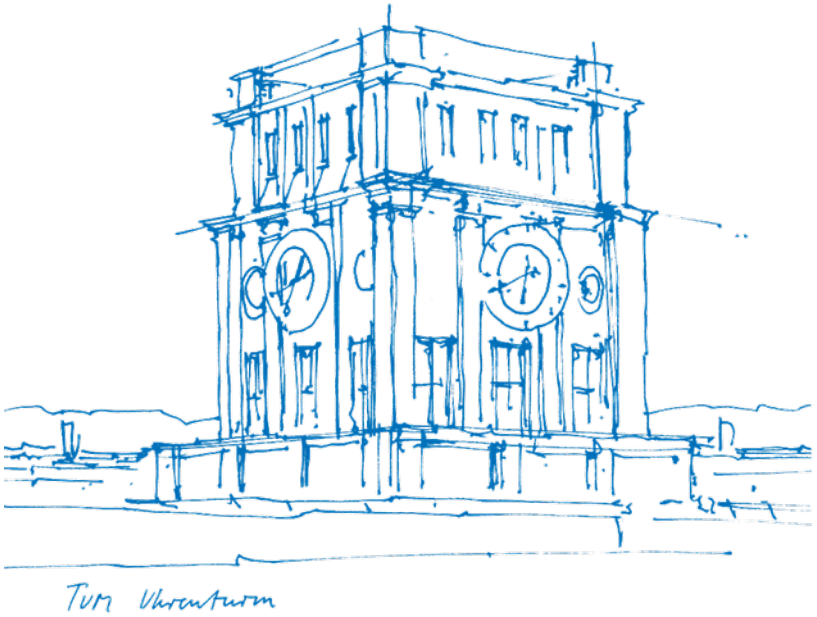
Depth from Motion for Smartphone AR

Camilo Rey
Supervisor: Lukas Köstler
Technical University of Munich
Munich. January 19, 2021



Outline

- Introduction
- Overview
- Method Description
- Experiments and Results
- Personal Comments
- Summary

Outline

- Introduction
- Overview
- Method Description
- Experiments and Results
- Personal Comments
- Summary

Introduction

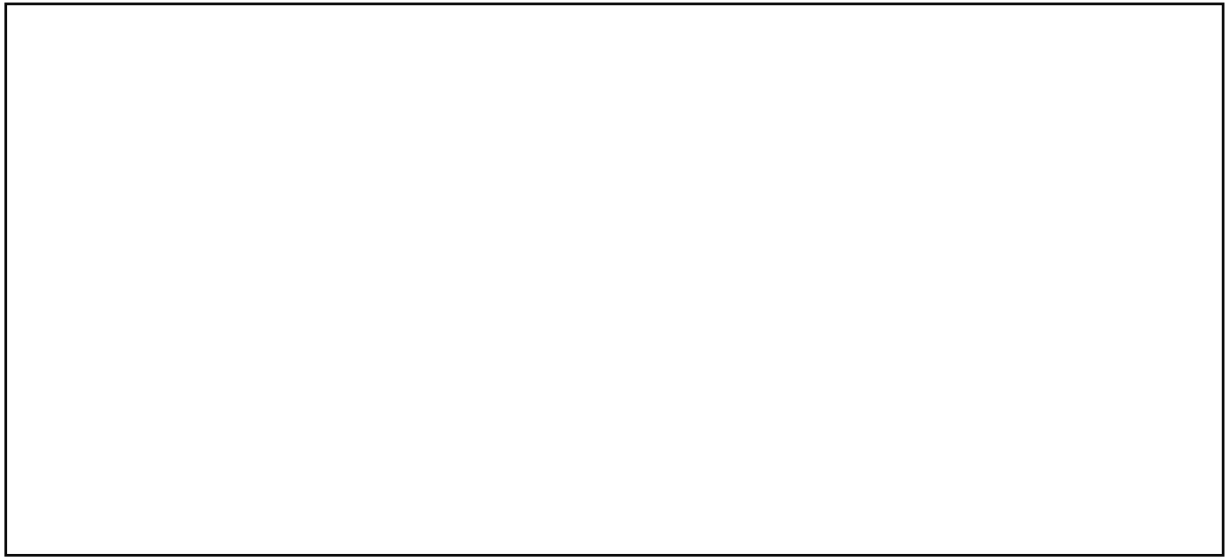


Fig. 1. ARCore Demo [1]

Introduction

Challenges of the task at hand:

- Support a wide range of phones
- Deliver dense depth at low latency and low computation

Achieved a novel pipeline capable of supplying depth maps at 30 Hz leveraging:

- Single RGB camera
- Single CPU

Introduction

Capabilities demonstrated on high-level AR Applications:

