Group Work - Computer Vision

Uncalibrated Photometric Stereo Constrained by Intrinsic Reflectance Image and Shape From Silhouette; A Reimplementation

Arthur Clarysse[*]
arthur.clarysse@vub.be
Vrije Universiteit Brussel
Brussels, Belgium

Jens Dumortier[†]
jens.dumortier@vub.be
Vrije Universiteit Brussel
Brussels, Belgium

Guillaume Tibergyn[‡]
guillaume.tibergyn@vub.be
Vrije Universiteit Brussel
Brussels, Belgium

1 Summary

The reimplementation in this group project was based on the paper "*Uncalibrated photometric stereo constrained by intrinsic reflectance image and shape from silhoutte*" by [Hashimoto et al. 2019]. In this paper, they try to estimate the surface normal, which in itself is not a very challenging problem, However, in our case, the direction of the light sources is unknown, making it much harder to solve. Mainly because of its ambiguous nature. Fortunately, it is possible to solve this problem by adding two constraints: the intrinsic reflection (albedo) constraint and an approximate normal constraint. This results in a reasonable estimate of the surface normal; an overview of the estimations and constraints are shown in Figure 1.

The main goal of the original paper is to implement an uncalibrated photometric stereo pipeline that estimates surface normals from multiple images without prior knowledge of light directions. This pipeline first estimates the intrinsic reflectance (or albedo) and the estimated normal, which are later used to calculate the constraints. Finally, we can calculate the guide normal and use it to make a 3d representation of the image.

As a dataset, we use the photometric stereo dataset made by Harvard [Xiong et al. [n. d.]]. The dataset contains multiple shaded images of different animals, including cats, frogs, turtles, and more.

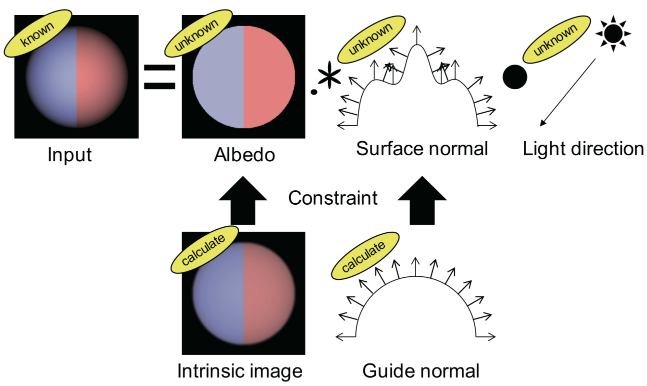


Figure 1: Overview of the known and estimated parts of the paper [Hashimoto et al. 2019]

2 Implementation

2.1 Dataset & Preprocessing

We first collected data from the dataset by [Xiong et al. [n. d.]], then selected the object we wanted to create a 3D representation of. In our case, we chose the cat image, we did not perform any preprocessing steps.

2.2 Albedo Estimation

After preprocessing, we can estimate the albedo. To do this, we can start by calculating the average image; this results in an image with reduced shading and shadows. Unfortunately, some shading effects remain; we can solve this by applying a bilateral filter.

The result from this step is the estimated albedo, and is shown in Figure 2a. This result is still unusable in standard photometric stereo techniques because of its ambiguities, but fortunately, we can use the uncalibrated photometric stereo framework to address this.

2.3 Singular Value Decomposition

The purpose of the pipeline is to find the true surface matrix and the true light matrix, **S** and **L** respectively. To find these matrices, we start by creating the shading (intrinsic illumination) matrix by dividing the image i_f by the albedo values $\hat{a}$ above the threshold T_a:

$$\hat{i}_{pf} = \frac{i_{pf}}{\hat{a}_p}, \qquad \text{for all pixels } p \text{ with } \hat{a}_p > T_a \tag{1}$$

Stacking the $\hat{P}$ valid pixels over all F images yields the shading matrix $\hat{\mathbf{I}} \in \mathbb{R}^{\hat{P}\times F}$. We then apply SVD to the shading matrix $\hat{\mathbf{I}}$:

$$\hat{\mathbf{I}} = \mathbf{U}\mathbf{W}\mathbf{V}^\top. \tag{2}$$

Keeping only the three largest singular values gives the rank-3 approximation, one rank for every dimension in the 3D space.

