

Polarimetric Dense Monocular SLAM Supplementary Material

The supplementary document explains 1) the polarization camera; 2) the polarimetric phase-angle estimation algorithm; and 3) the photometric error term in the energy minimization. All of these are well-established techniques and are presented to make our submission self-contained.

1. Polarization Camera

A polarization camera has an array of linear polarizers in front of its CMOS sensor, where neighboring cells have different polarization angles. Figure 1 (b) illustrates such an array. In this way, the four neighboring pixels will capture images under different polarization angles. Assuming spatial smoothness of the incoming signal, we can obtain four images of different polarization from a single shoot in the same spirit of demosaicing of the RGB Bayer pattern [5].

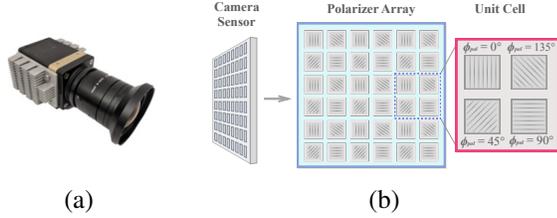


Figure 1: (a) A Polarization camera with 8mm lens. (b) An array of linear polarizers in front of its CMOS sensor, capturing multiple polarized images in a single shot.

2. Phase Angle Estimation

The scene radiance of an incident unpolarized light through a linear polarizer at a polarization angle ϕ_{pol} is expressed as follows [1]:

$$I(\phi_{pol}) = \frac{I_{max} + I_{min}}{2} + \frac{I_{max} - I_{min}}{2} \cos(2(\phi_{pol} - \phi)). \quad (1)$$

The formula indicates that the radiance follows a cosine curve within the range $[I_{min}, I_{max}]$. We have four measurements per pixel over the four different polarization angles ($\phi_{pol} = \phi_0, \phi_{45}, \phi_{90},$ or ϕ_{135}). These angles are given by the calibration process and are constants. We use simple linear and trigonometric algebra to solve ϕ . Specifically, we

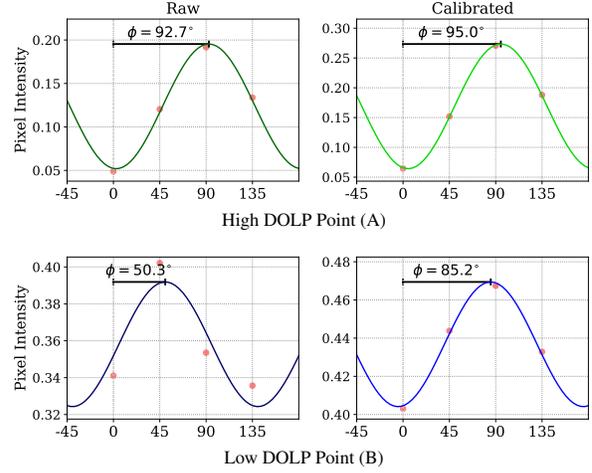


Figure 2: Computing the phase angle ϕ at two pixels (marked with Green (High DOLP) and Blue (Low DOLP) in Figure 3 (d)). Left and right are the cosine curve fitted to the observations (red dots) before and after the flat-field calibration.

rewrite the Equation (1) as:

$$\begin{aligned} I(\phi_{pol}) &= \frac{I_{max} + I_{min}}{2} + \frac{I_{max} - I_{min}}{2} \cos(2\phi_{pol} - 2\phi) \\ &= \frac{I_{max} + I_{min}}{2} + \frac{I_{max} - I_{min}}{2} \cos 2\phi \cos 2\phi_{pol} \\ &\quad + \frac{I_{max} - I_{min}}{2} \sin 2\phi \sin 2\phi_{pol}. \end{aligned} \quad (2)$$

Let $\alpha = \frac{I_{max} + I_{min}}{2}$, $\beta = \frac{I_{max} - I_{min}}{2} \cos 2\phi$, $\gamma = \frac{I_{max} - I_{min}}{2} \sin 2\phi$, the Equation (2) becomes:

$$I(\phi_{pol}) = \alpha + \beta \cos 2\phi_{pol} + \gamma \sin 2\phi_{pol}. \quad (3)$$