- Real-time navigation
- Shopping
- Fun photos

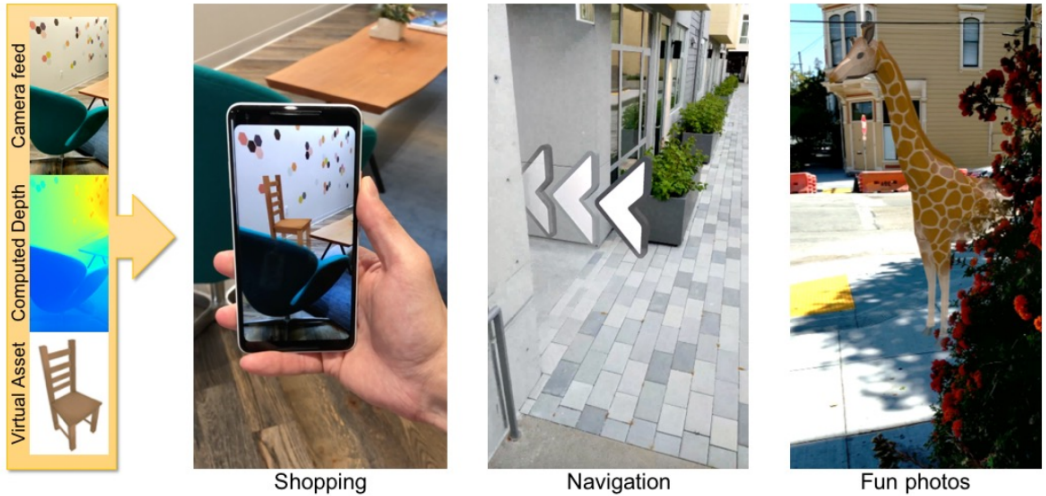
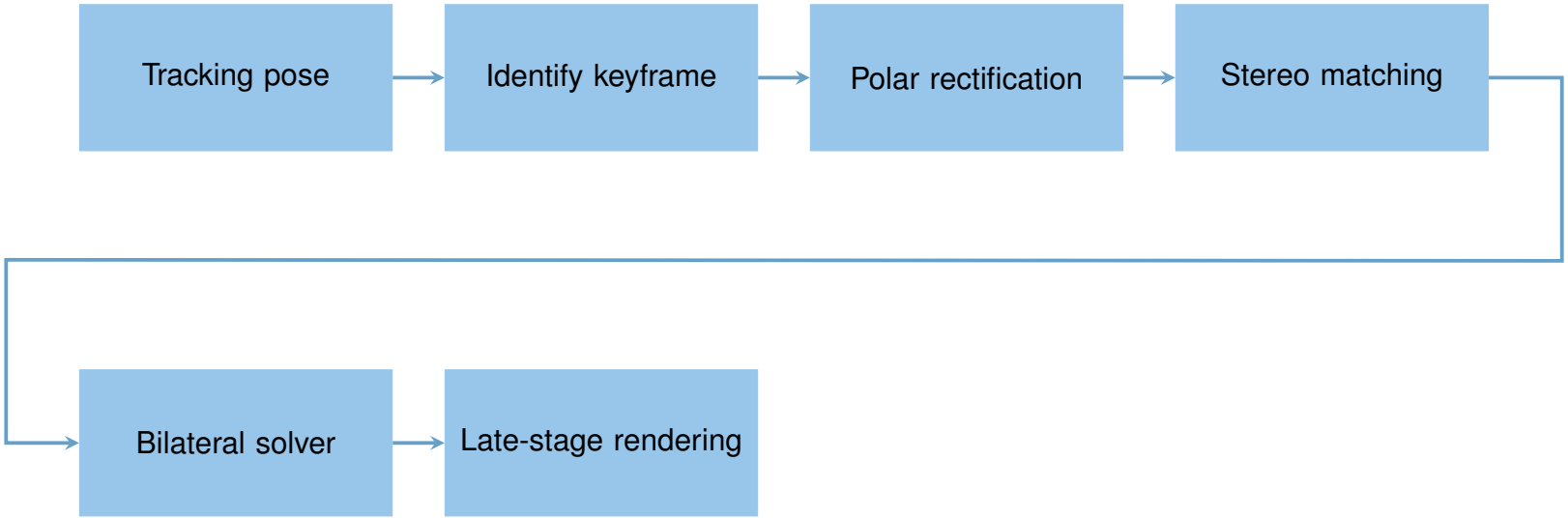


Fig. 2. AR Occlusions [2]

Outline

- Introduction
- Overview
- Method Description
- Experiments and Results
- Personal Comments
- Summary

Overview


Tracking pose
Identify keyframe
Polar rectification
Stereo matching
Bilateral solver
Late-stage rendering

