가하는 속도가 느리기 때문에 수렴 속도가 늦춰진 형태이다.



(a) result of albedo and surface normal (LS vs. RPCA)



(b) magnified normal map (LS vs. RPCA)

Figure 8: Comparison of estimation using LS and RPCA

결과적으로 각 물체의 albedo와 surface normals를 보면, LS보다 RPCA에서 더 매끄럽고 부드러운 surface normals를 가지며, albedo map에서의 highlight와 shadow에 대한 이상치 또한 감소한 것을 확인할 수 있다. 각각의 물체를 자세히 살펴보자. moai에서의 surface normals는 RPCA를 적용함으로써 좌우 외곽의 높은 highlight가 완화되었고, 코 옆과 아래 그리고 턱 밑의 shadow가 완화되었으며, 전체적인 표면이 부드럽게 변화하였다. moai의 albedo에서도 highlight와 shadow

가 완화된 것을 볼 수 있고, 전체적으로 밝기가 감소한 모습을 볼 수 있다. nike에서의 surface normals는 RPCA를 적용함으로써 특히 표면의 굴곡 표현이 매우 매끄럽게 변화하였으며, 특히 날개 부분의 굴곡과 높은 수준의 highlight 및 shadow가 완화되었다. nike의 albedo에서도 highlight와 shadow가 완화된 것을 볼 수 있고, 전체적으로 밝기가 감소한 모습을 볼 수 있다. choonsik에서의 surface normals는 특히 얼굴아래 팔과 맡닿아 있는 부분의 shadow와 이목구비의 shadow가 완화된 것을 확인할 수 있다. choonsik의 albedo에서는 shadow가 이전보다 더 부드럽게 표현되어 있다. toothless의 결과는 앞의 세 물체와 비교하였을 때, 특히 albedo map의 밝기가 극명하게 저하된 것을 확인할 수 있었다. 그 이유로 toothless 물체의 표면이 원래 어두운 색이며 눈 부분만 밝은 것으로 인해, 두 부분 모두 outlier로 간주되어 표현이 감소한 것으로 보인다. toothless의 surface normals는 다른 물체들에서 확인할 수 있었듯이 매끄러워진 표면과 감소된 outlier를 확인할 수 있었다. 종합하자면, RPCA를 적용함으로써 surface normals는 매끄럽고 부드러워지며, shadow 및 highlight의 표현이 완화되고, 특히 albedo map의 밝기가 전반적으로 감소된다는 점을 확인 할 수 있었다. weak RPCA의 경우 점진적인 수렴으로 인해 LS와 육안으로 거의 차이가 없는 결과를 보여주었다. 여기서 RPCA의 수렴 hyperparameter에 따라 RPCA의 outlier 분리 정도가 결정되어 결과에 영향을 미칠 수 있음을 확인할 수 있었다. surface normals와 albedo map을 바탕으로 판단하였을 때 RPCA가 weak RPCA보다 표현이 우수했다. 다만 toothless의 albedo map은 오히려 weak RPCA가 RPCA보다 결과가 좋았다.

다음으로 이 surface normals와 albedo map을 이용해 relighting한 결과를 Figure 9에 시각화하였다. relighting 결과는 전반적으로 LS와 weak RPCA가 오히려 RPCA보다 나은 성능을 보이는 것으로 나타난다. 그 이유로, RPCA의 relighting 결과에는 outlier들이 감쇠된 그리고 전체 밝기가 감소된 모습을 보이는데, 새로운 unknown light에 대한 ground truth를 보면 highlight와 shadow가 강조되어 있는 것을 들 수 있다. 해당 outlier들을 LS와 RPCA가 조금 더 잘 반영하여 결과가 좋은 것으로 보인다. 다만 grounding truth와 차이는 있더라도, 표면 표현에 집중하여 더 부드럽고 일반적인 relighting을 원한다면, 여전히 RPCA가 더 적합하다고 볼 수 있겠다.

