Monocular Visual Odometry and Structure from Motion

Jaroslav Moravec

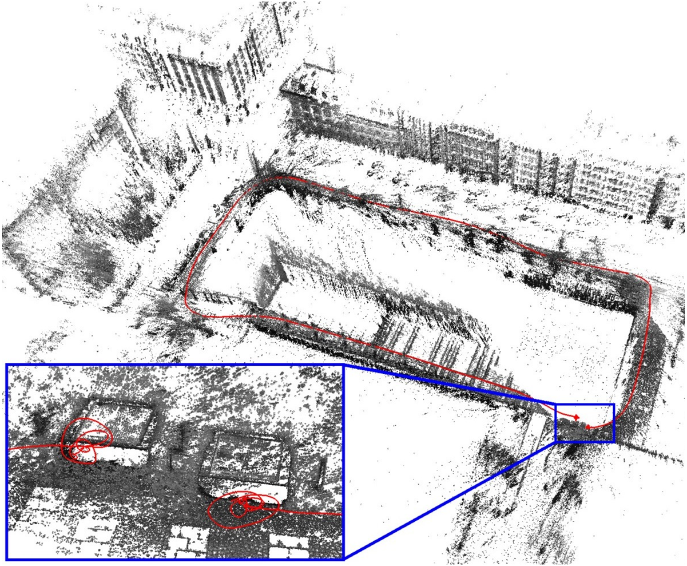
5. 11. 2021

Outline of the presentation

- Problem introduction
- Applications [22, 11, 28]
- Taxonomy of MVO
  - Direct vs Indirect
  - Sparse vs Dense
- Direct Sparse Odometry [8]
- CNN for pose and depth estimation [33]
- D3VO [31]
- Demo

Visual odometry and Structure from Motion

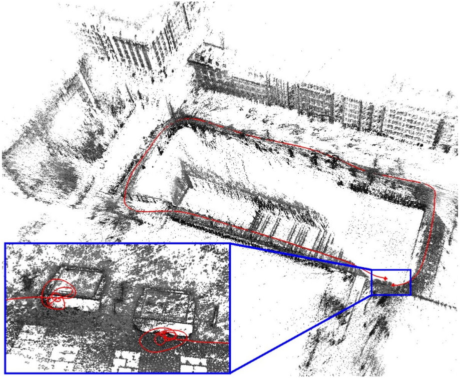
Problem introduction

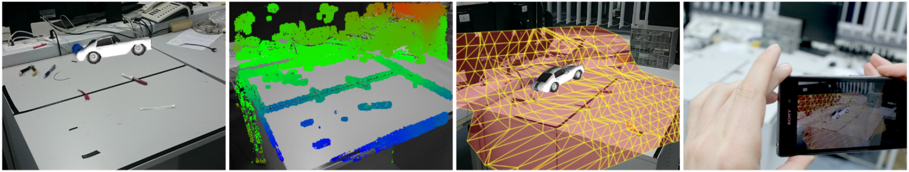


Applications

Visual odometry

- Navigation [18, 8]
  - Mars exploration [16, 5]
  - Aerial vehicles [14, 29]
  - Underwater vehicles [7, 9]
  - Automotive [33, 12, 31]
- Augmented reality [25, 4]
- Calibration [27, 13]





Applications

Structure from Motion

- Image-based 3D modeling [20, 23, 26, 10]
- Hand-eye calibration [1, 24]
- Augmented reality [17, 30]
- Video enhancement and stabilization [15, 32]
- Segmentation and recognition [3, 2]