Overview

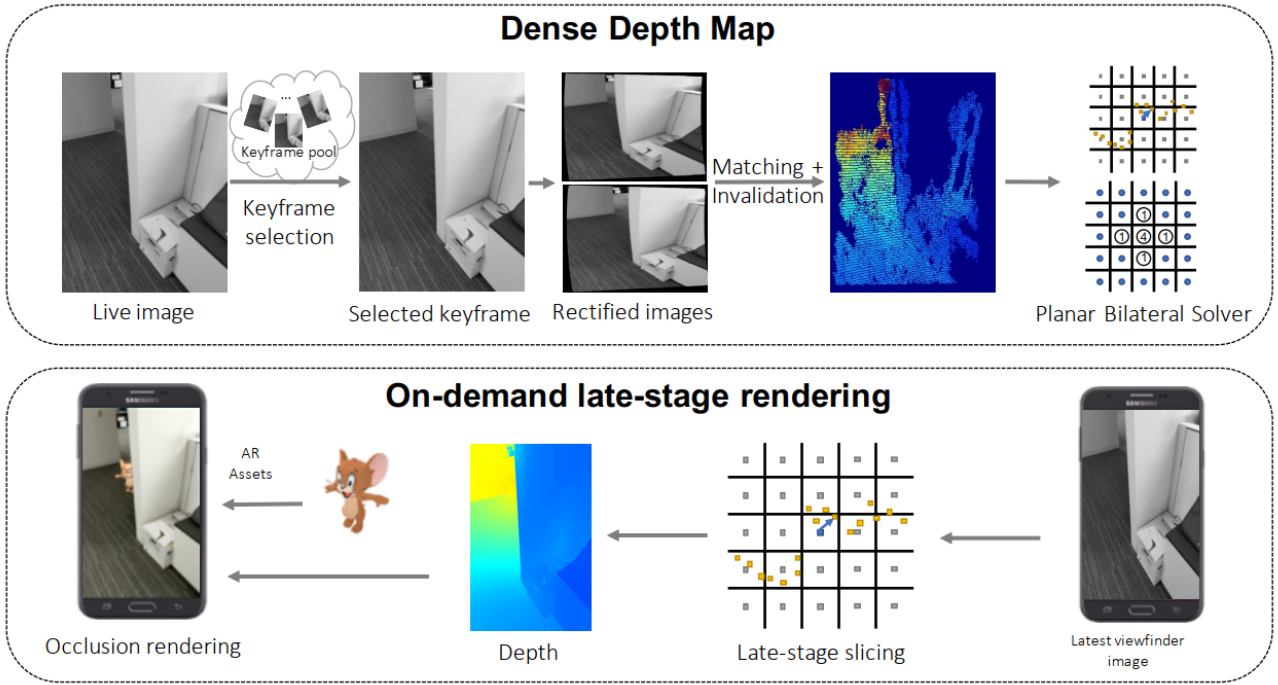


Fig. 3. Depth from Motion Pipeline [2]

Outline

- Introduction
- Overview
- Method Description
- Experiments and Results
- Personal Comments
- Summary

Keyframe Selection

Previous approaches for motion stereo:

- Keyframes from a fixed time delay [3], [4]
- Feature or geometry tracking [5], [6]
- Pixel-wise view selection [7], [8]

Soft-cost function is defined with:

- Fixed capacity pool of potential keyframes
- Metrics for the two frames evaluated
  - Baseline distance in 3D
  - Fractional overlap of the image areas
  - Measured error of pose-tracking statistics

Stereo Rectification

Some important aspects:

- Search can be constrained to a **1D problem**, given 2 cameras and their respective poses
  - $l' = Fx$
- Modern CPUs feature linear cache pre-fetch behavior
- Rectification can be either planar or polar
- The polar rectification technique described in [9] is used

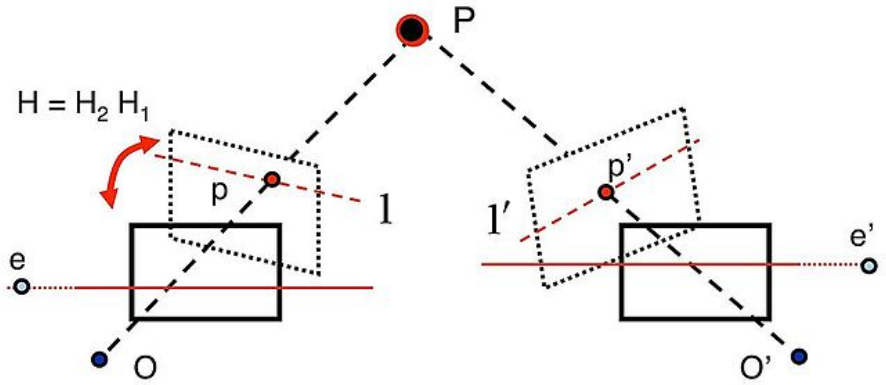


Fig. 4. Image rectification [10]

Stereo Rectification

Additional processing steps were implemented so that corresponding pixels in rectified images lie in a fixed disparity range:

- Estimating image-flip
  - Using dot product
  - Avoid breaking the requirement of $x' < x$

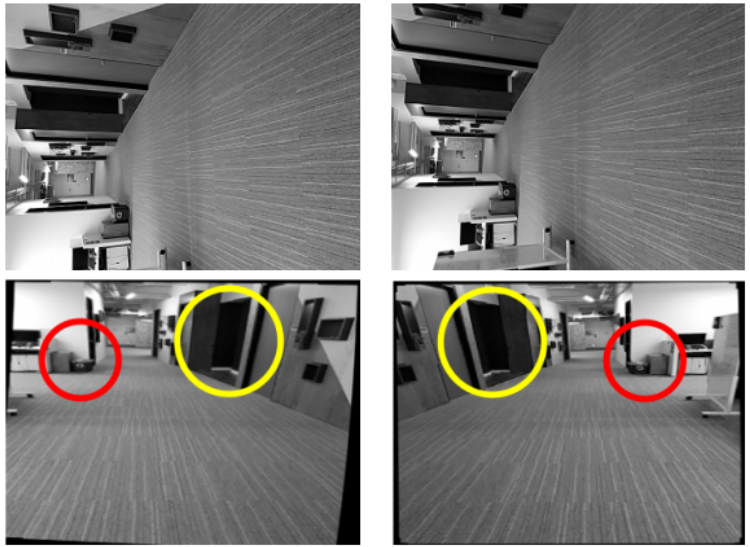


Fig. 5. Image flip [2]

Stereo Rectification

Additional processing steps were implemented so that corresponding pixels in rectified images lie in a fixed disparity range:

- Estimating image-flip
  - Using dot product
  - Avoid breaking the requirement of $x' < x$
- Estimating image-swap and x-shift
  - Estimate disparity ranges to predict which image is used as reference
  - Reference image and required amount of horizontal shift selected