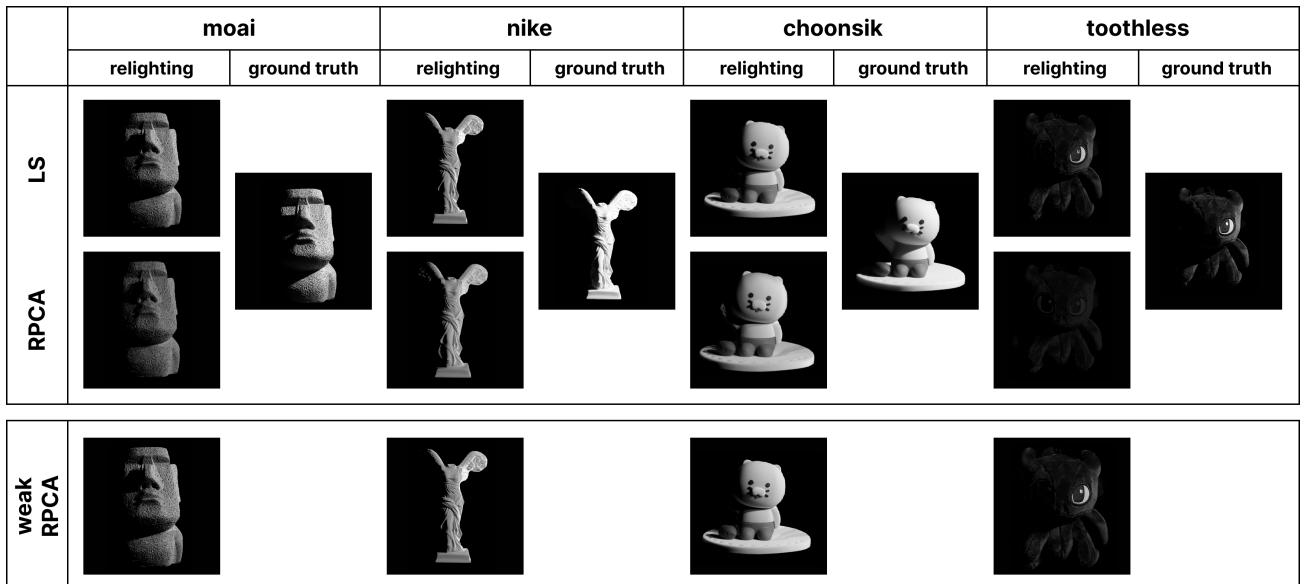


Figure 9: result of relighting (LS vs. RPCA)

아래 Figure 10은 relighting 결과를 MSE 지표를 통해 정량적으로 나타낸 표이다. 모든 물체에 대해 일관된 결과가 나타났는데, LS보다 RPCA의 MSE 오차가 높고, weak RPCA의 MSE 오차가 가장 낮게 나타났다. LS보다 RPCA의 MSE 오차가 높은 것에 대해서는 위에서 언급한 RPCA가 unknown light에 대한 relighting에서 덜 효과적임을 정성적으로 이야기한 것을 다시 근거로 들 수 있겠다. 특히 밝기의 전체적인 차이에 의해 오차가 크게 나타난 것으로 보인다. 너무 강한 RPCA는 오히려 정량적인 지표에서 오차를 키울 수 있음을 확인할 수 있다. weak RPCA의 MSE 오차가 가장 낮은 것을 통해, 적당한 수렴에 의해 적용된 RPCA으로 LS에서 조금 더 개선된 relighting을 수행할 수 있음을 확인할 수 있다.

	moai	nike	choonsik	toothless
LS	0.003431686810149389	0.005836483064140124	0.0023631540022866003	0.0011988123048224547
RPCA	0.004778866166860953	0.006075705233384553	0.002594467681702986	0.002463120139884482
weak RPCA	0.003450863586870668	0.005832907961283161	0.002360086108161282	0.0012082978985950045

Figure 10: result of relighting with MSE (LS vs. RPCA)

3.2 visualization of the Robust PCA

RPCA와 weak RPCA에 대해, lambertian term과 non-lambertian term이 어떻게 분리되는지 시각적으로 확인하고자, $D = A + E$ 의 각 행렬을 이미지로 저장하였다. 여기서 D 는 RPCA에 투입된 원본 이미지이고, A 가 lambertian term, E 가

non-lambertian term이다. Figure 11은 RPCA (strong)에 대한 시각화 결과이다. 특히 E 를 살펴보면, 모든 물체에 대해 특히 highlight에 대한 outlier들이 잘 분리된 것을 육안으로 확인할 수 있다. 주목할 점은 choonsik의 오른쪽 아래에 위치한 그림자 뒤의 비이상적인 highlight에 대해 아주 효과적으로 분리가 이루어지고 있다는 것이다. shadow에 대한 outlier는 A 에서 그림자 영역이 밝아진 것을 통해 확인할 수 있다 (E 에서는 값이 작기 때문에 어둡게 보임). 결론적으로 RPCA가 outlier들을 잘 완화하였음을 시각적으로 확인할 수 있다.

