Master Thesis Presentation

Comparison of Disparity Algorithms for Stereoscopic Videos

Ben John

May 25, 2016

University of Mannheim, Department of Praktische Informatik IV

Table of contents

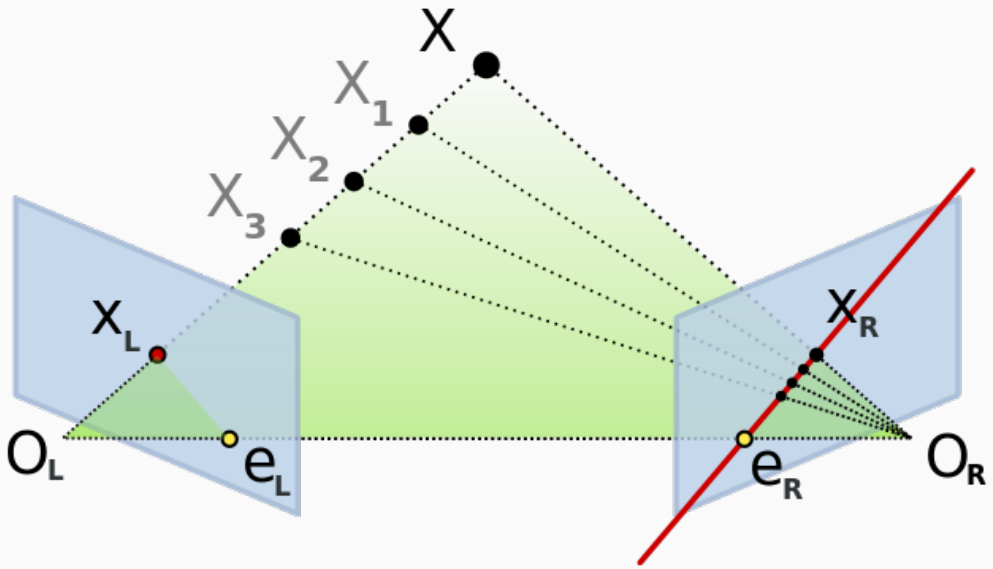
Motivation

Applications

- Depth-estimation via camera settings
- Kinect (sunlight)
- 3DTV (remapping)

Foundations

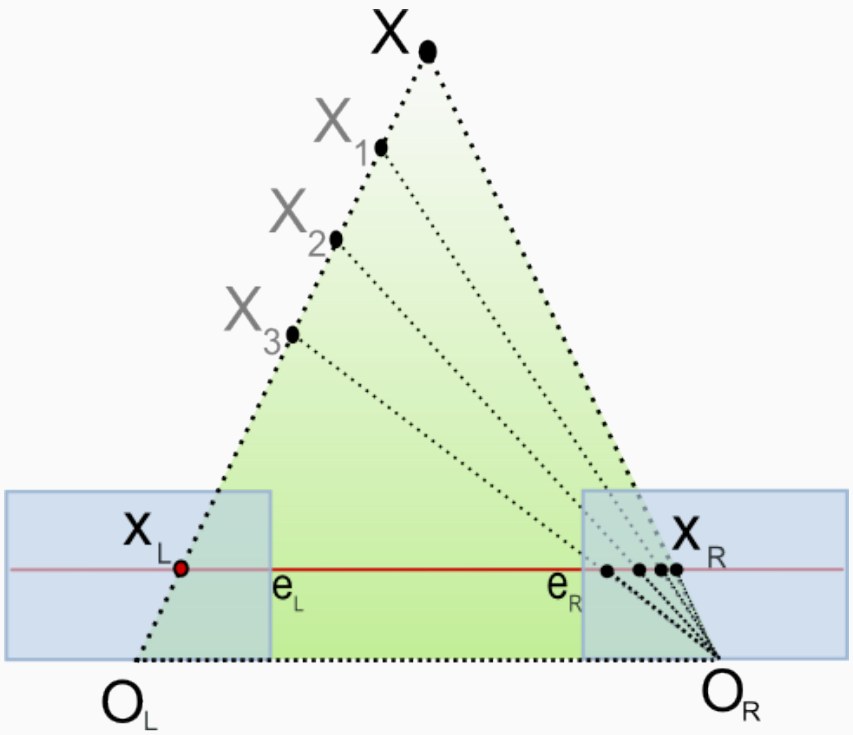
Epipolar geometry



Epipolar geometry[1]

[1]Source (accessed 02/2016): `https://en.wikipedia.org`.

Epipolar geometry

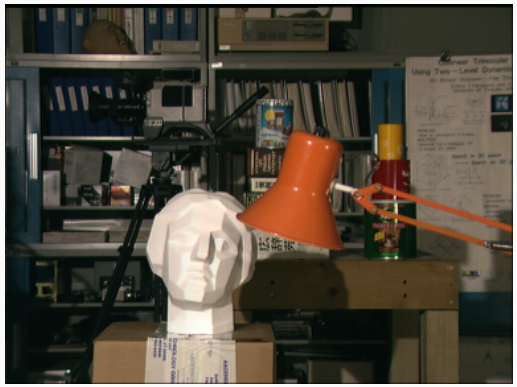


Epipolar geometry after image rectification[2]

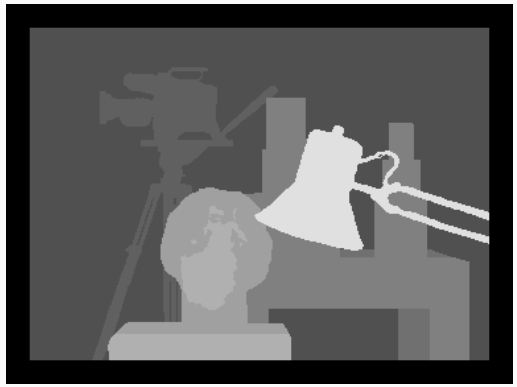
[2]Source (accessed 02/2016): `https://en.wikipedia.org`.

