Structure from Motion

Rachel Moan
UIUC
Champaign, Illinois
rmoan2@illinois.edu

Vlas Zyrianov
UIUC
Champaign, Illinois
vlasz2@illinois.edu

1. Introduction

The 3D reconstruction of a scene is a fundamental problem in computer vision. This problem is particularly challenging if you have only one image because the image formation process is not invertible, meaning that a 3D point cannot be recovered from a 2D point in an image [7]. A solution to this is to use multiple views of the same scene in order to recover the scene's 3D structure. This is called the Structure from Motion (SfM) problem.

The goal of SfM is to jointly "estimate both the three-dimensional positions of the points in some fixed coordinate system (the scene *structure*) and the projection matrices associated with the cameras observing them (or, equivalently, the apparent *motion* of the cameras relative to the points)" [7]. Creating an accurate structure and motion estimate is a challenging problem because the data the we use is imperfect: there are often small inaccuracies and detecting outliers can be difficult. Additionally, there is often some ambiguity is a scene due to occlusions or changes in lighting.

The Structure from Motion problem, while challenging, is essential to many computer vision and robotics tasks. This includes augmented reality, Simultaneous Localization and Mapping (SLAM), scanning or monitoring structures, and creating 3D maps for autonomous vehicles. At high level, the SfM pipeline has the following steps:

1. Extract keypoints
2. Calculate the fundamental and essential matrices of the cameras
3. Compute 3D points
4. Use Bundle Adjustment to minimize the reprojection error of the 3D reconstruction

2. The Classic Approach

2.1. Initialization

2.1.1 Keypoint Extraction

The first step of the standard SfM pipeline is to extract keypoints. Representative works include SIFT [16], SURF [3], and superpoint [5].

In the SfM task, (1) keypoints are assumed to have previously been matched and that (2) there are some errors and incorrect matches [7]. Robustness to incorrect matches is achieved through methods such as RANSAC (see 2.1.5).

2.1.2 Eight Point Algorithm

The Eight Point Algorithm [15] addresses the second step of the Structure From Motion problem, which is to calculate the essential and fundamental matrices of the cameras. The fundamental matrix, $\mathbf{F}$, encapsulates epipolar geometry. To find $\mathbf{F}$ given a correspondence, $\mathbf{x} \leftrightarrow \mathbf{x}'$, we first construct the homogeneous system that satisfies the epipolar constraint $\mathbf{x}^\intercal \mathbf{F} \mathbf{x} = 0$ where

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \mathbf{x}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}, \text{and } \mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}.$$

If we have n point correspondences, we can then write the linear system $\mathbf{A}\mathbf{f} = 0$ where

$$A = \begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \dots & & & & & & & & \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix}$$

and

$\mathbf{f} = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{21} & f_{22} & f_{23} & f_{31} & f_{32} & f_{33} \end{bmatrix}^\intercal$.

Note that we can solve this system with $n = 8$ (instead of 9) by fixing the undetermined scale of the fundamental matrix. However, if $n \geq 8$, the accuracy of the estimation of the fundamental matrix will be improved.

If the camera calibration is known, then we can calculate the essential matrix $\mathbf{E} = \mathbf{K}'^\intercal \mathbf{F} \mathbf{K}$ [8]. The essential matrix directly tells us the rotation and translation between two camera positions. This can then be used to determine the relationship between the 3D points in the scenes.

2.1.3 Normalized Eight Point Algorithm

While the standard Eight Point Algorithm is simple to implement, it is also highly susceptible to noise. This makes it unsuitable for real data. The normalized eight point algorithm [9] improves upon the standard 8 point algorithm by

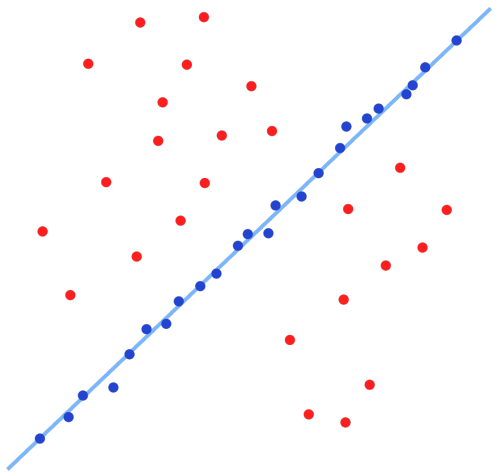

Figure 1. RANSAC algorithm. Despite the large amount of outliers and erroneous points (red), RANSAC allows finding the optimal best-fit solution. Credits: Wikipedia.org

normalizing the input points. This normalization has two parts:

1. Translate the points so that their centroid is at the origin. This ensures that small differences between points are not obscured by large offsets. For example, 100.1 and 100.2 are not significantly different, but .1 and .2 are significantly different.
2. Scaling the points so that the average distance is $\sqrt{2}$ from the origin. This ensures that, on average, the points are of the form $[1, 1, 1]^\intercal$.