With the four measured intensities $I(\phi_0)$, $I(\phi_{45})$, $I(\phi_{90})$, and $I(\phi_{135})$, we build an overdetermined linear system:

$$\begin{bmatrix} 1 & \cos 2\phi_0 & \sin 2\phi_0 \\ 1 & \cos 2\phi_{45} & \sin 2\phi_{45} \\ 1 & \cos 2\phi_{90} & \sin 2\phi_{90} \\ 1 & \cos 2\phi_{135} & \sin 2\phi_{135} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} I(\phi_0) \\ I(\phi_{45}) \\ I(\phi_{90}) \\ I(\phi_{135}) \end{bmatrix}. \quad (4)$$

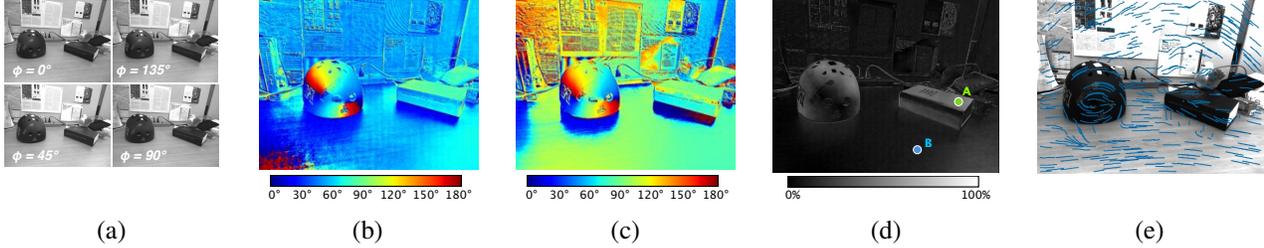


Figure 3: (a) Four images with different polarization angles. (b) Estimated azimuth angle map **without Flat-Field calibration** (Right bar shows the phase angle) (c) With Flat-Field Corrected calibration. In (c) there is obvious $/2$ -ambiguity, e.g. shadow area bellow the book and the helmet are diffuse dominated while other area in table reflect the lighting and is dominated by specular. (d) The DOLP map with two sampled pixel, the blue pixel sampled from low DOLP area while the green sampled at High DOLP area. (e) The iso-depth contour computed with the corrected azimuth angle map (c).

We find the least squares solution $[\alpha \ \beta \ \gamma]^T$, and compute the phase angle ϕ as

$$\phi = \begin{cases} 0 & \text{if } \beta = 0 \\ \frac{1}{2} \tan^{-1}\left(\frac{\gamma}{\beta}\right) & \text{otherwise} \end{cases} \quad (5)$$

Figure 2 shows the phase-angle estimation process for two pixels marked as green and blue in Figure 3 (d). The blue pixel has a low degree of linear polarization (DOLP), where the cosine curve fitting is noisy and produces a large error without the flat-field calibration. For the red pixel with high DOLP, this flat-field calibration is less critical, though improvement is also evident. Note that DOLP (ρ_{dolp}) measures the relative strength of the linear polarized light [4] and is given by:

$$\rho_{dolp} = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}. \quad (6)$$

Figure 3 (b) and (c) show the effects of the Flat-Field calibration over an entire image (Section 5). The calibration has dramatic effects on low DOLP areas (e.g., the desktop), while the angle estimation is accurate even without the calibration on high DOLP areas (e.g. the helmet).

3. Photometric term

The photometric term in the Equation (2) of the main paper is a standard photo-consistency function based on image intensities and gradients [2], defined between a source keyframe \mathbf{I} and a reference keyframe \mathbf{I}' . The photometric error for a pixel p with distance from the origin d_p and surface normal \mathbf{n}_p is,

$$E_{photo}(d_p, \mathbf{n}_p) = \sum_{q \in \mathcal{N}(p)} w(p, q) \rho(q, H(d_p, \mathbf{n}_p)q) \quad (7)$$

Here, $\mathcal{N}(p)$ denotes a local patch centered on pixel p (13×13 in our implementation). The weight w measures the color similarity at two pixels p and q as $w(p, q) = e^{-\frac{\|I_p - I_q\|}{\gamma}}$,

where I_p, I_q are the pixel intensity values at p, q respectively. The matching cost ρ measures the similarity of two pixels in image \mathbf{I} and \mathbf{I}' respectively. More specifically, given two corresponding pixels x and y , the cost is the color and gradient differences combined by a weight α .