Example for stereo image pair

(a) left input image



(b) right input image



(c) ground-truth data

Tsukuba benchmark stereo image pair of the University of Tsukuba.

Classification

- Local methods
  - Area matching
  - Feature matching
- Global methods
  - Dynamic programming
  - Graph cuts
  - Belief propagation

Processing steps

1. Compute of matching cost
2. Save values in disparity space image
3. Aggregate of cost values
4. Disparity refinement

Step 1: Matching cost

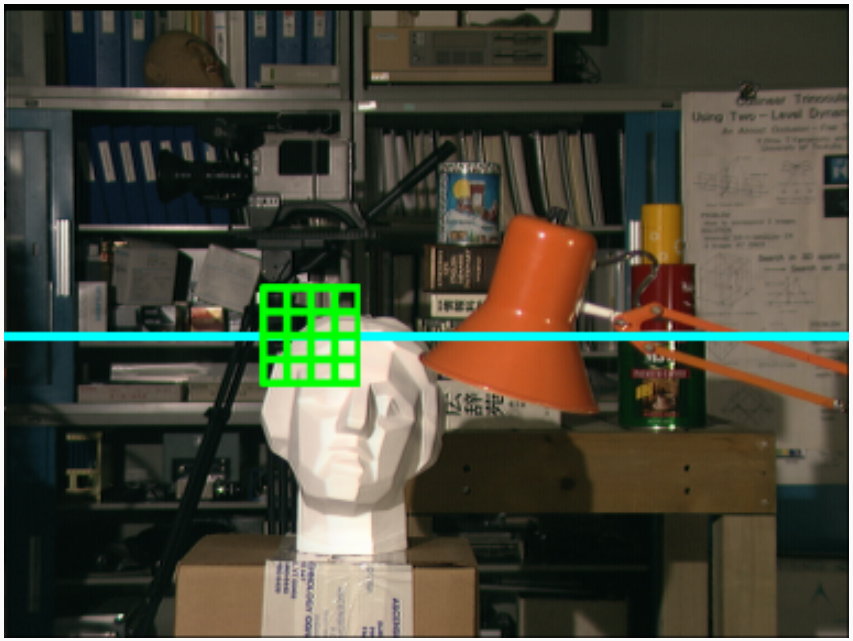
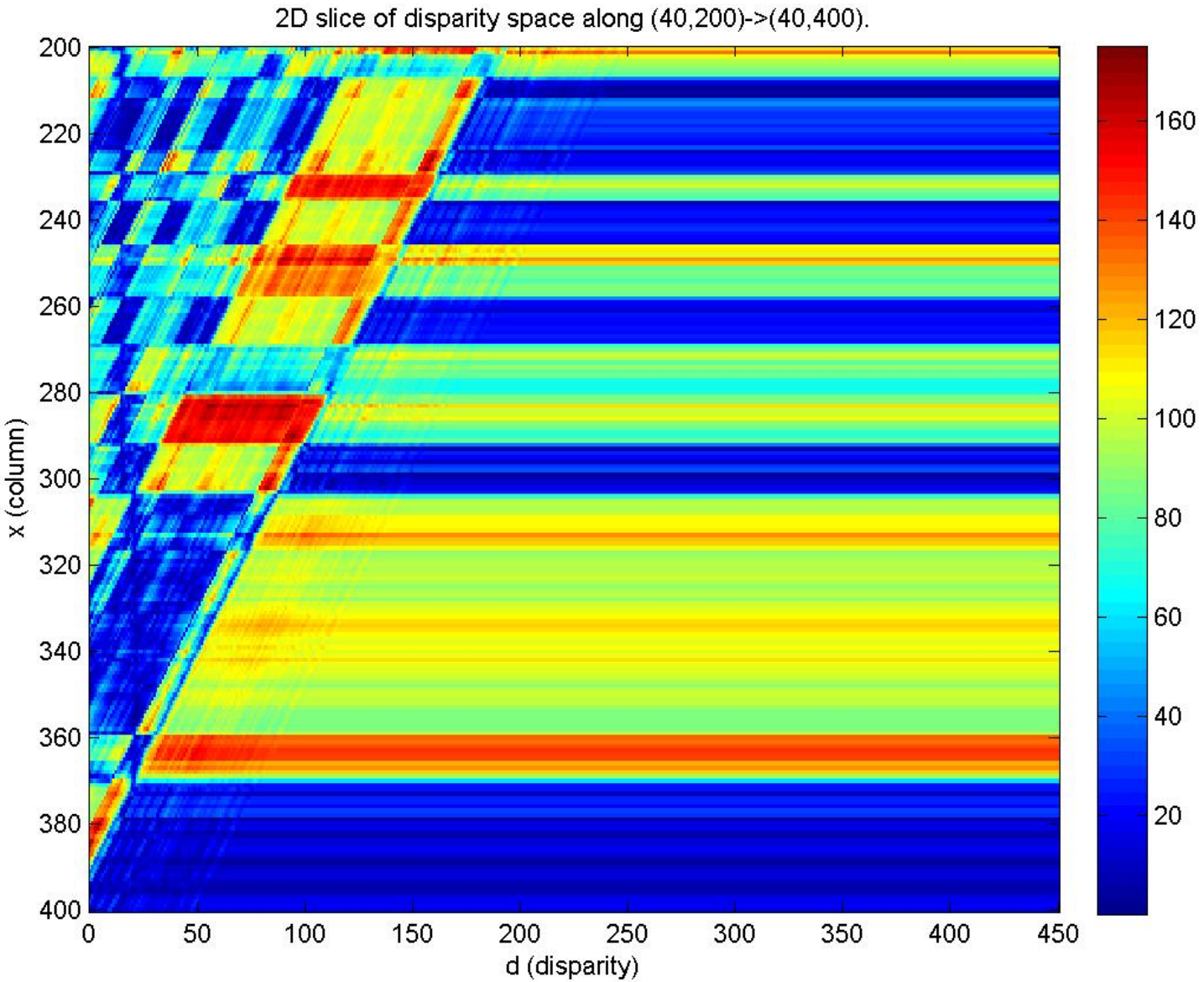


