Uncalibrated Structure from Motion on a Sphere

Supplementary Material

7. Extended derivation of spherical essential matrix form

If we expand out the form of E_{out}, we obtain the following:

$$\mathsf{E}_{\text{out}} = [\mathbf{r}_3 - \mathbf{z}]_\times \mathsf{R} \tag{13}$$

$$= \begin{bmatrix} 0 & -R_{33}+1 & R_{32} \\ R_{33}-1 & 0 & -R_{31} \\ -R_{32} & R_{31} & 0 \end{bmatrix} \mathsf{R} \tag{14}$$

$$= \begin{bmatrix} R_{23}R_{31} - R_{21}(R_{33}-1) & R_{23}R_{32} - R_{22}(R_{33}-1) & R_{23} \\ R_{11}(R_{33}-1) - R_{13}R_{31} & R_{12}(R_{33}-1) - R_{13}R_{32} & -R_{13} \\ R_{13}R_{21} - R_{11}R_{23} & R_{13}R_{22} - R_{12}R_{23} & 0 \end{bmatrix} \tag{15}$$

$$= \begin{bmatrix} R_{21}+R_{12} & R_{22}-R_{11} & R_{23} \\ R_{22}-R_{11} & -R_{21}-R_{12} & -R_{13} \\ R_{32} & -R_{31} & 0 \end{bmatrix}. \tag{16}$$

The simplifications in the last step come from the orthonormal property of rotation matrices, which gives the following equalities:

$$R_1 = R_2 \times R_3 \tag{17}$$

$$R_2 = R_3 \times R_1 \tag{18}$$

$$R_3 = R_1 \times R_2 \tag{19}$$

where R_1, R_2, R_3 denote the rows of R.

It follows that the essential matrix for spherical motion is fully described by six parameters $e_1, \ldots, e_6$ [37]:

$$\mathsf{E} = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_2 & -e_1 & e_4 \\ e_5 & e_6 & 0 \end{bmatrix} \tag{20}$$

where

$$e_1 = R_{21} + R_{12} \tag{21}$$

$$e_2 = R_{22} - R_{11} \tag{22}$$

$$e_3 = R_{23} \tag{23}$$

$$e_4 = -R_{13} \tag{24}$$

$$e_5 = R_{32} \tag{25}$$

$$e_6 = -R_{31}. \tag{26}$$

8. Proofs of Propositions 1 and 2

8.1. Proof of Proposition 1

Consider the essential matrix $\mathsf{E}(\mathbf{r}_{xy}, \theta_{xy}, \theta_z)$ of a pair of cameras with unit focal length and relative rotation $R(r_{xy}, \theta_{xy}, \theta_z)$. Proposition 1 states for any focal length f there is a pair of cameras with focal length f and relative rotation $R(r_{xy}, \theta'_{xy}, \theta_z)$ that has a fundamental matrix equivalent to E, where

$$\begin{aligned} \theta'_{xy}(f, \theta_{xy}) = \text{atan2}(2f\sin(\theta_{xy}), \\ (1+f^2)\cos(\theta_{xy}) + (1-f^2)). \end{aligned} \tag{27}$$

.

Let $c'_{xy} = \cos(\theta'_{xy})$ and $s'_{xy} = \sin(\theta'_{xy})$. From (27) we have

$$s'_{xy} = \frac{2fs_{xy}}{(1-f^2)c_{xy} + (1+f^2)}, \tag{28}$$

$$c'_{xy} = \frac{(1+f^2)c_{xy} + (1-f^2)}{(1-f^2)c_{xy} + (1+f^2)}. \tag{29}$$

For convenience we will re-write the expression for E here:

$$\mathsf{E}(\mathbf{r}_{xy}, \theta_{xy}, \theta_z) = \begin{bmatrix} (c_{xy}-1)\mathsf{S}(r_x, r_y, \theta_z) & -s_{xy}r_x \\ s_{xy}(c_z r_x + r_y s_z) \quad s_{xy}(c_z r_y - r_x s_z) & -s_{xy}r_y \\ & 0 \end{bmatrix}. \tag{30}$$

From (8) we have

$$\mathsf{F}(f, \mathbf{r}_{xy}, \theta'_{xy}, \theta_z) \sim \begin{bmatrix} \frac{(c'_{xy}-1)}{f}\mathsf{S}(r_x, r_y, \theta_z) & -s'_{xy}r_x \\ s'_{xy}(c_z r_x + r_y s_z) \quad s'_{xy}(c_z r_y - r_x s_z) & -s'_{xy}r_y \\ & 0 \end{bmatrix}. \tag{31}$$

Case 1: $\theta_{xy} \neq \pi$. Dividing E by s_{xy} and F by s'_{xy}, we have reduced the problem to showing that

$$\frac{c'_{xy} - 1}{f s'_{xy}} = \frac{c_{xy} - 1}{s_{xy}} \tag{32}$$

which is easily demonstrated.