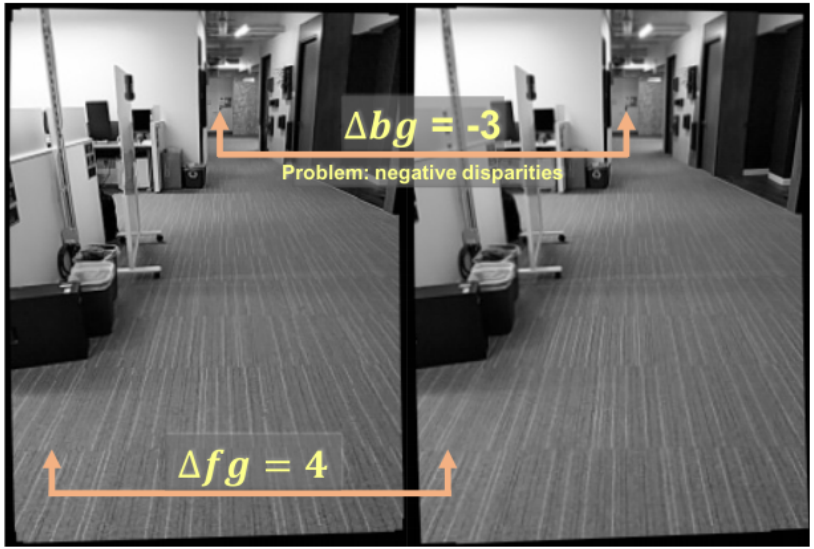


Fig. 6. Horizontal shift [2]

Stereo Rectification

Additional processing steps were implemented so that corresponding pixels in rectified images lie in a fixed disparity range:

- Estimating image-flip
  - Using dot product
  - Avoid breaking the requirement of $x' < x$
- Estimating image-swap and x-shift
  - Estimate disparity ranges to predict which image is used as reference
  - Reference image and required amount of horizontal shift selected
- Improving horizontal resolution for higher quality stereo-matching
  - Standard techniques have a similar sub-pixel accuracy of 0.2 pixels
  - Rectified images can be bigger than the original ones
  - Re-size so that disparity range matches maximum expected number of disparities
- Accounting for some pose uncertainty
  - Setting minimum expected number of disparities between [5, 10]
  - Disk of 20 pixels around the epipole is invalidated

Stereo-Matching

The goal in this section of the pipeline is to generate a **sparse depth map**. For this task the pairwise-CRF is used:

$$E(Y|D) = \sum_i \psi_u(y_i = l_i) + \sum_i \sum_{j \in N_i} \psi_p(y_i = l_i, y_j = l_j)$$

where ψ_u measures the likelihood that two pixels are in correspondence and ψ_p acts as a regularizer that encourages piece-wise smooth solutions.

We optimize this cost function using a hybrid of PatchMatch [11] and HashMatch [12]

- The basic idea of CRF inference using PatchMatch [11] is to propagate good disparity labels

Stereo-Matching

PatchMatch [11] consists of 3 steps:

- Random initialization of the disparity map
- Propagation of the disparity labels
  - $\text{argmin}_{f(x,y)}\{D(f(x,y)), D(f(x-1,y)), D(f(x,y-1))\}$

    where $f(x,y)$ is the disparity value for the pixel (x,y) and D(f(x, y)) is the error between the patch at pixel (x,y) in A and the pixel $(x,y)+f(x,y)$ in B
- Random search

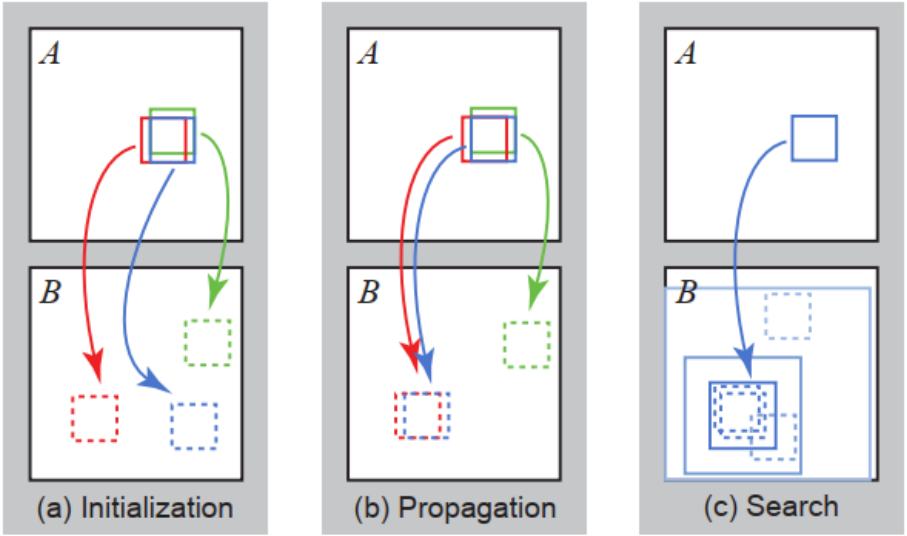


Fig. 7. PatchMatch steps [11]

Stereo-Matching

The authors introduced a propagation strategy that is tailored for modern CPUs

- Used the fact that a vector register of pixels represents a subset of a particular image row
- Achieved paralelism, and hence, improved performance