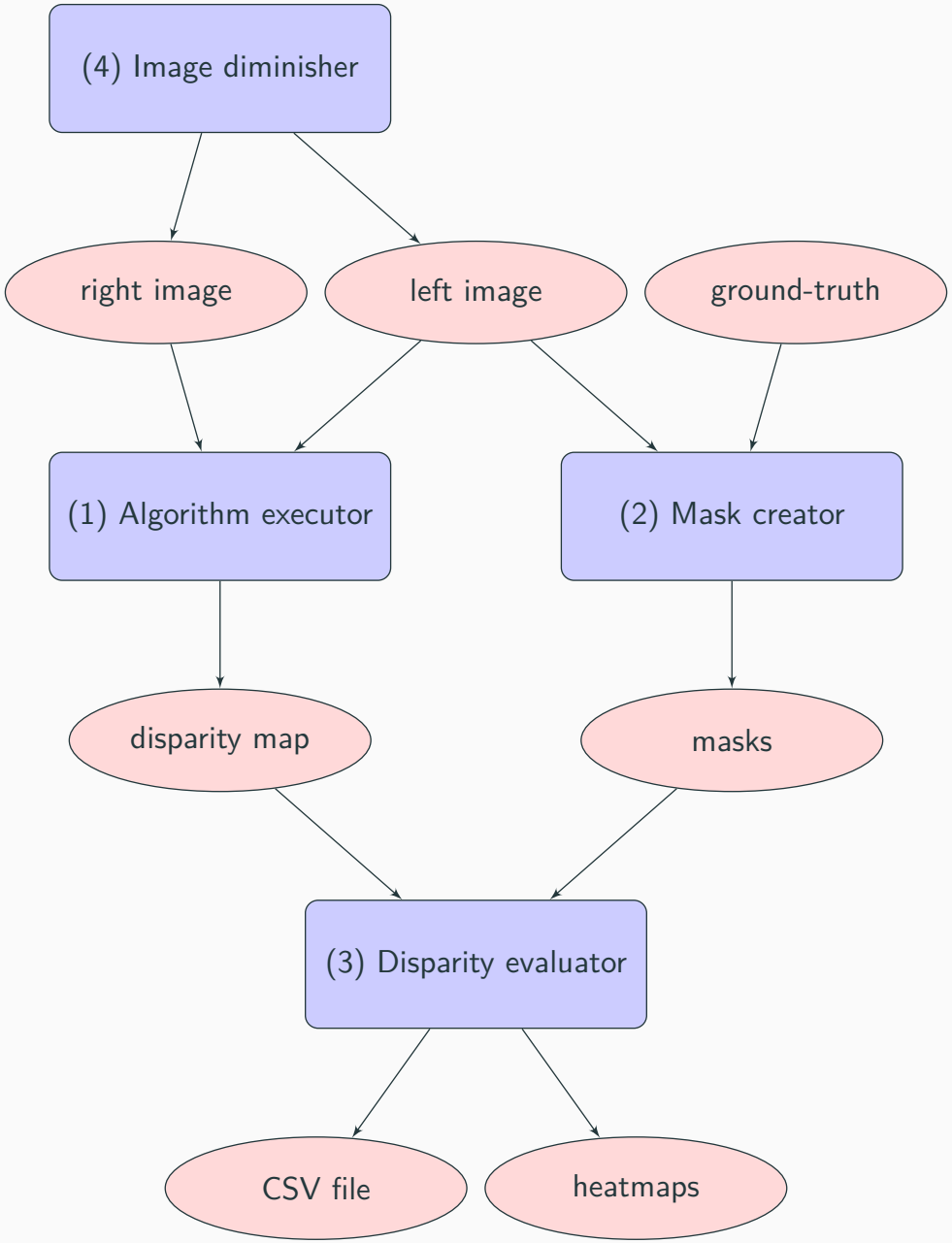
Illustration of block matching along a scanline.