Case 2: $\theta_{xy} = \pi$. We see that $s_{xy} = s'_{xy} = 0$ and $c_{xy} = c'_{xy} = -1$. Clearly $\mathsf{F} = \mathsf{E}/f$ so $\mathsf{F} \sim \mathsf{E}$.

This proves Proposition 1.

8.2. Proof of Proposition 2

Now consider the fundamental matrix $\mathsf{F}(f, \mathbf{r}_{xy}, \theta'_{xy}, \theta_z)$ of a pair of cameras with focal length f and relative rotation $\mathsf{R}(\mathbf{r}_{xy}, \theta'_{xy}, \theta_z)$. A pair of cameras with unit focal length and relative rotation $\mathsf{R}(\mathbf{r}_{xy}, \theta_{xy}, \theta_z)$ has an essential matrix equivalent to F, where

$$\begin{aligned} \theta_{xy} = \text{atan2}(2f\sin(\theta'_{xy}), \\ (1+f^2)\cos(\theta')_{xy} + (f^2-1)) \end{aligned} \tag{33}$$

This can be shown in a similar manner as the proof of Proposition 1.

9. Rotation decomposition

Algorithm 1 shows how we decompose a rotation R into a R_{xy} and R_z.

Algorithm 1 Rotation decomposition

function DECOMPOSEROTATION(R)
  $\mathbf{out} \leftarrow R_3$ ▷ third column of R
  $\mathbf{z} \leftarrow \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$
  $\theta_{xy} \leftarrow \cos^{-1}(\mathbf{z} \cdot \mathbf{out})$
  $\mathbf{r}_{xy} \leftarrow \text{NORMALIZE}(\mathbf{out} \times \mathbf{z})$
  $\mathsf{R}_{xy} \leftarrow \exp_{SO(3)}(\theta_{xy} \cdot \mathbf{r}_{xy})$
  $\mathsf{R}_z \leftarrow \mathsf{R}_{xy}^T \mathsf{R}$
  return $\mathsf{R}_{xy}, \mathsf{R}_z$
end function

10. Degenerate configurations

Regarding the ability to self-calibrate from spherical motion with three views: near-degenerate configurations do occur when at least two of the three camera positions are extremely close together, and thus the three-view configuration approaches a two-view configuration.

11. Convergence analysis

Figure 5 plots cost function (12) versus focal length for each image sequence in our various datasets. As explained in Section 4.2, we employ random search followed by iterative optimization to find the minimum of the cost function and thus initialize the focal length and camera rotations. The analysis shows that the minimum of the cost function is always near the true focal length value. In the PhoneSweep datasets, where the camera moves in a rough circle, there are other local minima at higher focal length values; these correspond to solutions where the camera motion completes several circles rather than a single circle. In the DeepView dataset, where we have a hemi-sphere of cameras, there is one clear minimum.

12. Synthetic data experiment

In each iteration of the synthetic data experiment we generate a spherical relative pose problem where the two cameras have a relative rotation angle between 0 and 10 degrees about the Y-axis, with outward-facing motion. We select fifteen random 3D points in front of the first camera at a random depth between 4 and 8 and project them to the second camera with a focal length of 600 px, ensuring that no point lies behind the second camera. We then add Gaussian noise with 1 px std. dev. to the 2D point observations.

Since the seven-point solver returns multiple solutions we choose the best solution according to the Sampson error metric.

Table 4. Results of our ablation study on the PhoneSweep dataset.

Method	RRA@5↑	RTA@5↑	AUC@30↑
Five point solver	87.59	73.27	79.78
No focal length search	58.41	46.38	53.72
No spherical BA	99.82	78.80	87.65
Ours	100.00	84.83	90.87

13. Ablation study

We performed an ablation study to evaluate the effect of the first three steps, which are unique to our approach. We tested the following ablations:

- Five point solver: Use the standard five-point essential matrix solver [32] instead of the three-point solver [37]
- No focal length search: Instead of searching for the optimal focal length to initialize the rest of the pipeline, use the initial focal length f_{guess}
- No spherical BA: Use directly the results of 1D focal length search in COLMAP without employing spherical BA first