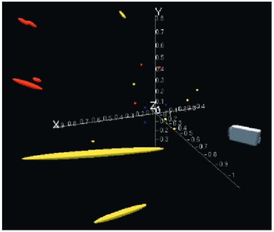
 

Taxonomy

Direct vs Indirect

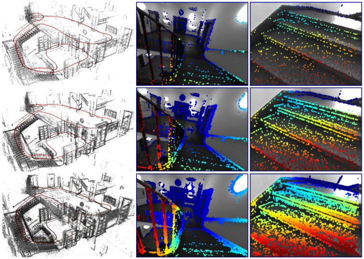
Indirect

- Generate an intermediate representation of raw measurements
- Using these intermediate values, calculate geometry and camera motion
- [21, 6]



Direct

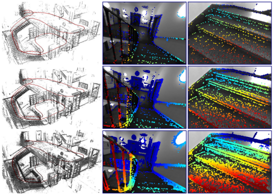
- Use the raw meassurements directly to calculate geometry and camera motion
- [8, 31, 12, 33]



Taxonomy

Dense vs Sparse

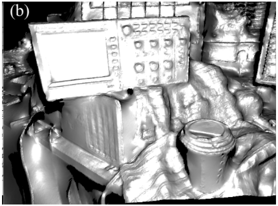
Sparse



- Use and reconstruct only a selected set of independent points (keypoints, e.g., corners)
- [21, 6, 8]

Dense

- Use and reconstruct all the points in the 2D image domain
- [12, 33, 19]

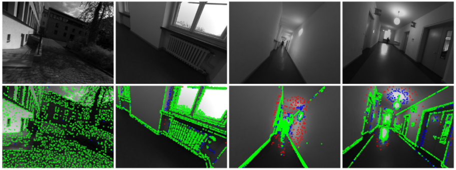


- Sparse $\Rightarrow$ use and reconstruct only small set of points

  $\rightarrow$ work with intensities
- Aim to keep a fixed number $N_p = 2000$ active points

1) Candidate point selection

- Choose points that are **well-distributed** in the image and have **high image gradient magnitude** w.r.t. their immediate surroundings
  1. Split the image to 32×32 regions
  2. Calculate an addaptive threshold gradient for that region $\overline{g} + g_{th}$
  3. Split the image to $d \times d$ blocks $\rightarrow$ select pixel with highest gradient magnitude if it surpasses region threshold



Direct Sparse Odometry (**DSO**)

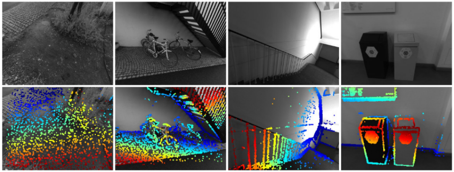
Points selection

2) Candidate point tracking

- Selected candidate points are tracked in subsequent images
  $\rightarrow$ discrete search along the epipolar line
- Best match is used to compute the depth of the candidate point

3) Candidate point activation

- Select new active points after maginalization of the old ones
- Candidate points are activated based on their distance from other active points



Direct Sparse Odometry (**DSO**)

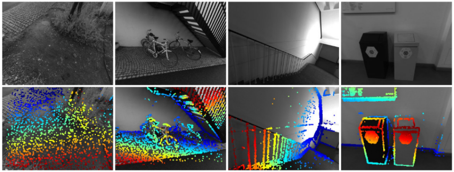
Frames selection

- Direct ⇒ use raw measurements

  → work with images
- Keep a window of $N_f = 7$ reference images (keyframes)

1) Initial frame tracking

- Tracking new frame w.r.t. latest KF:
  1. two-frame direct image alignment
  2. multi-scale image pyramid
  3. constant motion mode
- If the RMSE is still high, try RANSACing rotation



Direct Sparse Odometry (**DSO**)

2) Keyframe creation

- New KF is created based on three criterion:
  1. Field of view should change → mean square optical flow:

$$f = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||p - p'||^2}$$

  2. Translation causes occlusions and disocclusions → mean OF without rotation:

$$f_t = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||p - p'_t||^2}$$

  3. Camera exposure time changed significantly:

$$a = \left| \log\left(e^{a_j - a_i} t_j t_i^{-1}\right) \right|$$

- A new KF is taken if:

$$w_f \cdot f + w_{f_t} \cdot f_t + w_a \cdot a > T_{kf}$$

Direct Sparse Odometry (**DSO**)