Step 2: Disparity space image



Implementation

Overview


(4) Image diminisher
right image
left image
ground-truth
(1) Algorithm executor
(2) Mask creator
disparity map
masks
(3) Disparity evaluator
CSV file
heatmaps

Spatiotemporal stereo matcher (1)

Algorithm 1: CreateDisparitySpaceImage

Input: I_L, I_R, d_{max}, $wSize$

Output: C

```
1  step ← (wSize − 1)/2
2  C ← CreateMatrix(Cols(I_L), Rows(I_L), d_max)
3  for t ← 0 to Images(I_L) do
4      leftImage ← I_L(t)
5      rightImage ← I_L(t)
6      for y ← 0 + step to Rows(I_L(0)) − step do
7          for x ← 0 + step to Cols(I_L(0)) − step − d_max do
8              for d ← 0 to d_max do
9                  rect_L ← Rect{x − step, y − step, wSize, wSize}
10                 rect_R ← Rect{x + d − step, y − step, wSize, wSize}
11                 window_L ← leftImage(rect_L)
12                 window_R ← rightImage(rect_R)
13                 C(x, y, t, d) ← MatchingCost(window_L, window_R)
14 return C
```

Spatiotemporal stereo matcher (2)

Algorithm 2: GetDisparityMap

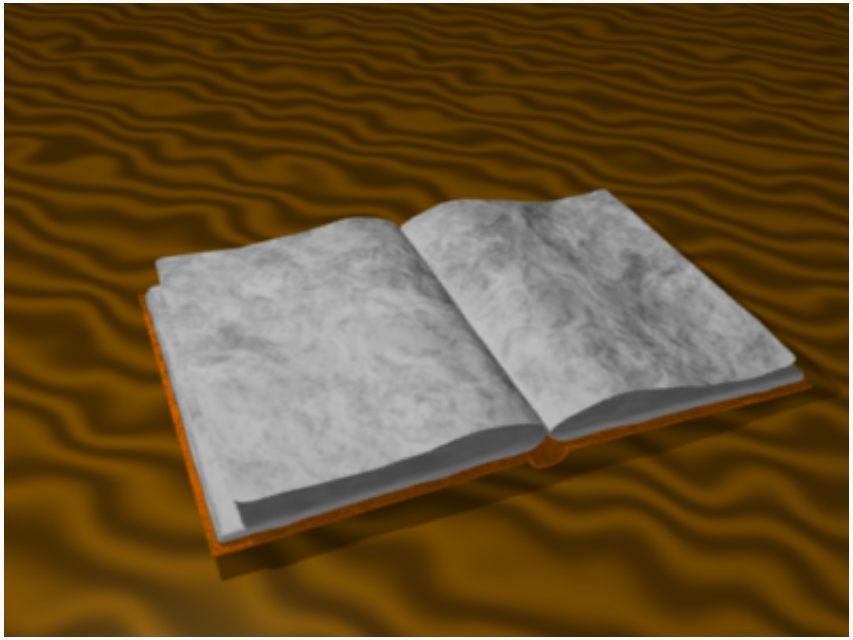
Input: C, t

Output: *DisparityMap*

```
1 DisparityMap ← CreateMatrix(Cols(C), Rows(C))
2 for t ← 0 to Frames(C) do
3     for y ← 0 to Rows(C) do
4         for x ← 0 to Cols(C) do
5             Cost ← (1/4) C(x, y, f0) + (2/4) C(x, y, f1) + (1/4) C(x, y, f2)
6             DisparityMap(x, y) ← BestMatch(Cost)
7 return DisparityMap
```

Masking modes

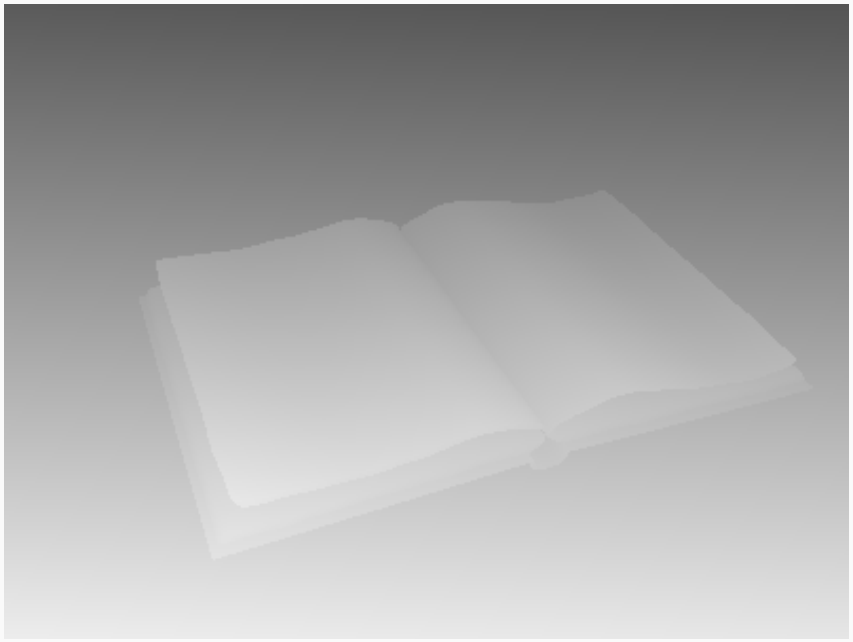
- Non-occluded mask
- Depth-discontinuity mask
- Textureless mask
- Saliency mask