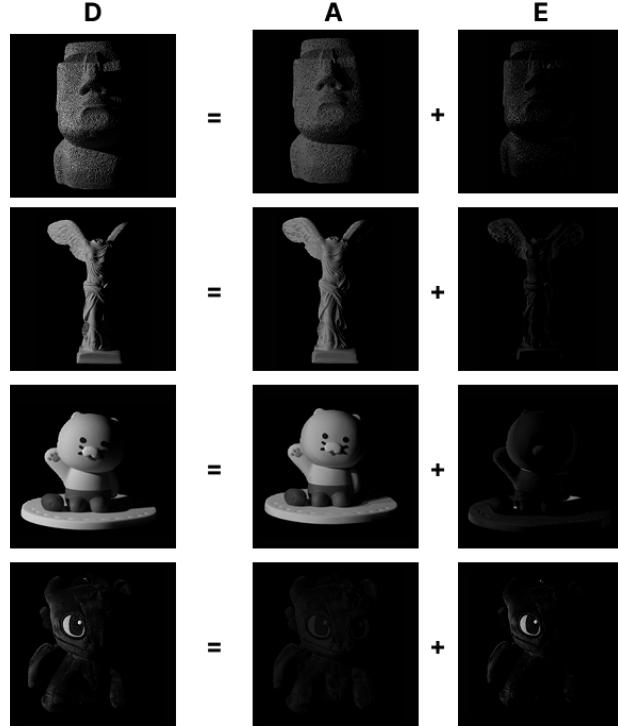


Figure 11: result of RPCA (strong)

다음 Figure 12는 weak RPCA에 대한 시각화 결과이다. weak RPCA는 RPCA(strong)에 비해 outlier를 덜 분리하고 있는 모습을 확인할 수 있다. 육안상으로는 분리되고 있는 모습을 바로 관측하기 어려워, E 의 밝기를 높여 분리된 non-lambertian term을 확인할 수 있었다. 주목할 점은 Figure 11의 RPCA(strong)에서의 E 와 거의 비슷한 영역에서 outlier 추출이 이루어지고 있다는 점이다. 즉, weak RPCA는 강도는 RPCA(strong)보다 약하지만, 분리된 outlier들은 정확하다는 것을 확인할 수 있다. 이를 통해 RPCA의 일관된 검출 능력을 확인할 수 있다.

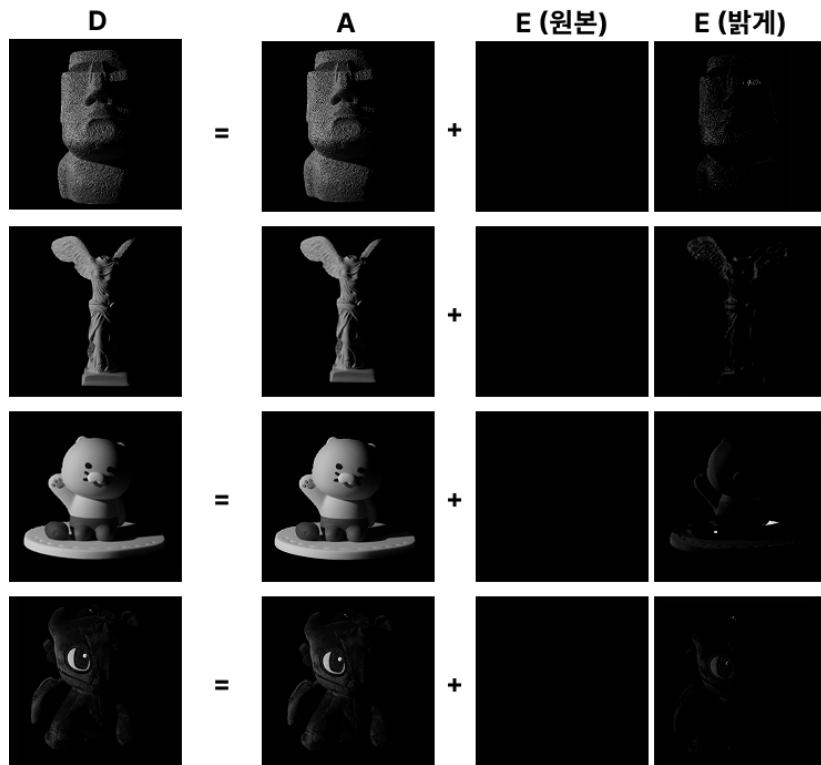


Figure 12: result of RPCA (weak RPCA)