This normalization significantly increases the accuracy of the estimation of the fundamental matrix.

2.1.4 Trifocal Tensor

The trifocal tensor method allows reconstructing three views at once [8]. Some SfM algorithms utilize this method to perform the initial reconstruction among three views instead of two.

2.1.5 RANSAC

RANSAC [6] is an iterative approach used to fit models in situations where data may contain a significant portion of outliers. In the case of Structure from Motion, RANSAC plays a critical role in robustly matching keypoints across images. Keypoints matching can have erroneous matches which can cause poor initial pose estimates.

2.2. Bundle Adjustment

In practice, the camera positions estimated the Normalized Eight Point Algorithm are not optimal as the algorithm is greedy and focuses on local pairs of cameras, instead of all cameras globally. Bundle Adjustment addresses this problem by jointly optimizing all the camera positions and keypoints.

For this reason, bundle adjustment is used as a post-processing step in the SfM pipeline. Specifically bundle adjustment attempts to jointly optimize camera parameters ($\{\mathbf{R}_k, \mathbf{t}_k\}$) and 3D locations ($\{\mathbf{X}_n\}$), with m cameras and n 3D points:

$$\min_{\{\mathbf{R}_m, \mathbf{t}_m\}, \{\mathbf{X}_n\}} \sum_n \sum_m \|\mathbf{x}_{nm} - \pi_{\mathbf{K}}([\mathbf{R}_m|\mathbf{t}_m]\mathbf{X}_n)\|_2^2$$

To simplify this expression, the sets $\{\mathbf{R}_k, \mathbf{t}_k\}$ and $\{\mathbf{X}_n\}$ are combined together into a single set , y_i. The ground truth 2d coordinate is replaced with x_i.

$$\min_y E(y_i, x) = \min_y \sum_i \|\pi_i(y_i) - x_i\|_2^2$$

This expression can further be simplified through linearization with Taylor expansion. The can be directly optimized with gradient descent as follows:

$$y^{(t+1)} = y^{(t)} - \gamma \nabla_y E(y_i, x)$$

2.2.1 Levenberg Marquardt Algorithm

A common approach to optimizing Bundle Adjustment is the Levenberg Marquardt algorithm [4], which resembles a ridge-regression with an automatically adjusting dampening factor.

Algorithm 1 Levenberg-Marquardt Algorithm

$\lambda = 10^{-4}$
$t = 0$
repeat
  $A, b \leftarrow$ linearize $E(y_i)$ at y^t
  $\Delta \leftarrow$ solve $(A^T A + \lambda \text{diag}(A^T A))\Delta = A^T b$
  if $E(y^t + \delta) < E(y^t)$ **then**
    $y^{t+1} = y^t + \Delta$
    $\lambda \leftarrow \lambda / 10$
  else
    $y^{t+1} = y^t$
    $\lambda \leftarrow 10\lambda$
  end if
until convergence
return y^t

2.3. Global and Incremental SfM

SfM pipelines can be broadly split into two categories: incremental and global. Incremental methods [19, 23, 25] operate iteratively. At each step, camera transformations

are estimated for a set of images that pass a threshold during RANSAC [19]. After that bundle adjustment is used to jointly optimize the estimate, after which the process is repeated. In contrast, global methods process the entire collection of images at once [11, 20, 22, 29]. Sim et al. [24] optimizes point angles wrt view. Moulon et al. [20] optimize point Euclidean distance.

2.4. Applications: Large-Scale reconstruction

SfM has been successfully applied for large-scale reconstruction of real-world environments from in-the-wild and internet images.

Photo Tourism [25] provides an SfM-based interface to browse photo collections geometrically. Agarwal et al. [1] demonstrate how landmarks in Rome can be reconstructed from internet images acquired from Flickr. Running SfM at a large scale is a computationally demanding activitiy. To accelerate the process, Agarwal et al. [2] propose the use of the Schur Complement trick in preconditioning and Conjugate Gradients for bundle adjustment. Heinly et al. [10] further scale up SfM with a streaming-based approach to reconstruct landmarks from accross the world using images acquired from Yahoo.