Frame of book sequence

Masking modes

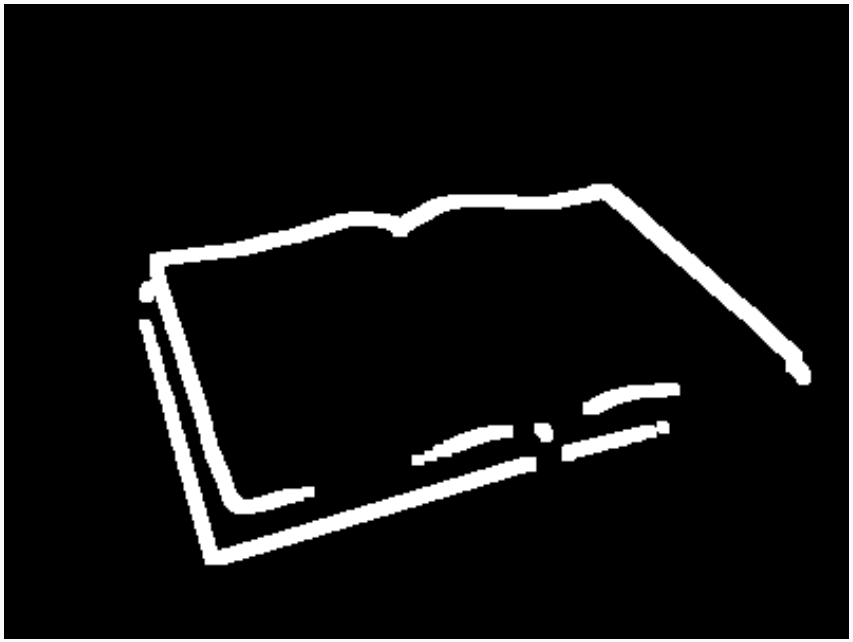
- Non-occluded mask
- Depth-discontinuity mask
- Textureless mask
- Saliency mask



Ground-truth disparity map

Masking modes

- Non-occluded mask
- Depth-discontinuity mask
- Textureless mask
- Saliency mask



Depth-discontinuity at object borders

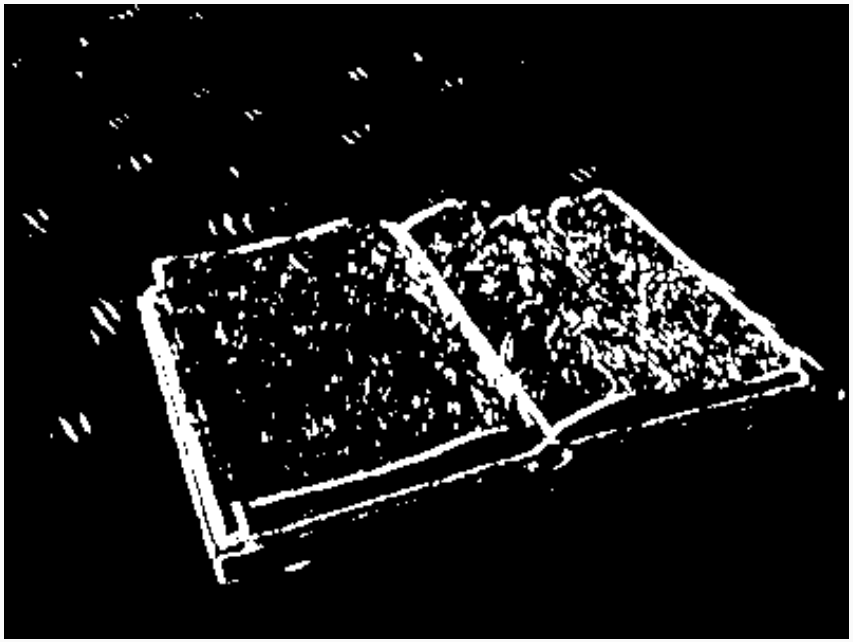
Masking modes

- Non-occluded mask
- Depth-discontinuity mask
- Textureless mask
- Saliency mask

Non-occluded mask

Masking modes

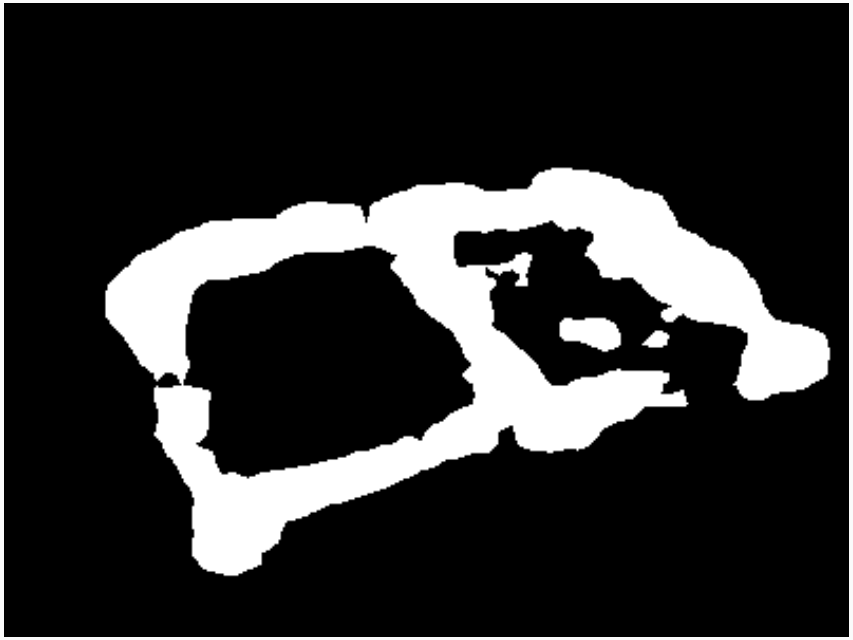
- Non-occluded mask
- Depth-discontinuity mask
- Textureless mask
- Saliency mask