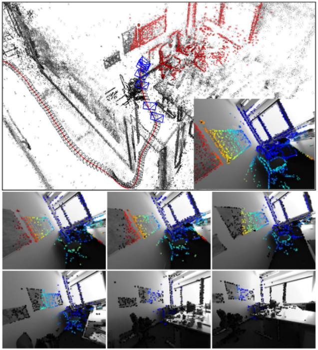
Frames selection

3) Keyframe marginalization

- Given active KFs $I_1, \ldots, I_n$:
  1. We keep two latest KFs I_1, I_2
  2. If only 5 % of KF points is visible in I_1, it is marginalized
  3. If more than 7 KFs are active, we marginalize frames that are distant from others:

$$s(I_i) = \sqrt{d(i,1)} \sum_{j \in \{3,\ldots,n\} \setminus i} \frac{1}{d(i,j) + \varepsilon}$$

- We first marginalize points in the KF and then the KF itself



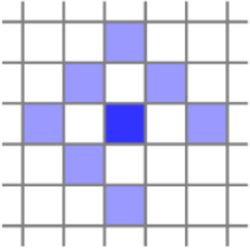
Direct Sparse Odometry (**DSO**)

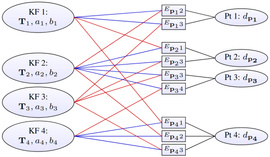
Photometric error and optimization

- Given a reference image I_i and a target image I_j, the photometric error of a point $\mathbf{p} \in I_i$ is defined as:

$$E_{\mathbf{p}}^{j} = \sum_{\mathbf{p} \in \mathcal{N}_{\mathbf{p}}} w_{\mathbf{p}} \left\| \left(I_j[\mathbf{p}'] - b_j\right) - \frac{t_j e^{a_j}}{t_i e^{a_i}} \left(I_i[\mathbf{p}] - b_i\right) \right\|_{\gamma},$$

$$p' = P_c(\mathbf{R} P_c^{-1}(\mathbf{p}, d_{\mathbf{p}}) + \mathbf{t}), \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \mathbf{T}_j \mathbf{T}_i^{-1}$$





- The total error is:

$$E_{\text{photo}} = \sum_{i \in \mathcal{F}} \sum_{\mathbf{p} \in \mathcal{P}_i} \sum_{j \in \text{obs}(\mathbf{p})} E_{\mathbf{p}}^{j}$$

- Optimizing $(\mathbf{T}_i, \mathbf{T}_j, d, \mathbf{c}, a_i, a_j, b_i, b_j)$ with sliding window using Gauss-Newton algorithm

Direct Sparse Odometry (**DSO**)

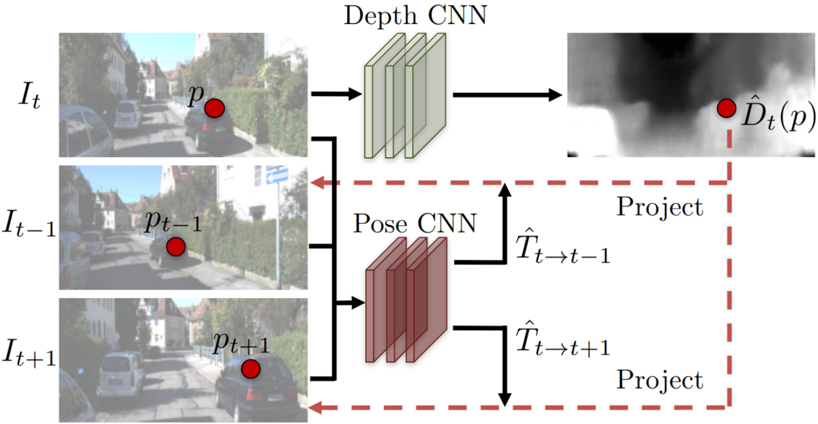
Experiments

SfMLearner

Overview

[33]