3.3 Relighting with Photometric Factor Classification

마지막으로, PFC(Photometric Factor Classification)을 활용한 relighting 결과를 확인해보자. 위 논의에서 RPCA에 의한 relighting 결과는 outlier가 완화된 형태로 나와 오히려 오차를 증가시킬 수 있음을 확인하였다. 이를 해결하기 위해 PFC로 photometric factor를 분리하여 highlight factor는 더욱 밝게, shadow factor는 더욱 어둡게 보정하는 방식을 도입하였고, 아래 Figure 13과 14에 그 결과를 나타내었다. 이전의 결과들과 RPCA+PFC, weak RPCA+PFC를 비교해보자.

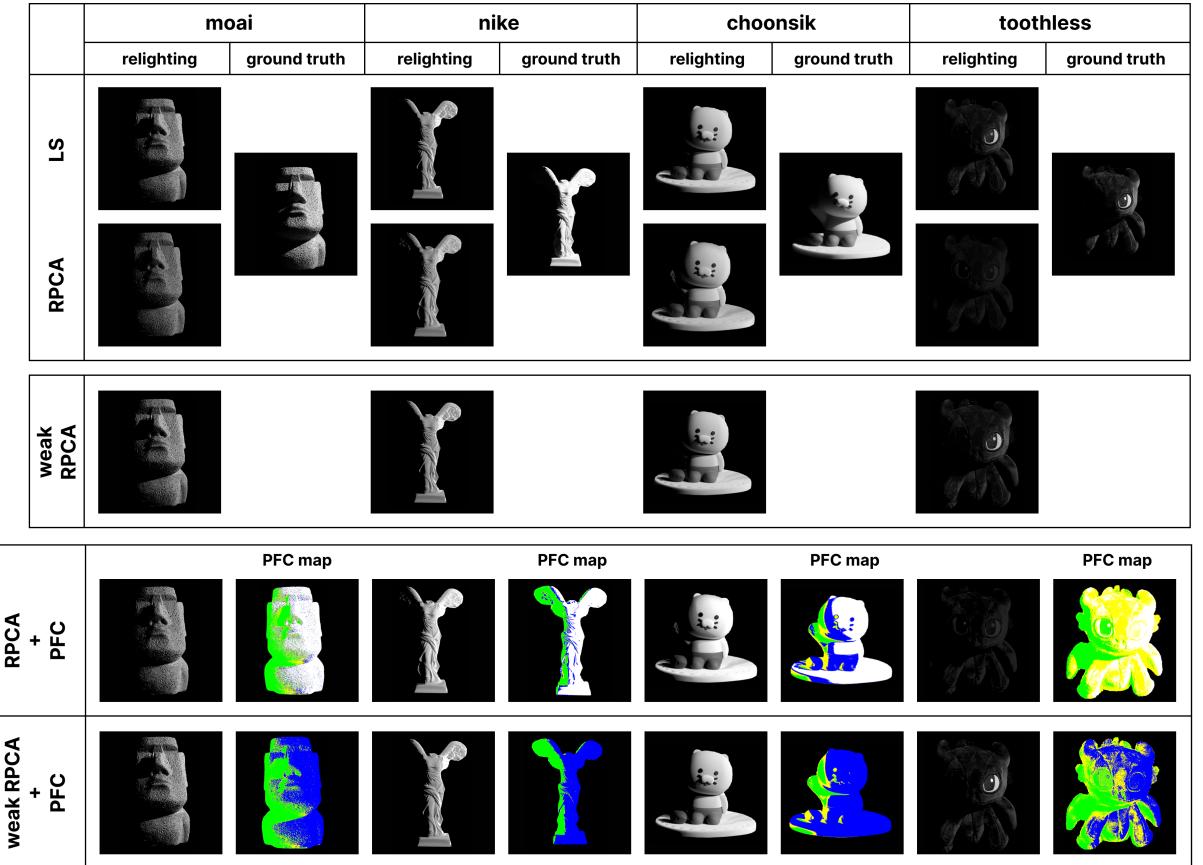


Figure 13: result of PFC relighting (compare all)