Textureless regions

Masking modes

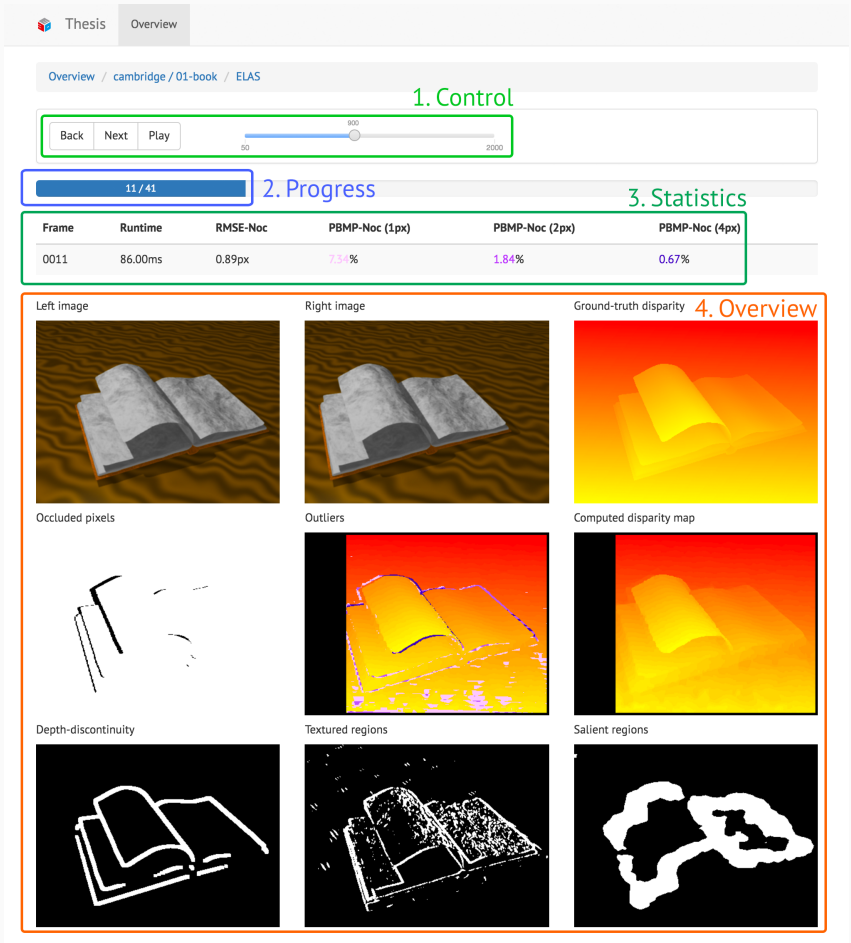
- Non-occluded mask
- Depth-discontinuity mask
- Textureless mask
- Saliency mask



Salient pixels

Web viewer

- Visualization of evaluation engine
- Written in Node.js
- Displaying statistical information



Detail view

Demo

Evaluation

Overview

Analyzed algorithms

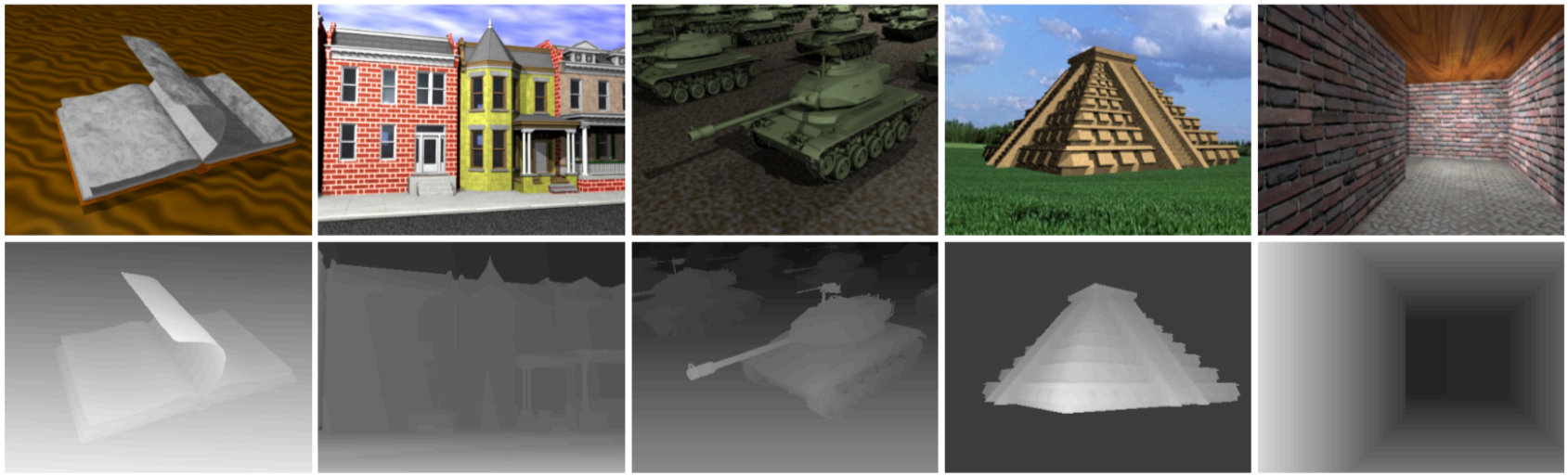
- OpenCV implementations
  - (1) Semi-global stereo matcher
  - (2) Block matcher
  - (10) Simple block matcher
- (3) Efficient large scale stereo matcher (ELAS)
- Middlebury MRF library
  - (4) Iterated conditional modes (ICM)
  - (5-6) Graph cuts (swap and extension)
  - (7-9) Various belief propagations implementations
- Own stereo matcher implementation
  - (11) Simple block-matcher
  - (12-13) Simple spatiotemporal consistent block-matcher