- Jointly train two different CNNs to predict depth and pose



SfMLearner

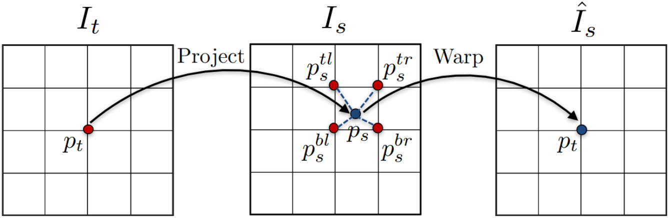
Loss function

- Given some target image I_t and source image I_s, the objective is formulated using view synthesis:

$$\mathcal{L}_{\text{vs}} = \sum_s \sum_{p \in I_t} |I_t(p) - \hat{I}_s(p)|$$

- I.e.,

$$p_s \sim P_c(\hat{T}_{t \to s} P_c^{-1}(\mathbf{p}_t, \hat{D}_t(\mathbf{p}_t)))$$



SfMLearner

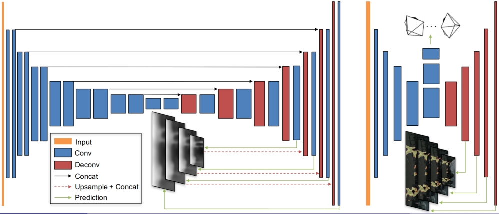
Explainability and network architectures

- There are many assumption on monocular view synthesis
  $\rightarrow$ To make the process more robust, they also use the explainability network that predicts, where the view synthesis will be succesful
- The loss function for view synthesis is then:

$$\mathcal{L}_{vs} = \sum_s \sum_{p \in I_t} \hat{E}_s(p) |I_t(p) - \hat{I}_s(p)|$$

- The total loss is defined as:

$$\mathcal{L}_{total} = \mathcal{L}^l_{vs} + \lambda_s \mathcal{L}^l_{smooth} + \lambda_e \sum_s \mathcal{L}_{reg}(\hat{E}^l_s)$$



SfMLearner

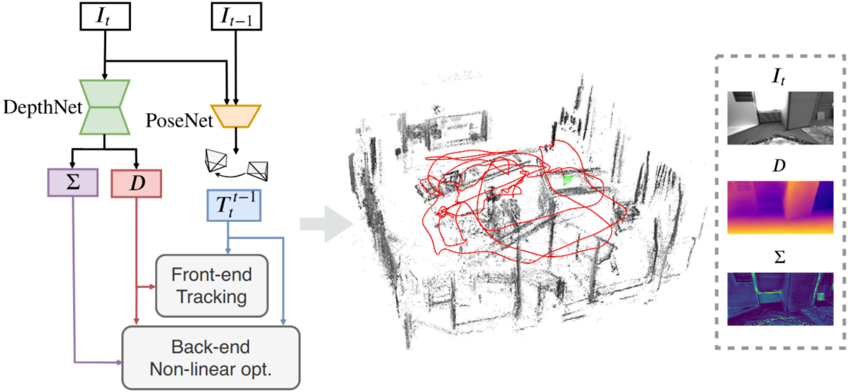
Experiments

D3VO

Overview

[31]

- Combines the previous two methods:
    1. Self-supervised networks for depth, pose and uncertainty
    2. Windowed sparse photometric bundle adjustment



D3VO

Self-supervised networks

- Minimize the photometric reprojection error:

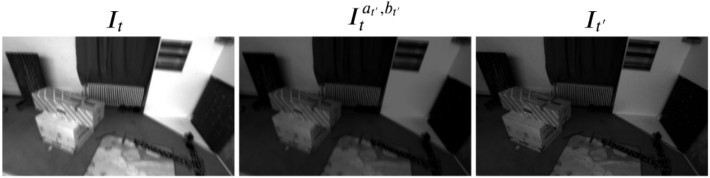
$$\mathcal{L}_{\text{self}} = \frac{1}{n} \sum_{\mathbf{p} \in I_t} \min_{t'} r(I_t, \hat{I}_{t'}), \text{ where}$$

$$r(I_a, I_b) = \frac{\alpha}{2}(-\text{SSIM}(I_a, I_b)) + (1-\alpha)||I_a - I_b||_1$$

- Modeling the change of camera exposure:

$$I^{a,b} = aI + b$$

$$\implies \mathcal{L}_{\text{self}} = \frac{1}{n} \sum_{\mathbf{p} \in I_t} \min_{t'} r(I_t^{a_{t'}, b_{t'}}, \hat{I}_{t'})$$



D3VO

Self-supervised networks: Uncertainty

- Uncertainty Σ_t works similarly to the exaplainability in SfMLearner:

$$\mathcal{L}_{\text{self}} = \frac{1}{n} \sum_{\mathbf{p} \in I_t} \frac{\min_{t'} r(I_t^{a_{t'}, b_{t'}}, \hat{I}_{t'})}{\Sigma_t} + \log \Sigma_t$$

- The total loss isthe combination of these self-spervised losses and the regularization losses on multiscale images:

$$\mathcal{L}_{\text{total}} = \frac{1}{s} \sum_{s} (\mathcal{L}^s_{\text{self}} + \lambda \mathcal{L}^s_{\text{reg}}), \text{ where}$$

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{smooth}} + \beta \left[\sum_{t'} (a_{t'} - 1)^2 + b_{t'}^2 \right]$$

- **DepthNet** Input: I_t, Output: D_t, D^s_t, Σ_t
- **PoseNet** Input: $(I_t, I_{t'})$, Output: $\mathbf{T}_t^{t'}, a_{t'}, b_{t'}$

D3VO

- Using predictions from self-supervised network $\hat{D}, \hat{\Sigma}, \hat{\mathbf{T}}_t^{t'}$
- Incorporating predictions to boost DSO [8]

1) Photometric energy

- Virtual stereo term $E_{\mathbf{p}}^{\dagger}$:

$$E_{\text{photo}} \sum_{i \in \mathcal{F}} \sum_{\mathbf{p} \in \mathcal{P}_i} \left(\lambda E_{\mathbf{p}}^{\dagger} + \sum_{j \in \text{obs}(\mathbf{p})} E_{\mathbf{p}}^{j} \right), \text{ where}$$

$$E_{\mathbf{p}}^{\dagger} = w_{\mathbf{p}} \left|\left| I_i^{\dagger}[\mathbf{p}^{\dagger}] - I_i[\mathbf{p}] \right|\right|_{\gamma}$$

2) Pose energy

$$E_{\text{pose}} \sum_{i \in \mathcal{F} \setminus 0} \log \left[\hat{\mathbf{T}}_{i-1}^{i} \mathbf{T}_{i}^{i-1} \right] \Sigma_{\hat{\zeta}_{i-1}^{i}}^{-1} \log \left[\hat{\mathbf{T}}_{i-1}^{i} \mathbf{T}_{i}^{i-1} \right]$$

$\Longrightarrow$ Optimize $E_{\text{total}} = E_{\text{photo}} + E_{\text{pose}}$ using the Gauss-Newton method

D3VO

Experiments

Demo
Calibration

- Obtain several pictures of some calibration target
- Detect markers positions from several locations and optimize camera parameters $\rightarrow$ OpenCV

$\Longrightarrow$ Intrinsic parameters of my phone camera:

$$\mathbf{K} = \begin{pmatrix} 1440.62 & 0 & 953.99 \\ 0 & 1443.11 & 551.98 \\ 0 & 0 & 1 \end{pmatrix}$$