$$\hat{\mathbf{I}} \approx \mathbf{U}'\mathbf{W}'\mathbf{V}'^\top \tag{3}$$

From which we define the pseudo surface and pseudo light matrices as

$$\mathbf{S}' = \mathbf{U}' \in \mathbb{R}^{\hat{P}\times 3} \tag{4}$$

$$\mathbf{L}' = \mathbf{W}'\mathbf{V}'^\top \in \mathbb{R}^{3\times F} \tag{5}$$

[*]Student number: 0616411
[†]Student number: 0617692
[‡]Student number: 0618147

However, two ambiguities remain. First, we have that the SVD factorization $\hat{\mathbf{I}} = \mathbf{SL}$ is not unique. For any invertible 3×3 matrix $\mathbf{A}$, $\mathbf{SA}$ and $\mathbf{A}^{-1}\mathbf{L}$ produce the same image matrix, so $\mathbf{S}$ and $\mathbf{L}$ are only defined up to an arbitrary linear transformation. This ambiguity can be resolved by using the constant albedo constraint.

Second, we have that — even when using the albedo constraint —, an orthogonal (rotation) ambiguity $\mathbf{R}$ exists. For any orthogonal 3×3 matrix $\mathbf{L}$, $\mathbf{SR}$ and $\mathbf{R}^\top\mathbf{L}$ also produce the same image matrix, so $\mathbf{S}$ and $\mathbf{R}$ are still only defined up to a global rotation. When the second ambiguity is resolved by using the guide normal constraint, we are left with only one solution for the scaled normals.

2.4 Constant Albedo Constraint

We can calculate the true surface matrix $\mathbf{S}$ and true light matrix $\mathbf{L}$ by disambiguating the matrices using the ambiguity matrix $\mathbf{A}$.

$$\begin{aligned} \mathbf{S} &= \mathbf{S}'\mathbf{A} \\ \mathbf{L} &= \mathbf{A}^{-1}\mathbf{L}' \end{aligned} \tag{6}$$

The following steps show how to calculate $\mathbf{A}$:

1. We start by defining the matrix $\mathbf{C}$ and vector $\mathbf{b}$:

$$\mathbf{C} = \begin{bmatrix} s_{x1}'^2 & s_{y1}'^2 & s_{z1}'^2 & 2s_{x1}'s_{y1}' & 2s_{y1}'s_{z1}' & 2s_{z1}'s_{x1}' \\ s_{x2}'^2 & s_{y2}'^2 & s_{z2}'^2 & 2s_{x2}'s_{y2}' & 2s_{y2}'s_{z2}' & 2s_{z2}'s_{x2}' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{x\hat{P}}'^2 & s_{y\hat{P}}'^2 & s_{z\hat{P}}'^2 & 2s_{x\hat{P}}'s_{y\hat{P}}' & 2s_{y\hat{P}}'s_{z\hat{P}}' & 2s_{z\hat{P}}'s_{x\hat{P}}' \end{bmatrix} \tag{7}$$

$$\mathbf{b} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{bmatrix}^\top$$

Such that

$$\mathbf{Cb} = \mathbf{1} \tag{8}$$

$$\mathbf{b} = \mathbf{C}^+\mathbf{1} \tag{9}$$

With
- $\hat{P}$: The number of pixels for which the albedo value is larger than a certain threshold T_a
- $\mathbf{C}^+$: The pseudo-inverse of $\mathbf{C}$
- $\mathbf{1} = 1_{\hat{P}}$ (a column vector of $\hat{P}$ ones)

2. Now we can define the symmetric matrix $\mathbf{B}$:

$$\mathbf{B} = \mathbf{AA}^\top = \begin{bmatrix} b_1 & b_4 & b_6 \\ b_4 & b_2 & b_5 \\ b_6 & b_5 & b_3 \end{bmatrix} \tag{10}$$