2.5. Applications: SLAM

Simultaneous Localization and Mapping (SLAM) is a specific instantiation of the SfM problem where the camera motion and scene structure is estimated in real time.

3. Modern Approaches

Modern techniques improve upon classic techniques in two primary ways: (1) Better internal representations, such as NerF, Gaussian Splats, etc., which provide improvements in novel-view rendering, more accurate reconstructions, and richer details for downstream tasks. This emphasizes the ”structure” in structure from motion. (2) Better priors, typically achieved through deep-learning data-driven approaches, enable the estimation of camera locations in under-constrained environments.

3.1. Improvements in Structure estimation

One way to improve upon traditional SfM approaches is to use better representations for the structure itself. More accurate reconstructions of 3D scenes not only gives us a better rendering, but it could also be used to better inform SfM in the future. Better representations of the scene will give us a better estimate of the scene structure.

3.1.1 Neural Radiance Fields

One way to represent a scene is to use Neural Radiance Fields (NeRF) [18]. NeRF utilizes learning to create a continuous representation of a scene. It performs volume rendering by marching along pixel rays and querying a neural network to get the color and opacity of each pixel. This volume rendering technique is differentiable, so NeRF is optimized to minimize rendering loss. This optimization results in state of the art renderings. NeRF's main downside is that it is slow to train and render. In practice, marching along pixel rays is a bottleneck in rendering time and it is too slow to be used online for problems like SLAM, which require fast rendering. Additionally, NeRF relies on knowledge of exact camera poses in order to learn scene representations. Bundle-Adjusting Neural Radiance Fields (BARF) [14] proposes training NeRF with inaccurate or nonexistant camera poses. It uses Bundle Adjustment to jointly optimize for both registration and reconstruction. The results demonstrate that BARF has similar high-quality results as NeRF, but is more applicable to SLAM applications since it uses imperfect data.

3.1.2 Signed Distance Fields and Meshes

Another method to represent a 3D scene is to use meshes. Meshes have the advantage of being memory efficient and they are excellent for representing the texture of a scene. One method that utilizes meshes is NeuralRecon [26]. NeuralRecon performs 3D scene reconstruction in real time using monocular video with known camera poses. The idea is to “jointly reconstruct and fuse sparse truncated signed distance field (TSDF) volumes for each video fragment incrementally” [26]. Then they convert the TSDF to a highly accurate mesh representation. NeuralRecon has a coarse-to-fine structure that gradually improves the accuracy of the TSDF at each iteration. Unlike prior work, which estimates depth maps frame-by-frame and then fuses the results later, NeuralRecon directly reconstructs local features by representing them as a TSDF volume. This allows the network to learn local smoothness and produce a locally coherent geometric estimation. To ensure that the reconstruction is globally consistent, NeuralRecon uses gated recurrent units in which the current reconstructed fragment is conditioned on the previous global reconstruction. The resulting mesh is dense, highly accurate, and globally consistent.

Scaling up to large scenes while retaining high-quality reconstructions is a significant challenge in SfM. NICE-SLAM [31] is another algorithm that produces an accurate mesh and scales well to large scenes. Taking inspiration from NeRF, NICE-SLAM uses a neural network to represent the scene as a continuous function. It combines the accuracy of this neural implicit representation with the scalability of a mesh representation. It is a hierarchical, grid-based method that iteratively improves its estimate of the scene. NICE-SLAM scales well to large scenes because it specifically allows for local updates to be made, which is lacking in global approaches to SfM.

3.1.3 Gaussian Splatting

One modern representation that has gained traction in recent years is gaussian splatting. Gaussian splatting is a type of rasterization technique that draws several gaussians, each with a position, covariance matrix (which controls the shape), color, and opacity value. The result is a much smoother, more accurate, and dense 3D structure estimation. In contrast, NeRF renders a scene by using marching rays [18], which slows down the rendering time. Gaussian splats, however, are much faster to render and they take advantage of a 3D scene's naturally sparse structure.

In order to render a 3D scene, most 3D gaussian splatting methods first estimate motion offline with standard SfM techniques, and then render the scene using gaussian splats. This, however, is significantly limited by the fact that one must first run SfM offline, which makes this approach unsuitable for applications such as SLAM. Gaussian Splatting SLAM [17], however, does not rely on running SfM offline. Instead, they utilize 3D Gaussians as their only SLAM representation. Additionally, they perform differentiable rasterization with the gaussian splats, which allows them to capture fine scene details and represent challenging object structure by directly differentiating information from all of the pixels [17]. This results in a SLAM pipeline that is able to capture scenes more accurately and render objects with more materials than prior SLAM methods.