Demo

Pose, depth and reconstruction

- 1080p, 30 fps video around school premises

- DSO [8]: Trajectory & Sparse reconstruction
- SfMLearner [33]: Trajectory & Dense reconstruction
- (MonoDepth [12]: Trajectory & Dense reconstruction)

Comparison with other odometries

- LiDAR vs Stero vs Mono

Visual Odometry / SLAM Evaluation 2012

40	RotRocc			0.88 %	0.0025 [deg/m]	0.3 s	2 cores @ 2.0 Ghz (C/C++)	☐
M. Buczko and V. Willert: Flow-Decoupled Normalized Reprojection Error for Visual Odometry. 19th IEEE Intelligent Transportation Systems Conference (ITSC) 2016.								
41	D3VO			0.88 %	0.0021 [deg/m]	0.1 s	1 core @ 2.5 Ghz (C/C++)	☐
N. Yang, L. Stumberg, R. Wang and D. Cremers: D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2020.								
42	SD-DEVO			0.88 %	0.0028 [deg/m]	0.06 s	1 cores @ 3.6 Ghz (C/C++)	☐
129	VISO2-M		code	11.94 %	0.0234 [deg/m]	0.1 s	1 core @ 2.5 Ghz (C/C++)	☐
A. Geiger, J. Ziegler and C. Stiller: StereoScan: Dense 3d Reconstruction in Real-time. IV 2011.								
130	MonoDepth2		code	12.59 %	0.0312 [deg/m]	1 s	1 core @ 2.5 Ghz (C/C++)	☐
C. Godard, O. Mac Aodha, M. Firman and G. Brostow: Digging into self-supervised monocular depth estimation. ICCV 2019.								
131	MEGO			12.89 %	0.0451 [deg/m]	0.75 s	1 core @ 2.5 Ghz (C/C++)	☐

Reference I

[1] Nicolas Andreff, Radu Horaud, and Bernard Espiau. "Robot hand-eye calibration using structure-from-motion". In: *The International Journal of Robotics Research* 20.3 (2001), pp. 228–248.

[2] Sid Yingze Bao et al. "Semantic structure from motion with points, regions, and objects". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 2703–2710.

[3] Gabriel J Brostow et al. "Segmentation and recognition using structure from motion point clouds". In: *European conference on computer vision*. Springer. 2008, pp. 44–57.

[4] Mingwei Cao et al. "Fast monocular visual odometry for augmented reality on smartphones". In: *IEEE Consumer Electronics Magazine* (2020).

Reference II

[5] Yang Cheng, Mark Maimone, and Larry Matthies. "Visual odometry on the Mars exploration rovers". In: *2005 IEEE International Conference on Systems, Man and Cybernetics*. Vol. 1. IEEE. 2005, pp. 903–910.

[6] Andrew J Davison et al. "MonoSLAM: Real-time single camera SLAM". In: *IEEE transactions on pattern analysis and machine intelligence* 29.6 (2007), pp. 1052–1067.

[7] Matthew Dunbabin et al. "A hybrid AUV design for shallow water reef navigation". In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE. 2005, pp. 2105–2110.

[8] Jakob Engel, Vladlen Koltun, and Daniel Cremers. "Direct sparse odometry". In: *IEEE transactions on pattern analysis and machine intelligence* 40.3 (2017), pp. 611–625.

Reference III

[9] Brendan P Foley et al. "The 2005 Chios ancient shipwreck survey: New methods for underwater archaeology". In: *Hesperia* (2009), pp. 269–305.

[10] Jan-Michael Frahm et al. "Building rome on a cloudless day". In: *European conference on computer vision*. Springer. 2010, pp. 368–381.

[11] Friedrich Fraundorfer and Davide Scaramuzza. "Visual odometry: Part ii: Matching, robustness, optimization, and applications". In: *IEEE Robotics & Automation Magazine* 19.2 (2012), pp. 78–90.