With
- $b_i \in \mathbf{b}$

Performing singular value decomposition on $\mathbf{B}$ results in:

$$\mathbf{B} = \mathbf{U}_B\mathbf{W}_B\mathbf{U}_B^\top \tag{11}$$

3. As a result, we can calculate the ambiguity matrix as follows:

$$\mathbf{A} = \mathbf{U}_B\mathbf{W}_B^{\frac{1}{2}} \tag{12}$$

4. Finally, we can update the pseudo light and surface matrices as follows:

$$\begin{aligned} \mathbf{S}'' &= \mathbf{S}'\mathbf{A} \\ \mathbf{L}'' &= \mathbf{A}^{-1}\mathbf{L}' \end{aligned} \tag{13}$$

Note that $\mathbf{A}$ is only determined up to an orthogonal factor, leaving a residual rotation ambiguity that is resolved by the guide normal constraint.

2.5 Guide Normal Constraint

The original paper by Hashimoto et al. [2019] does not explicitly explain how the guide normal is calculated. However, to complete the pipeline, we needed to add this step.

To remove the final ambiguity, we can use the orthogonal matrix $\mathbf{R}$. We can formulate the true surface and light matrices in terms of $\mathbf{R}$ and the pseudo surface matrix $\mathbf{S}''$ and pseudo light matrix $\mathbf{L}''$ that we obtained in the previous section:

$$\begin{aligned} \mathbf{S} &= \mathbf{S}''\mathbf{R} \\ \mathbf{L} &= \mathbf{R}^\top\mathbf{L}'' \end{aligned} \tag{14}$$

1. We begin by computing the silhouette image $\mathbf{H}$ (Figure 2b) by detecting the object's boundary pixels in the image. This silhouette is a binary mask in which values above the albedo threshold T_a are set to 1, and those below are set to 0.

$$\mathbf{H}(p) = \begin{cases} 1, & \hat{a}_p > T_a \\ 0, & otherwise \end{cases} \tag{15}$$

With
- $\mathbf{H}$: The silhouette image with pixels p
- $\hat{a}_p$: The albedo estimate of the pixel p
- T_a: The threshold

2. Then we can compute the approximate shape $\tilde{\mathbf{H}}$ (Figure 2c) from the silhouette image; this shape is a height map where the center of the image is high (e.g., 1) and the edges are low (e.g., 0). This step can be performed using the distance transform of $\mathbf{H}$.

3. The approximate shape is then normalized to prevent extreme gradients; the range 0 to $\frac{\text{image width}}{2}$ is used for this step.

4. The next step involves taking the gradient of the approximate shape $\tilde{\mathbf{H}}$:

$$\left(\frac{\partial\tilde{\mathbf{H}}}{\partial y}, \frac{\partial\tilde{\mathbf{H}}}{\partial x}\right) = \nabla\tilde{\mathbf{H}}. \tag{16}$$

5. Now, we can calculate the guide normal $\tilde{\mathbf{S}}$ (Figure 2d) from the estimated height map $\tilde{\mathbf{H}}$ by stacking the components (Equation 29). We then normalize each normal vector to unit length by dividing by its ℓ_2-norm (Equation 30) and skipping divisions by zero (Equation 31).

6. With the guide normal $\tilde{\mathbf{S}}$ calculated, we can calculate the ambiguity orthogonal matrix $\mathbf{R}$:

$$\mathbf{R} = \mathbf{S}''^{+}\tilde{\mathbf{S}} \tag{17}$$

Using the following relation:

$$\tilde{\mathbf{S}} = \mathbf{S}''\mathbf{R} \tag{18}$$

7. We can orthogonalize the matrix $\mathbf{R}$ into $\tilde{\mathbf{R}}$ as follows:

$$\mathbf{R} = \mathbf{Q}_R\,\mathbf{R}_R \tag{19}$$

$$\tilde{\mathbf{R}} = \mathbf{Q}_R \tag{20}$$

8. After orthogonalizing $\mathbf{R}$ into $\tilde{\mathbf{R}}$, the final true surface and light matrices — $\mathbf{S}$ and $\mathbf{L}$ respectively — can be calculated:

$$\mathbf{S} = \mathbf{S}''\tilde{\mathbf{R}} \tag{21}$$

$$\mathbf{L} = \tilde{\mathbf{R}}^{\top}\mathbf{L}'' \tag{22}$$

2.6 Constructing the Image Matrix

After following the previous steps correctly, we obtained the true light matrix $\mathbf{L}$ and the true surface matrix $\hat{\mathbf{S}}$. However, while $\mathbf{L}$ contains all light direction for all pixels P, $\hat{\mathbf{S}}$ only contains the non-background pixels $\hat{P}$. We will reconstruct the complete true surface matrix $\mathbf{S}$ for all pixels P using equation 25.

$$\hat{\mathbf{S}} = \begin{bmatrix} s_{1x} & s_{1y} & s_{1z} \\ s_{2x} & s_{2y} & s_{2z} \\ \vdots & \vdots & \vdots \\ s_{\hat{P}x} & s_{\hat{P}y} & s_{\hat{P}z} \end{bmatrix} \tag{23}$$

$$\mathbf{L} = \begin{bmatrix} l_{x1} & l_{x2} & \cdots & l_{xF} \\ l_{y1} & l_{y2} & \cdots & l_{yF} \\ l_{z1} & l_{z2} & \cdots & l_{zF} \end{bmatrix} \tag{24}$$

With

- P: The number of pixels
- $\hat{P}$: The number of foreground pixels
- F: The number of images

Using the image matrix $\mathbf{I}$:

$$\mathbf{I} = \mathbf{S}\mathbf{L} = \begin{bmatrix} i_{11} & i_{12} & \cdots & i_{1F} \\ i_{21} & i_{22} & \cdots & i_{2F} \\ \vdots & \vdots & \ddots & \vdots \\ i_{P1} & i_{P2} & \cdots & i_{PF} \end{bmatrix} \tag{25}$$

We can calculate $\mathbf{S}$ as follows:

$$\mathbf{S} = \mathbf{I}\,\mathbf{L}^{+} \tag{26}$$

With $\mathbf{L}^{+}$ The pseudo-inverse of $\mathbf{L}$ and

$$\mathbf{S} = \begin{bmatrix} s_{1x} & s_{1y} & s_{1z} \\ s_{2x} & s_{2y} & s_{2z} \\ \vdots & \vdots & \vdots \\ s_{Px} & s_{Py} & s_{Pz} \end{bmatrix} \tag{27}$$

The resulting surface matrix $\mathbf{S}$ is shown in Figure 2f. A complete schematic overview is presented in Figure 7. We provide a flowchart rather than pseudocode, as we believe a high-level overview with equations is more useful here than restating the steps above in algorithmic form.

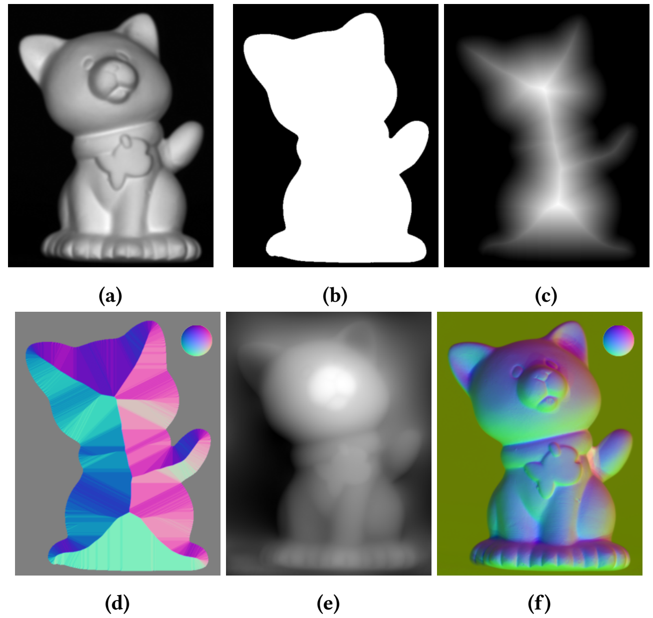


Figure 2: Intermediate results of our uncalibrated photometric stereo pipeline: (a) estimated albedo, (b) estimated silhouette, (c) estimated approximate shape, (d) estimated guide normal, (e) estimated depth map, and (f) estimated normal map. With a sphere shown in (d) and (f) as a legend, indicating how the colours map to surface directions.