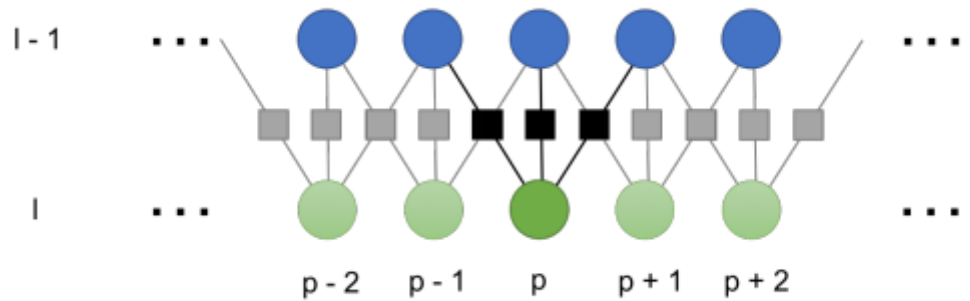


Fig. 8. Independence of horizontal line pixels [2]

Stereo-Matching

Then, most of the unstable predictions are invalidated:

- CRF-cost thresholding
- Decision tree approximation of connected component analysis

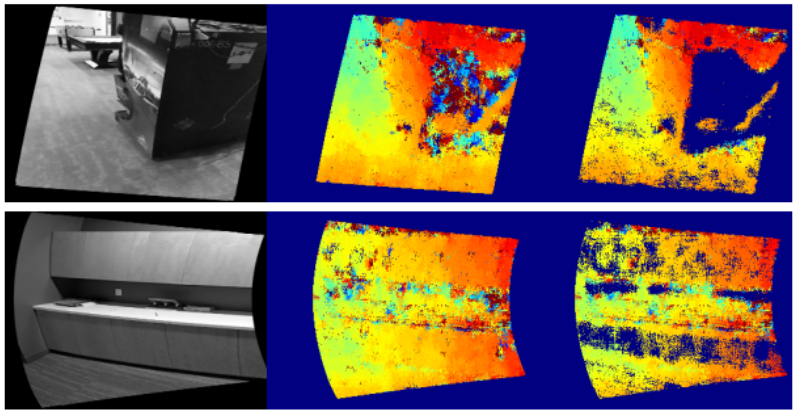


Fig. 9. CRF-cost invalidation in disparity space [2]

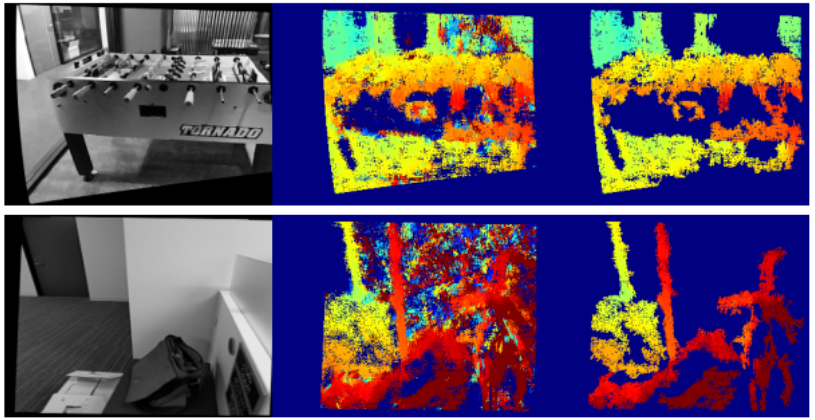


Fig. 10. Connected component invalidation in disparity space [2]

Stereo-Matching

Then, most of the unstable predictions are invalidated:

- CRF-cost thresholding
- Decision tree approximation of connected component analysis

The final step in this section is the computation of depth values using the obtained sparse disparity map

- Solving the linear problem described in [13] which is not optimal but fast to solve

Bilateral Solver Extensions

At this point the following step is the densification of the depth map. Here the **bilateral solver [14]** is applied, since it encourages smoothness while maintaining fidelity with respect to some observation (the sparse depth map in this case).

This technique formulates the following optimization problem:

$$\underset{x}{\text{minimize}} \frac{\lambda}{2} \sum_{i,j} \hat{W}_{i,j}(x_i - x_j)^2 + \sum_i c_i(x_i - t_i)^2$$

where **t** is a "target" image (the noisy sparse depth map), **c** is a "confidence" image (the inverse of the invalidation mask), **x** is the recovered "output" image retrieved by the solver, λ is the smoothness parameter, and **W** the bilateral affinity matrix defined as:

$$W_{i,j} = exp\left(-\frac{\left(p_i^x - p_j^x\right)^2 + \left(p_i^y - p_j^y\right)^2}{2\sigma_{xy}^2} - \frac{(r_i - r_j)^2}{2\sigma_r^2}\right)$$

where each weight is computed based on a "reference" image **r** (the grayscale image from the camera)

Bilateral Solver Extensions

It is important to understand the effect of the bilateral affinity matrix **W**

$$W_{i,j} = exp\left(-\frac{\left(p_i^x - p_j^x\right)^2 + \left(p_i^y - p_j^y\right)^2}{2\sigma_{xy}^2} - \frac{(r_i - r_j)^2}{2\sigma_r^2}\right)$$