먼저, PFC를 수행한 후 결과는 아래 두 줄의 PFC map에 나타내었다. 전체적으로 4가지 영역 (S, D, A, C)가 효과적으로 잘 분리된 것을 확인할 수 있다. RPCA+PFC의 경우, 특히 S 에 해당하는 specular reflection의 비율이 많은 것으로 확인된다. 또한 toothless의 경우, 표면의 검은 부분이 C 의 cast shadow로 오인되어 분류된 것을 확인할 수 있었다. weak RPCA+PFC의 경우, RPCA+PFC에서 S 에 해당되었던 부분들이 일반 D 로 표시된 것을 확인할 수 있다. 일반적인 그림자에 해당하는 A 에 대해서는 RPCA+PFC와 weak RPCA+PFC가 비슷하게 나타났다. 이러한 PFC 분리 영역 별로 weight를 주어 relighting을 수행한 결과는 각 PFC map의 왼쪽편에 나타나 있다. 종합적으로 특히 highlight 부분의 밝기가 올라가며 ground truth의 밝은 부분과 밝기와 비슷해지는 것을 확인 할 수 있다. 이는 albedo map \circ RPCA를 거치며 밝기가 낮아지는 현상을 완화시키며 성능을 향상시킨다고 해석할 수 있다.

	moai	nike	choonsik	toothless
LS	0.003431686810149389	0.005836483064140124	0.0023631540022866003	0.0011988123048224547
RPCA	0.004778866166860953	0.006075705233384553	0.002594467681702986	0.002463120139884482
RPCA + PFC	0.0033956409046867085	0.003899350562725865	0.0025679959723933984	0.0023224902049736222
weak RPCA	0.003450863586870668	0.005832907961283161	0.002360086108161282	0.0012082978985950045
weak RPCA + PFC	0.003055892480518413	0.0049135715699916436	0.002281459625719819	0.001176014147688225

Figure 14: result of relighting with MSE (compare all)

위의 Figure 14의 표에서 RPCA+PFC와 weak RPCA+PFC의 물체별 MSE 오차를 확인할 수 있다. RPCA에 비해 특히 RPCA+PFC의 오차가 매우 효과적으로 감소한 것을 확인할 수 있다. weak RPCA에 비해서도 weak RPCA+PFC의 오차가 효과적으로 감소하였다. 가장 작은 오차를 보이는 것은 weak RPCA+PFC로, LS보다 좋은 정량적 성능을 보여준다. 결과적으로 PFC weight를 이용한 relighting이 정량적인 오차를 줄이는 데에 매우 효과적임을 나타낸다.

4 Conclusion

Photometric Stereo 문제에 대해, 우리는 RPCA와 PFC weight을 도입하여 robust한 결과를 확인할 수 있었다. RPCA를 도입함으로써 매끄럽고 부드러운 surface normals를 구할 수 있으며, shadow와 highlight와 같은 non-Lambertian outlier가 완화된 albedo map과 surface normals를 구할 수 있다. 다만, RPCA를 적용하는 과정에서 albedo map의 전체적인 밝기가 감소한다는 한계점이 있었지만, PFC weight을 활용한 relighting을 통해 이를 해결할 수 있었다. PFC로 분리된 specular highlight 영역에 적절한 비율로 밝기를 높여줌으로써 relighting 성능을 높일 수 있었다. relighting에 대한 정량적인 MSE 오차를 가장 최소화할 수 있는 모델은 weak RPCA + PFC method였다. 본 프로젝트에서는 11개의 light source를 사용하여 문제를 풀었지만, 더 많은 조명 조건이 추가된다면 normal estimation에서의 ambiguity가 완화되어 더 높은 relighting 품질을 기대할 수 있을 것이다. 뿐만 아니라, RPCA의 수렴 정도를 hyperparameter를 통해 조절하는 실험을 정밀하게 수행함으로써 더 높은 relighting 성능을 가지는 최적점을 찾을 수도 있을 것이다.

References

- [1] R. J. Woodham, "Photometric method for determining surface orientation from multiple images," in *Optical Engineering*, vol. 19, no. 1, pp. 139–144, 1980.
- [2] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?: Recovering low-rank matrices from sparse errors," in *Proceedings of the IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 1–4, 2010.
- [3] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma, "Convex Optimization Based Low-Rank Matrix Completion and Recovery for Photometric Stereo and Factor Classification," *Manuscript submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2011.
- [4] Z. Lin, M. Chen, and Y. Ma, "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices," *arXiv preprint arXiv:1009.5055*, 2010.
- [5] Z. Lin, M. Chen, and Y. Ma, "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices," *arXiv preprint arXiv:1009.5055v3*, 2013.
- [6] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma, "Robust photometric stereo via low-rank matrix completion and recovery," in *Proceedings of the 10th Asian Conference on Computer Vision (ACCV)*, vol. Part III, pp. 703–717, 2010.