3 Results

3.1 Qualitative Results

The pipeline output are the surface matrix $\mathbf{S}$ and light matrix $\mathbf{L}$, the first of which can be easily visualized; the results are shown in Figure 3a. From this result, a height map can be generated using the Frankot-Chellappa algorithm; this result is shown in Figure 3b.

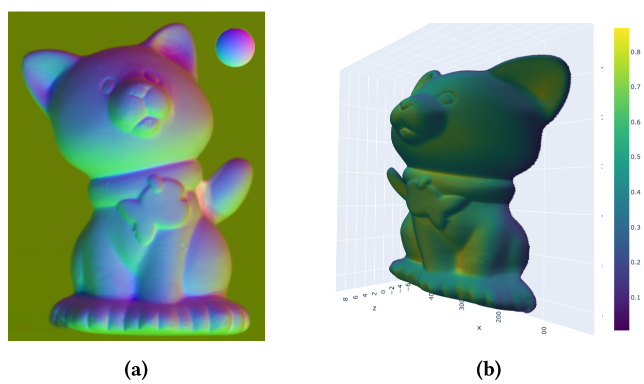
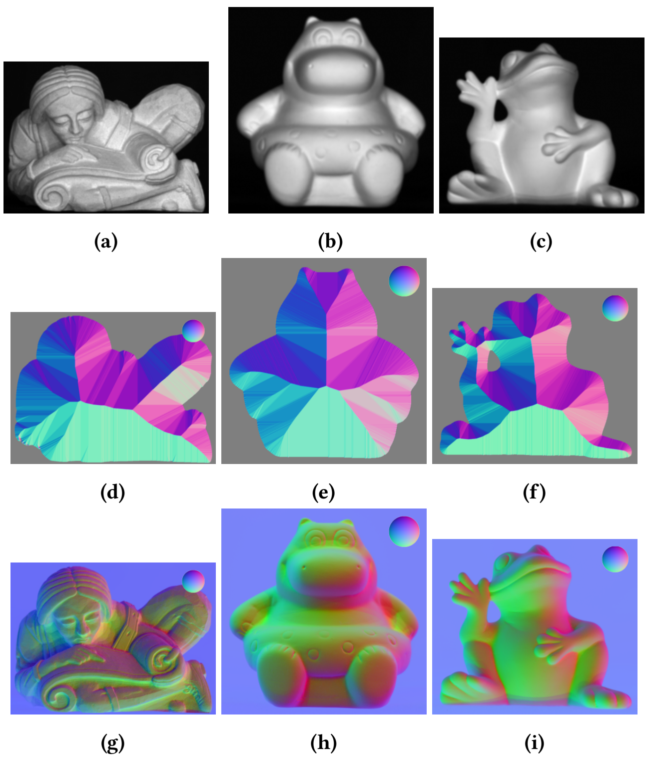


Figure 3: The result of the pipeline, showing (a) the estimated normal map and (b) a 3D reconstruction obtained from the estimated normal map using the Frankot-Chellappa algorithm. With a sphere shown in (a) as a legend, indicating how the colours map to surface directions.

We also ran other objects from the dataset through the uncalibrated photometric stereo pipeline; the results are shown in Figure 4. The cat (Figure 3b), the scholar (Figure 4g), and the frog (Figure 4i) show very good results, the 3D shape of the object closely represents the expected true shape. However, the result of the rhinoceros is less good, we can see that the resulted shape is wrong. Especially when looking at the bottom of the nose, where we see it pointing upwards instead of downwards. An explanation for this wrong shape can be that the rhino object has less corners and larger, smoother surfaces. Making it more difficult for the algorithm to correctly estimate the depth of the object.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 4: Results for *scholar, rhino,* and *frog* as inputs to the pipeline, showing (a-c) the albedo estimates, (d-f) the guide-normal estimates, and (g-i) the normal maps. With a sphere shown in (d-i) as a legend, indicating how the colours map to surface directions.