[12] Clément Godard et al. "Digging into self-supervised monocular depth estimation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 3828–3838.

Reference IV

[13] Ryoichi Ishikawa, Takeshi Oishi, and Katsushi Ikeuchi. "Lidar and camera calibration using motions estimated by sensor fusion odometry". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 7342–7349.

[14] Jonathan Kelly and Gaurav S Sukhatme. "An experimental study of aerial stereo visual odometry". In: *IFAC Proceedings Volumes* 40.15 (2007), pp. 197–202.

[15] Feng Liu et al. "Content-preserving warps for 3D video stabilization". In: *ACM Transactions on Graphics (ToG)* 28.3 (2009), pp. 1–9.

[16] Mark Maimone, Yang Cheng, and Larry Matthies. "Two years of visual odometry on the mars exploration rovers". In: *Journal of Field Robotics* 24.3 (2007), pp. 169–186.

Reference V

[17] Jonathan Mooser et al. "Applying robust structure from motion to markerless augmented reality". In: *2009 Workshop on Applications of Computer Vision (WACV)*. IEEE. 2009, pp. 1–8.

[18] Hans Peter Moravec. "Obstacle avoidance and navigation in the real world by a seeing robot rover". PhD thesis. Stanford University, 1980.

[19] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. "DTAM: Dense tracking and mapping in real-time". In: *2011 international conference on computer vision*. IEEE. 2011, pp. 2320–2327.

[20] Marc Pollefeys et al. "Image-based 3D acquisition of archaeological heritage and applications". In: *Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*. 2001, pp. 255–262.

Reference VI

[21] Rene Ranftl et al. "Dense monocular depth estimation in complex dynamic scenes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4058–4066.

[22] Davide Scaramuzza and Friedrich Fraundorfer. "Visual odometry [tutorial]". In: *IEEE robotics & automation magazine* 18.4 (2011), pp. 80–92.

[23] Grant Schindler, Panchapagesan Krishnamurthy, and Frank Dellaert. "Line-based structure from motion for urban environments". In: *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*. IEEE. 2006, pp. 846–853.

[24] Jochen Schmidt, Florian Vogt, and Heinrich Niemann. "Calibration–free hand–eye calibration: a structure–from–motion approach". In: *Joint Pattern Recognition Symposium*. Springer. 2005, pp. 67–74.

Reference VII

[25] Thomas Schöps, Jakob Engel, and Daniel Cremers. "Semi-dense visual odometry for AR on a smartphone". In: *2014 IEEE international symposium on mixed and augmented reality (ISMAR)*. IEEE. 2014, pp. 145–150.

[26] Sudipta N Sinha et al. "Interactive 3D architectural modeling from unordered photo collections". In: *ACM Transactions on Graphics (TOG)* 27.5 (2008), pp. 1–10.

[27] Zachary Taylor and Juan Nieto. "Motion-based calibration of multimodal sensor extrinsics and timing offset estimation". In: *IEEE Transactions on Robotics* 32.5 (2016), pp. 1215–1229.

[28] Ying-mei Wei et al. "Applications of structure from motion: a survey". In: *Journal of Zhejiang University SCIENCE C* 14.7 (2013), pp. 486–494.

Reference VIII

[29] Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. "Monocular-SLAM–based navigation for autonomous micro helicopters in GPS-denied environments". In: *Journal of Field Robotics* 28.6 (2011), pp. 854–874.

[30] Ming-Der Yang et al. "Image-based 3D scene reconstruction and exploration in augmented reality". In: *Automation in Construction* 33 (2013), pp. 48–60.

[31] Nan Yang et al. "D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 1281–1292.

[32] Guofeng Zhang et al. "Video stabilization based on a 3D perspective camera model". In: *The Visual Computer* 25.11 (2009), pp. 997–1008.

Reference IX

[33] Tinghui Zhou et al. "Unsupervised learning of depth and ego-motion from video". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1851–1858.