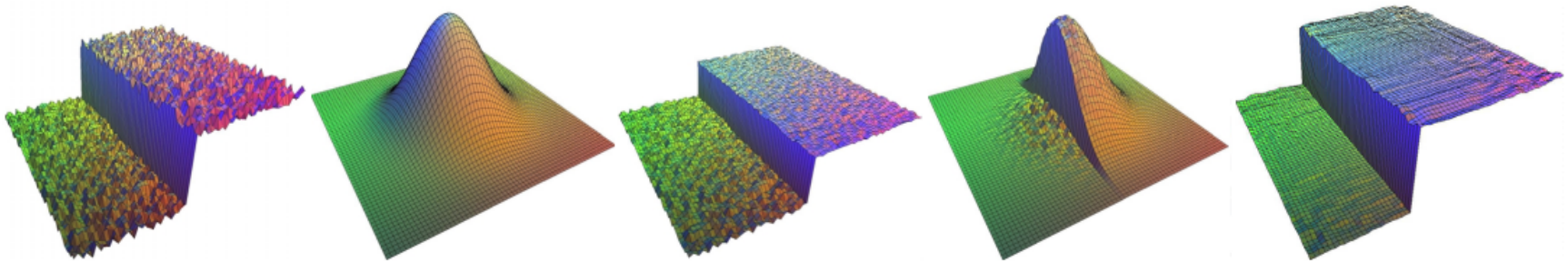


Fig. 11. Bilateral filter [15]

Bilateral Solver Extensions

This problem is extremely expensive:

- Can be taken from the **pixel-space** to the **bilateral-space**
- Significant reduction in the number of variables

This is accomplished by using the bilateral grid representation of the image derived from the following decomposition of $\hat{W}$:

$$\hat{W} = S^T \bar{B} S$$

Now the relation between spaces can be expressed as follows:

$$x = S^T y$$

with x the pixel values and y the values for each bilateral grid vertex.

Bilateral Solver Extensions

The following image is included for a better understanding of this problem domain transformation where the **slice operation** corresponds to the multiplication by S^T

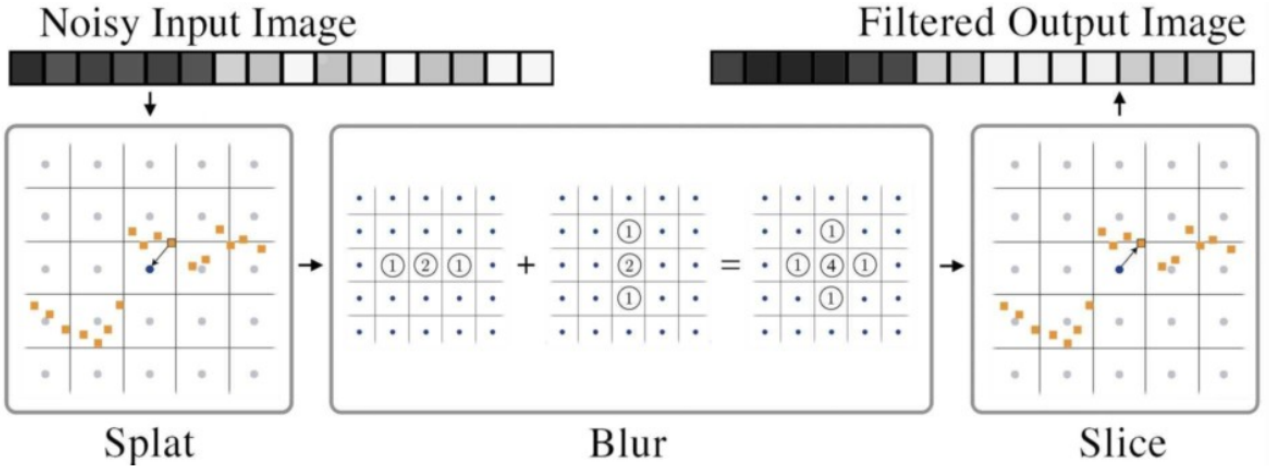


Fig. 12. Bilateral grid [16]

Bilateral Solver Extensions

The bilateral solver minimizes the squared distances between pixels that are bilaterally close

- Strong bias towards fronto-parallel depth maps

The authors deviced the **planar** bilateral solver

- Embedding the bilateral solver in a per-pixel plane-fitting algorithm
- Not confounded by foreshortened surfaces
  - Slanted planes

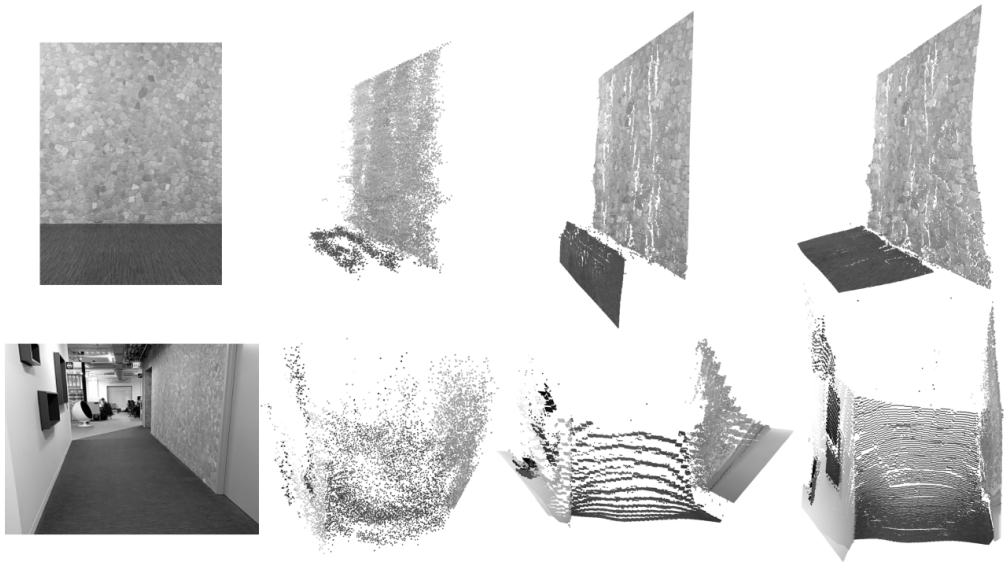


Fig. 13. Bilateral solver vs. Planar bilateral solver [2]