3.2 Quantitative Results

Visually inspecting our results is not enough to properly test our model. Therefore, we calculated an error measure that gives us more insight into how well our pipeline predicts the surface normal. Our error measure quantifies the difference between the true surface map **T** and the estimated surface map **S** as the ℓ_2-norm distance. Note that the true surface map we use is the result of the calibrated estimation of the surface map. The dataset creators computed their scores by re-rendering the estimated normal map into a shading image and comparing it with the captured image. Unfortunately, we cannot replicate this evaluation method because the calibrated lighting parameters needed to render the shading images under the original illumination are not available; therefore, we instead compare our estimated surface map to the calibrated photometric-stereo result used as ground truth. Lower values indicate a closer match (Equation 28).

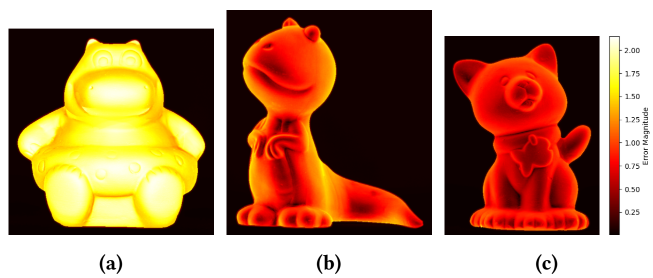
We used this error to test different objects from the dataset; the results are shown in Table 1. There, we can observe that the errors of the frog and hippo are much higher than those of the cat, lizard, and pig. These results reflect the same conclusion as when looking at the 3D representation of the objects.

Figure 5 is a visualized version of the error scores, where we can draw the same conclusion. The pipeline has much more difficulty predicting the correct shape of the hippo than it does for the other objects. We can also see that the model has more difficulty predicting flat surfaces; we assume this is because flat surfaces lack shadows that the pipeline is based on. The shadow-rich areas exhibit lower errors, further supporting our assumption.

$$\mathbf{D} = \|\mathbf{S} - \mathbf{T}\|_2 \tag{28}$$

With

- **S**: The estimated surface map
- **T**: The true surface map
- **D**: The distance between the true surface map and the estimated surface map



(a) (b) (c)

Figure 5: Magnitude of the error of the normal map visualized for (a) scholar, (b) lizard, and (c) cat

4 Discussion

4.1 Guide Normal Constraint

A crucial step in calculating the true surface and light matrices is applying the guide normal constraint explained in Section 2.5. One would thus expect the original authors to explain this constraint in great detail. However, they state the following: "*The algorithm to calculate guide normal used in this paper is quite different from other existing methods, however, we skip to explain it since it is a combination of existing techniques developed in the field of image processing*".

Therefore, we implemented this step using other sources. Unfortunately, since the computation is essentially a straightforward combination of standard image processing operations, few papers or books provide a complete, end-to-end algorithm for constructing the guide normal. We therefore combined modern tools like generative AI with additional references to obtain a practical, well-functioning guide-normal procedure. This resulted in a well-commented working function described in Algorithm 1.

A slight difference we observed compared to the paper is that the resulting guide normals are less smooth than the original ones. Our results still showed excellent results, so we did not see this as an issue.

4.2 Threshold T_a

The authors of the original paper talk about using a threshold T_a when calculating the shading image. Unfortunately, they do not explain how they arrived at this threshold. Therefore, we plotted the frequency of the albedo values $\hat{a}_p$ using 256 bins and visually determined a reasonable threshold. We observed that most object values lie above 0.2, while background values peak below 0.1, suggesting 0.1 as a natural threshold. The plot we used is shown in Figure 6.