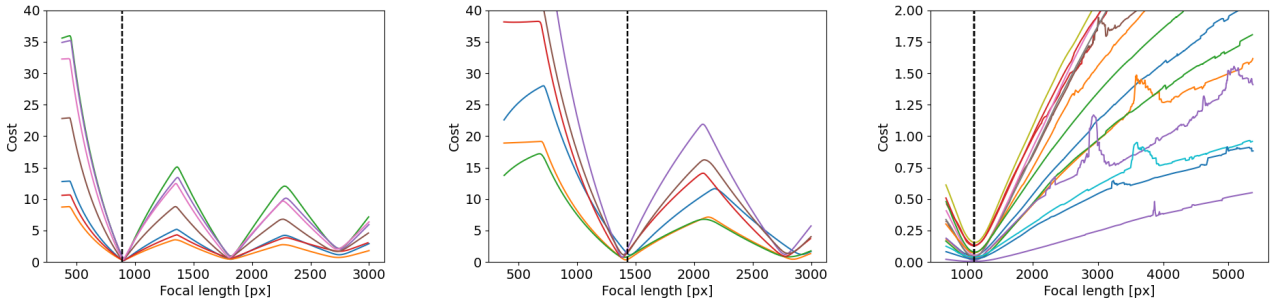
We performed the ablation study using the PhoneSweep dataset. The results are summarized in Table 4. As can be seen, each ablation leads to a drop in performance.

14. Applications

To highlight possible applications of our method, we tested view synthesis and dense depth estimation (Figure 6). We use our estimated camera poses and 3D points as input to Splatfacto-W [42], a view synthesis method based on 3D Gaussian Splatting [20].

In the PhoneSweep dataset, since the input videos were captured handheld, the camera path was roughly circular but naturally moved up and down and in and out (Figure 7). We rendered the synthesized views on a perfect circle of radius 0.5. Note that in our spherical SFM pipeline the cameras are initialized to a unit sphere. We didn't test more extreme deviations from the original views since in our videos the amount of parallax is relatively slight and so the view synthesis problem is not well-constrained.

The visual quality of the results suggests that our camera pose estimates are accurate enough to enable compelling downstream applications.



(a) iPhone 13 (b) Nexus 5X (c) Deep View

Figure 5. Plots of cost function (12) versus focal length. Each colored line represents the cost function for one image sequence from the dataset. The black dashed line indicates the average ground truth focal length value for the dataset.

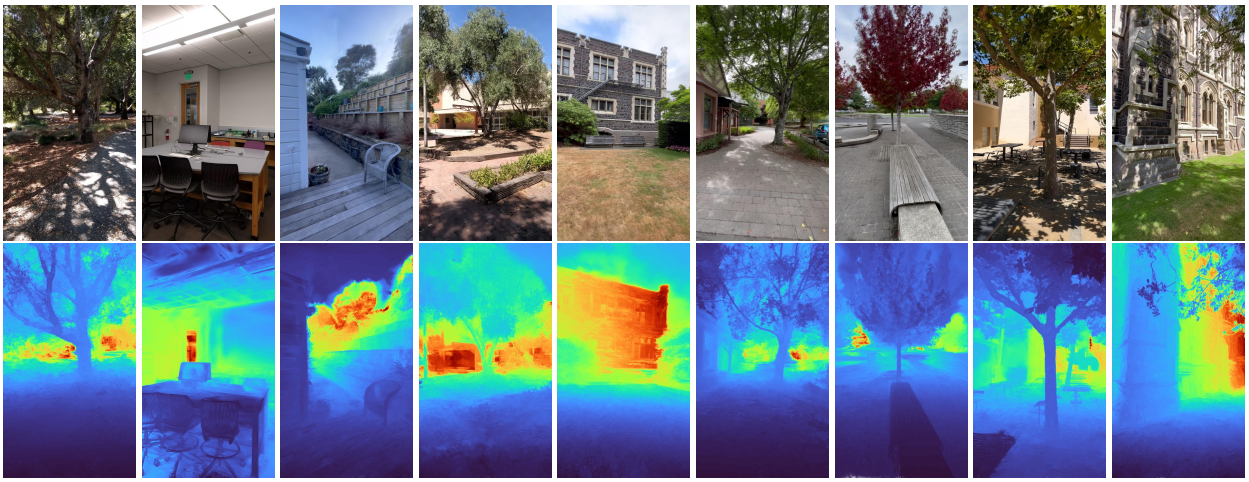
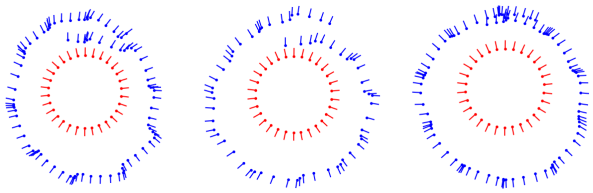


Figure 6. Example view synthesis (top row) and dense depth (bottom row) results from running Splatfacto-W [42] on Phone Sweep scenes using our SFM results for the input camera poses and 3D point cloud.



(a) *arboretum4* (b) *deck* (c) *scott_shand*

Figure 7. Top-down visualization of input camera viewpoints (blue) and a subset of the synthesized viewpoints (red) for three sequences from the PhoneSweep dataset. The dot indicates the camera center and the line indicates the viewing direction.