Bilateral Solver Extensions

Further improvements of the bilateral solver were introduced:

- **Late-stage slicing** for extremely low-latency edge-aware depth estimates
- Exponential moving average of the bilateral grid of depths for temporally consistent solutions
- Warm start of the gradient descent using previous frame's solution

Fig. 14. Late-stage slicing [2]

Bilateral Solver Extensions

Further improvements of the bilateral solver were introduced:

- Late-stage slicing for extremely low-latency edge-aware depth estimates
- Exponential moving average of the bilateral grid of depths for **temporally consistent** solutions
- Warm start of the gradient descent using previous frame's solution

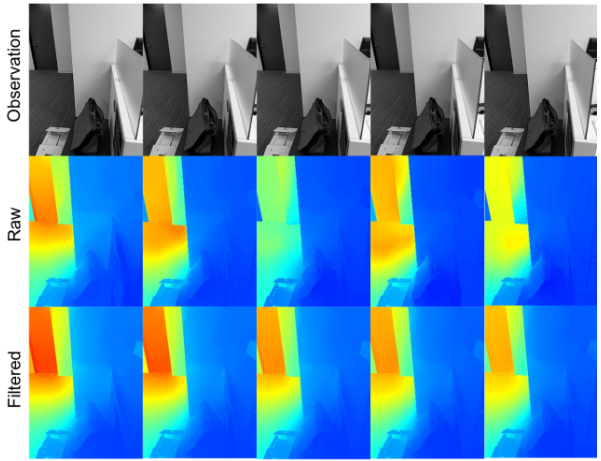


Fig. 15. Temporal filtering [2]

Outline

- Introduction
- Overview
- Method Description
- Experiments and Results
- Personal Comments
- Summary

Experiments and Results

The overall performance of the stereo matching approach was tested on the Middlebury Stereo Dataset V3 [17] and quantitatively compared to other alogrithms:

- Intel R200
- Local expansion moves introduced by [18]
  - Offline technique

	Austr	AustrP	Bicyc2	Class	ClassE	Compu	Crusa	CrusaP	Djemb	DjembL	Hoops	Livgrm	Nkuba	Plants	Stairs	Average
Intel R200	70.5	14.4	21.3	37.7	72.2	38.1	53.2	31.4	18.3	52.4	52.6	44.1	45.4	50.7	66.5	40.9
LocalExp	3.65	2.87	2.98	1.99	5.59	3.37	3.48	3.35	2.05	10.3	9.75	8.57	14.4	5.4	9.55	5.43
Ours	67.6	25.0	29.2	40.9	57.3	35.5	57.5	40.4	19.9	42.8	52.6	39.8	37.1	51.7	34.9	40.4

Table 1. Bad 2.0 error on the dense benchmark[2]

Experiments and Results

The introduced approach was also qualitatively compared to:

- **Deep Learning approaches**
  - End-to-end learning of geometry and context for deep stereo regression, by Kendall et al. [19]
  - Unsupervised monocular depth estimation with left-right consistency, by Godard et al. [20]
- iPhoneX
  - Stereo pair
  - Powerful computational resources

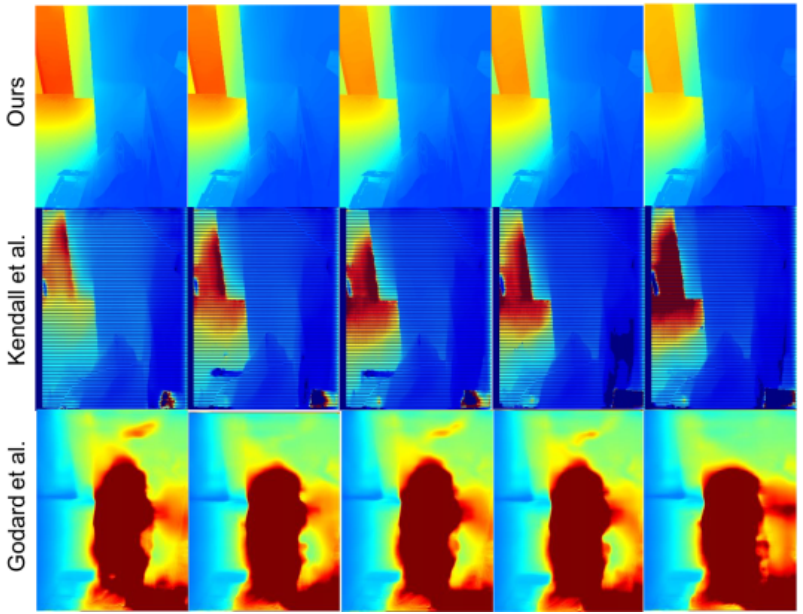


Fig. 16. Comparison to deep neural networks [2]