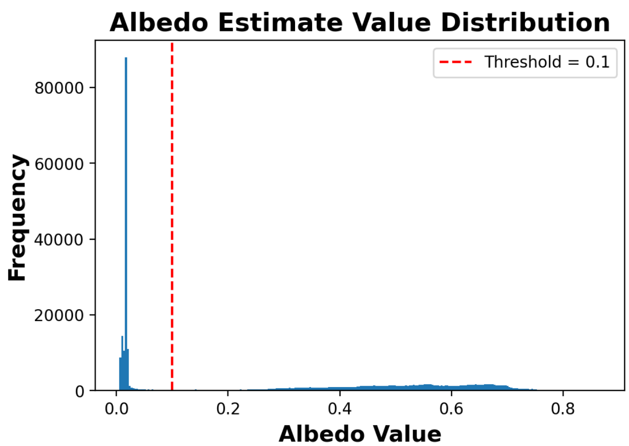


Figure 6: Binned albedo values using 256 bins, with the chosen threshold marked by the red vertical line. Clearly, the initial peak corresponds to low-albedo background pixels, while the distribution at higher values captures the object region.

4.3 General

The reproduced pipeline from the original paper generally works well. When looking at the error values in Table 1, we see that most objects are predicted with low error scores. These results are further strengthened by examining the 3D and error plots.

5 Reproducibility Checklist

- ☒ Code provided & documented
- ☒ Data included
- ☒ Environment included
- ☒ Run instruction included
- ☒ Random seeds (No seeds were used)
- ☒ Expected outputs
- ☒ Algorithm pseudo code included

References

Shuhei Hashimoto, Daisuke Miyazaki, and Shinsaku Hiura. 2019. Uncalibrated photometric stereo constrained by intrinsic reflectance image and shape from silhoutte. In *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE, Tokyo, Japan, 1–6. doi:10.23919/MVA.2019.8758025

Ying Xiong, Ayan Chakrabarti, Ronen Basri, Steven J. Gortler, David Jacobs, and Todd Zicker. [n. d.]. From Shading to Local Shape. https://vision.seas.harvard.edu/qsfs/index.html

A Guide Normal Calculations

- The equations for calculating the guide normal:

$$\mathbf{N}(x,y) = \begin{pmatrix} -\dfrac{\partial \tilde{\mathbf{H}}}{\partial x}(x,y) \\ -\dfrac{\partial \tilde{\mathbf{H}}}{\partial y}(x,y) \\ 1 \end{pmatrix}, \tag{29}$$

$$\eta(x,y) = \|\mathbf{N}(x,y)\|_2, \tag{30}$$

$$\tilde{\mathbf{S}}(x,y) = \begin{cases} \dfrac{\mathbf{N}(x,y)}{\eta(x,y)}, & \eta(x,y) \neq 0, \\ \mathbf{0}, & \eta(x,y) = 0. \end{cases} \tag{31}$$

B Error Values

	Cat	Frog	Hippo	Lizard	Pig
Mean Error	0.4	0.8	0.84	0.29	0.38
Max Error	2.15	2.03	1.83	1.66	1.87

Table 1: Placeholder results for two animals.

C Pseudo Code for Guide Normal Constraint

Algorithm 1 Compute guide normal from albedo estimate

Require: Albedo estimate $\hat{\rho}$, threshold T_a
Ensure: Guide normal map $\mathbf{N}_g$ (and optional intermediate outputs)

Silhouette from albedo
$\mathbf{H} \leftarrow \mathbb{1}[\hat{\rho} > T_a]$ ▷ binary mask of the object

Approximate shape from silhouette
$\mathbf{Z} \leftarrow \textsc{DistanceTransform}(\mathbf{H})$ ▷ height-like map (higher near center)
$\mathbf{Z} \leftarrow \textsc{Normalize}(\mathbf{Z})$ ▷ scale to a bounded range

Convert shape to normals
$(\partial_y \mathbf{Z}, \partial_x \mathbf{Z}) \leftarrow \textsc{Gradient}(\mathbf{Z})$
$\mathbf{N}_g \leftarrow \textsc{Stack}(-\partial_x \mathbf{Z}, -\partial_y \mathbf{Z}, \mathbf{1})$
$\mathbf{N}_g \leftarrow \textsc{UnitNormalize}(\mathbf{N}_g)$