$$\rho(x, y) = (1 - \alpha) \cdot \min(\|I_x - I'_y\|, \gamma_{col}) + \alpha \cdot \min(\|\nabla I_x - \nabla I'_y\|, \gamma_{grad}), \quad (8)$$

where the parameters γ_{col} and γ_{grad} are truncate thresholds for better robustness. We fix $\alpha = 0.9$, $\gamma_{col} = 10.0$ and $\gamma_{grad} = 2.0$ in our implementation. The correspondence between the two images are computed by a plane induced Homography [3, Page 327] defined as,

$$H(d_p, \mathbf{n}_p) = \mathbf{K}(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}_p^T}{d_p})\mathbf{K}^{-1}, \quad (9)$$

where \mathbf{R} and \mathbf{t} are the relative rotation and translation between two frames, and \mathbf{K} is the camera intrinsic matrix.

4. System computation complexity

Table 1 lists the complexity and computation time for each step in Algorithm 1 for the example *statue*. Please note we exclude the time on DSO and InfiniTAM (depth map fusion).

Here, we analyze the complexity of each step of the Algorithm 1. (Step 1) The PatchMatch Stereo for rough surface estimation has time complexity of $O(nmp^2)$ at each iteration, where m is the number of pixels per video frame, n is number of neighbors, and p is the patch size. (Step 2) The disambiguation has time complexity of $O(lm)$, where we trace a contour of l pixels for each of the m pixels for disambiguation. (Step 3) The Depth Consistency Check has time complexity of $O(m)$, where we project all the pixels to another frame to check the consistency. (Step 4) We assign the value of \mathbf{z} to \mathbf{a} , which has time complexity of $O(m)$. The time spend on this step is negligible. (Step 5 and 6) We

estimate a depth distribution along the iso-depth contour at each pixel, which has time complexity of $O(lm)$. (Step 7) We compute the KL divergence of the two depth distributions on each pixel, which has time complexity $O(m)$. (Step 8) The PatchMatch Stereo finds \mathbf{z} that minimizes the data term, which has time complexity of $O(nmp^2)$. (Step 9) The RoF [4] method finds \mathbf{a} that minimizes the smooth term, which has time complexity of $O(m)$.

Steps	Complexity	Time (ms)
Step 1: PatchMatch (4 iterations)	$O(nmp^2)$	275.29
Step 2: Disambiguation	$O(lm)$	116.23
Step 3: Depth Consistency Check	$O(m)$	0.67
<i>init. total</i>		393.82
Step 5 & 6: Trace Depth on Two Views (1 iter)	$O(lm)$	67.83
Step 7: Inlier Validation (1 iter)	$O(m)$	1.28
Step 8: Optimize Data Term (1 iter)	$O(nmp^2)$	63.63
Step 9: Optimize Smooth Term (1 iter)	$O(m)$	6.60
<i>iter. total (6 iterations)</i>		828.39

Table 1: Complexity and Computation Time of each step for the example *statue*. See the text for more details.

Our entire system is highly parallized. The DSO is running on CPU in realtime to provide a keyframe every 1-2 second. Our depth computation (Algorithm 1) runs on a Titan X GPU (with 3584 CUDA cores) for depth computation of the keyframes. It takes about 1.2 second to process one keyframe. Finally, the InfiniTAM runs on another Titan X GPU to fuse the individual depth maps to a mesh model in realtime.

References

- [1] Z. Cui, J. Gu, B. Shi, P. Tan, and J. Kautz. Polarimetric multi-view stereo. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [2] S. Galliani, K. Lasinger, and K. Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proc. of Internatoinal Conference on Computer Vision (ICCV)*, pages 873–881, 2015.
- [3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [4] A. Kadambi, V. Taamazyan, B. Shi, and R. Raskar. Polarized 3d: High-quality depth sensing with polarization cues. In *Proc. of Internatoinal Conference on Computer Vision (ICCV)*, 2015.
- [5] H. S. Malvar, L.-w. He, and R. Cutler. High-quality linear interpolation for demosaicing of bayer-patterned color images. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3.