SplaTAM, another SLAM method that utilizes gaussian splatting, achieves camera pose estimation and map estimation that is 2 times better than non-gaussian methods [12].

3.2. Improvements in Motion estimation

Recently, learning-based approaches for estimating camera transformations have become popular. Adopting representations that neural networks can effectively operate on is a central challenge in 3D vision problems. Directly estimating rotation matrices has not been shown to work well in practice. For this reason, works innovate by adopting novel representations that are suitable for deep learning.

Cameras as Rays [30] adopts a ray-based camera representation for camera extrinsics and intrinsics that is based on 6D Plucker coordinates. The proposed ray-based representations offers a large degree of redundancy and is camera-centric, making it highly suitable for training models. Using this representation, the authors train a transformer to denoise camera positions directly from images.

Dust3R [27] adopts a point-based representation. Through this representation, the authors train a transformer (based on the CroCo architecture [28]) that takes 2 images as input, performs cross-attention on features between the two imgaes, and predicts per-pixel 3D points for both input images (i.e., the final output of the model is two sets of point clouds). To address the scale ambiguity, a normalizing loss term is added that keeps the average of the points within a predefined range. In addition, the predicted 3D points for the second images are within the coordinate frame of the first image. Because of this, matching points can be effectively done through Nearest-Neighbors. Based on the matches, camera extrinsics can be estimated through Epipolar geometry. The method is fully feed-forward and doesn't require repeated optimization.

Mast3R [13] builds on the Dust3R architecture and adds a head that directly predicts matching features. Through this modification, Mast3R can achieve half the pose error rate of Dust3R.

4. Conclusion and Future Work

Estimating the structure of a scene and the motion of cameras is an essential problem to many tasks in computer vision, especially SLAM. Current work in SfM is able to solve the problem under various challenging scenarios, yet there are many areas for future improvement. The feature extraction step is often the most computationally expensive and while existing methods are quite accurate, more work could be done to improve the efficiency of this step [21]. Obstruction and ambiguity also still present a challenge in SfM and future work could focus on how to better fill in missing or ambiguous information. Recent papers tend to focus on either improving the quality of the estimated structure or the quality of motion estimation. In the future, we expect to see a unification of these two high level directions: better structures can be exploited to estimate more accurate motion and vice versa.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *ICCV*, 2009. 3

[2] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11*, pages 29–42. Springer, 2010. 3

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 1

[4] Frank Dellaert and Michael Kaess. 2017. 2

[5] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 1

[6] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, page 381–395, 1981. 2

[7] David A. Forsyth and Jean Ponce. *Computer Vision - A Modern Approach, Second Edition.* Pearson, 2012. 1

[8] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. 1, 2

[9] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997. 1

[10] Jared Heinly, Johannes L. Schönberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world in six days. In *CVPR*, 2015. 3

[11] Nianjuan Jiang, Zhaopeng Cui, and Ping Tan. A global linear method for camera pose registration. In *ICCV*, 2013. 3

[12] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In *CVPR*, 2024. 4

[13] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024. 4

[14] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 3

[15] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293 (5828):133–135, 1981. 1

[16] David G. Lowe. Distinctive image features from scale-invariant keypoints. *ICCV*, 60:91–110, 2004. 1

[17] Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian Splatting SLAM. In *CVPR*, 2024. 4

[18] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3, 4

[19] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Adaptive structure from motion with a contrario model estimation. In *ACCV*. 2, 3

[20] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *ICCV*, 2013. 3

[21] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017. 4

[22] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global Structure-from-Motion Revisited. In *ECCV*, 2024. 3

[23] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2

[24] Kristy Sim and R. Hartley. Recovering camera motion using l∞ minimization. In *CVPR*, 2006. 3

[25] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA, 2006. ACM Press. 2, 3

[26] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. NeuralRecon: Real-time coherent 3D reconstruction from monocular video. *CVPR*, 2021. 3

[27] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 4

[28] Weinzaepfel, Philippe and Leroy, Vincent and Lucas, Thomas and Brégier, Romain and Cabon, Yohann and Arora, Vaibhav and Antsfeld, Leonid and Chidlovskii, Boris and Csurka, Gabriela and Revaud Jérôme. CroCo: Self-Supervised Pre-training for 3D Vision Tasks by Cross-View Completion. In *NeurIPS*, 2022. 4

[29] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *ECCV*, 2014. 3

[30] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *ICLR*, 2024. 4

[31] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3