Mask background
$\mathbf{N}_g[p] \leftarrow \mathbf{0}$ for all pixels p with $\hat{\rho}[p] \le T_a$

return $\mathbf{N}_g$ ▷ optionally also return $\mathbf{H}$ and $\mathbf{Z}$

D Schematic Overview

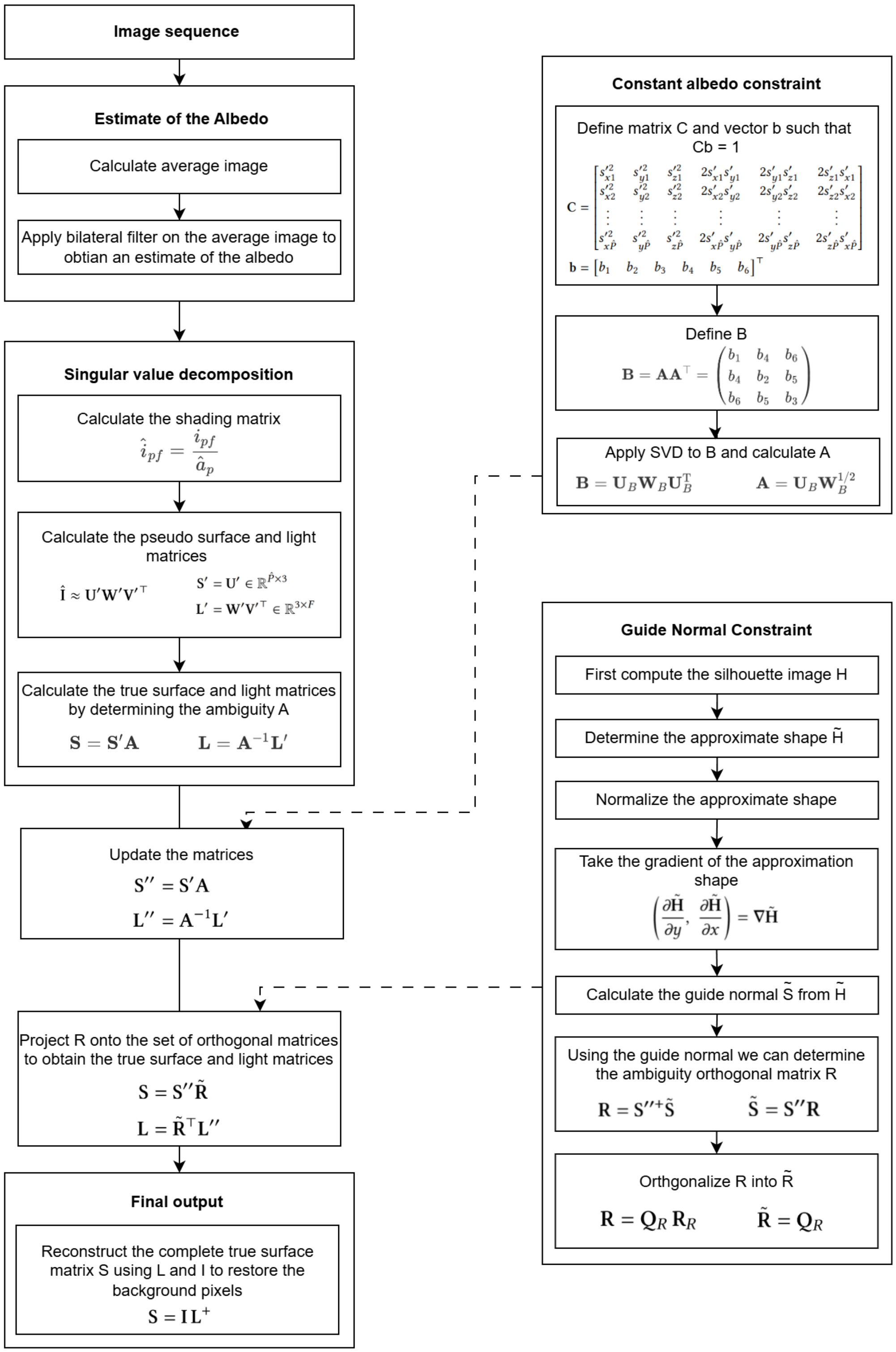


Figure 7: A schematic overview of the pipeline