Experiments and Results

The introduced approach was also qualitatively compared to:

- Deep Learning approaches
  - End-to-end learning of geometry and context for deep stereo regression, by Kendall et al. [19]
  - Unsupervised monocular depth estimation with left-right consistency, by Godard et al. [20]
- **iPhoneX**
  - Stereo pair
  - Powerful computational resources

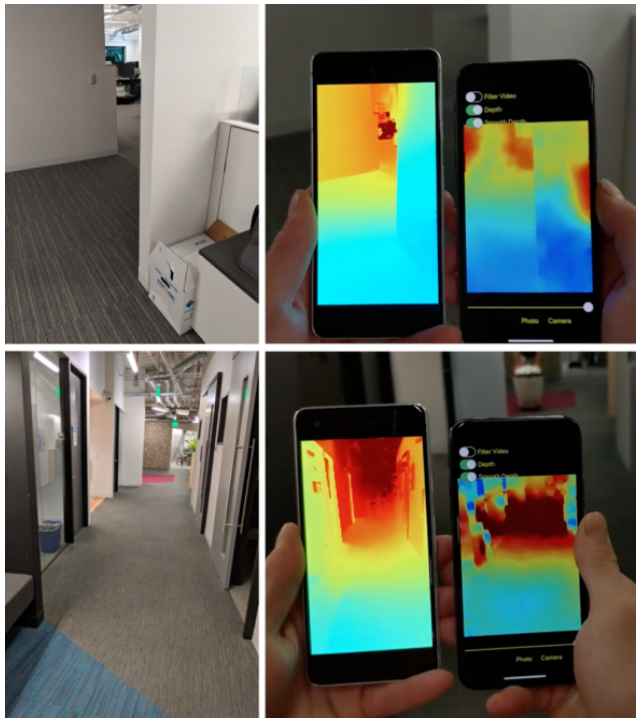


Fig. 17. Pixel 2 (on the left) compared to iPhoneX (on the right) [2]

Outline

- Introduction
- Overview
- Method Description
- Experiments and Results
- Personal Comments
- Summary

Outline

- Introduction
- Overview
- Method Description
- Experiments and Results
- Personal Comments
- Summary

Summary

This works succesfully combines and extends several techniques in order to achieve low-latency edge-aware depth maps on mid-range to high-end phones, which conveys some limitations:

- Monocular setting
- Limited computational resources
- Free motion of the user

Technical contributions of the novel depth from motion pipeline:

- Use of polar rectified images for efficient stereo matching
- New keyframe selection strategy
- Highly optimized stereo matching algorithm: PatchMatch[11] + HashMatch[12]
- New extensions of the bilateral solver for depth post-processing
  - Higher quality point clouds by introduction of plane-fitting
  - Warm initialization
  - Temporal filtering
- New late stage rendering step that provides a fluid low-latency experience to users

Thanks for your attention!

Do you have any questions?

References I

[1] G. A. VR. (2019), "Get ready for the arcore depth api," Youtube, [Online]. Available: https://www.youtube.com/watch?v=1q0-jdknbTs.

[2] J. Valentin, A. Kowdle, J. T. Barron, N. Wadhwa, M. Dzitsiuk, M. Schoenberg, V. Verma, A. Csaszar, E. Turner, I. Dryanovski, *et al.*, "Depth from motion for smartphone ar," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–19, 2018.

[3] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3d reconstruction and 6-dof tracking with an event camera," in *European Conference on Computer Vision*, Springer, 2016, pp. 349–364.

[4] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "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.

[5] K. Karsch, C. Liu, and S. B. Kang, "Depth transfer: Depth extraction from video using non-parametric sampling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2144–2158, 2014.

[6] P. Li, D. Farin, R. K. Gunnewiek, *et al.*, "On creating depth maps from monoscopic video using structure from motion," in *Proc. of IEEE Workshop on Content Generation and Coding for 3D-television*, 2006, pp. 508–515.

[7] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision*, Springer, 2016, pp. 501–518.

References II

[8] E. Zheng, E. Dunn, V. Jojic, and J.-M. Frahm, "Patchmatch based joint view selection and depthmap estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1510–1517.

[9] M. Pollefeys, R. Koch, and L. Van Gool, "A simple and efficient rectification method for general motion," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, IEEE, vol. 1, 1999, pp. 496–501.

[10] W. contributors. (2020), "Image rectification," [Online]. Available: `https://en.wikipedia.org/wiki/Image_rectification`.

[11] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[12] S. R. Fanello, J. Valentin, A. Kowdle, C. Rhemann, V. Tankovich, C. Ciliberto, P. Davidson, and S. Izadi, "Low compute and fully parallel computer vision with hashmatch," in *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2017, pp. 3894–3903.

[13] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[14] J. T. Barron and B. Poole, "The fast bilateral solver," in *European Conference on Computer Vision*, Springer, 2016, pp. 617–632.

[15] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 2002, pp. 257–266.

References III

[16] J. T. Barron, A. Adams, Y. Shih, and C. Hernández, "Fast bilateral-space stereo for synthetic defocus," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4466–4474.

[17] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *German conference on pattern recognition*, Springer, 2014, pp. 31–42.

[18] T. Taniai, Y. Matsushita, Y. Sato, and T. Naemura, "Continuous 3d label stereo matching using local expansion moves," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 11, pp. 2725–2739, 2017.

[19] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 66–75.

[20] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 270–279.