Datasets (1)

- Cambridge
  - 5 sequences
  - at 500 x 400 resolution
- Stereoscopic Video Dataset with precise Depth and Disparity information (SVDDD)
  - 2 sequences
  - at 1920 x 1080 resolution

Datasets (2)

Cambridge



Overview of sequences

Datasets (3)

SVDDD

(a) 02 rabbit scene

(b) 03 apple scene

Overview of sequences

Metrics

- **Percentage of bad matching pixels**

  $\frac{1}{n} \sum_{x,y=0} (|d_a(x,y) - d_e(x,y)| > \delta_t)$

- **RMS-Error**

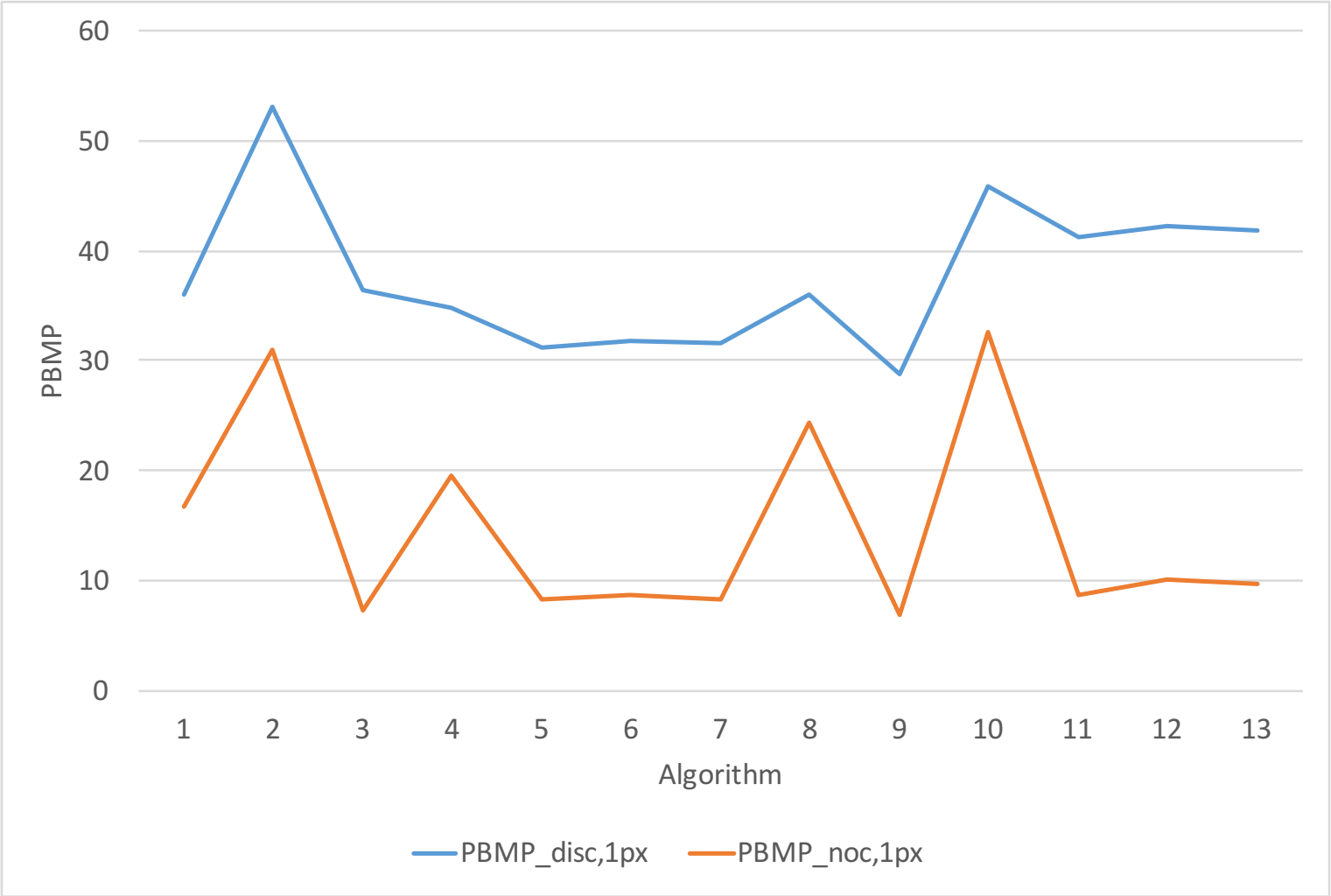
  $\sqrt{\frac{1}{n} \sum_{x,y=0} (d_a(x,y) - d_e(x,y))^2}$

Results (1) - Own stereo matcher

	10 CVSM	11 SNSM	12 SNTU	13 SNTW
S1 - book	32.61%	8.72%	10.07%	9.65%
S2 - street	25.64%	11.79%	8.76%	8.90%
S3 - tanks	13.26%	6.08%	8.71%	7.29%
S4 - temple	38.96%	12.98%	11.15%	11.26%
S5 - tunnel	8.60%	0.93%	4.54%	2.15%
∅	23,81%	8,10%	8,66%	7,85%

Result table for comparison of own implementation with Cambridge dataset

Results (2) - Depth-discontinuity mask



Depth-discontinuity mask applied on the book sequence.

Results (3) - Textureless mask

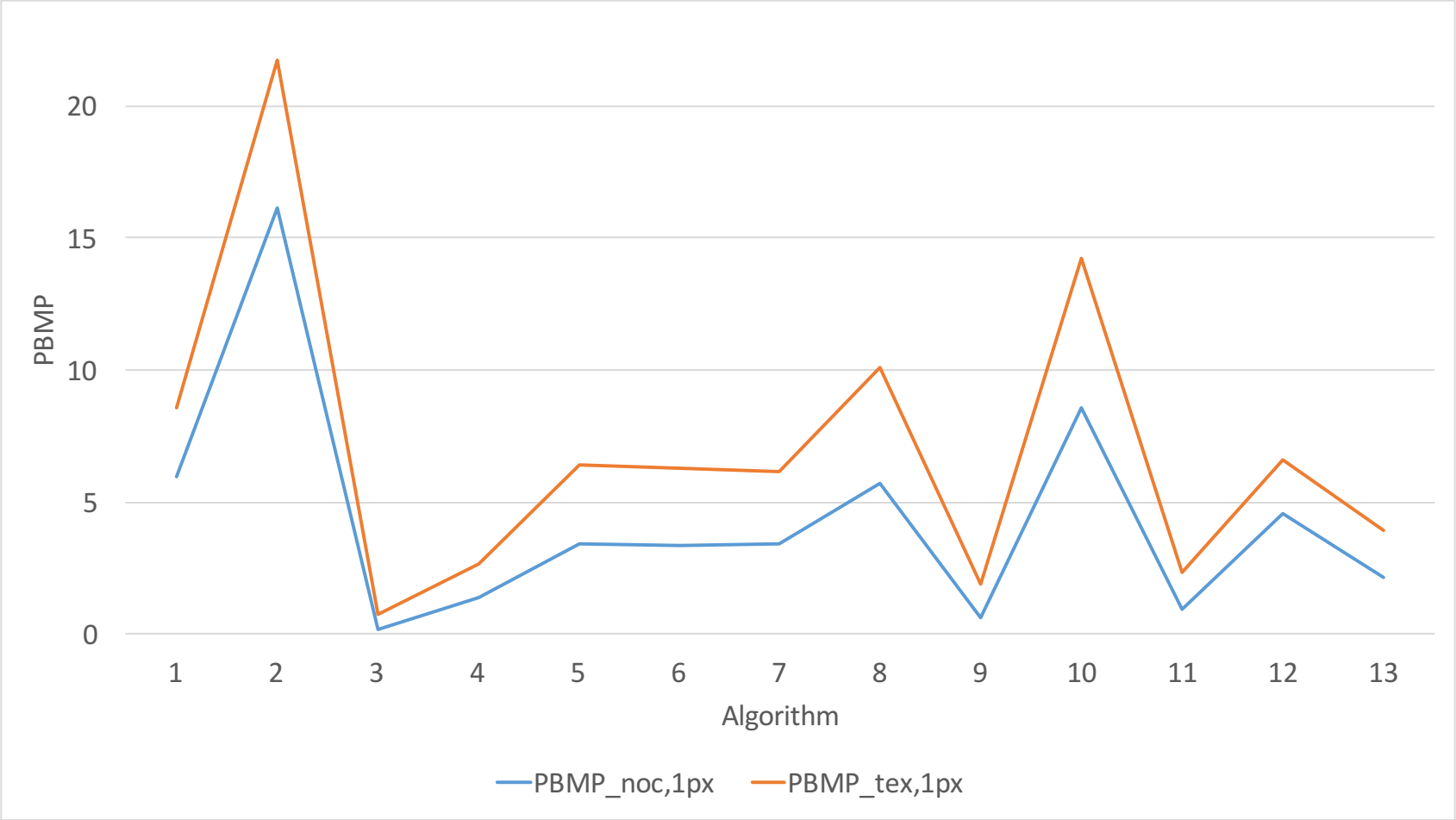


Chart of textureless region mask applied on the tunnel sequence.

Results (4) - Outliers

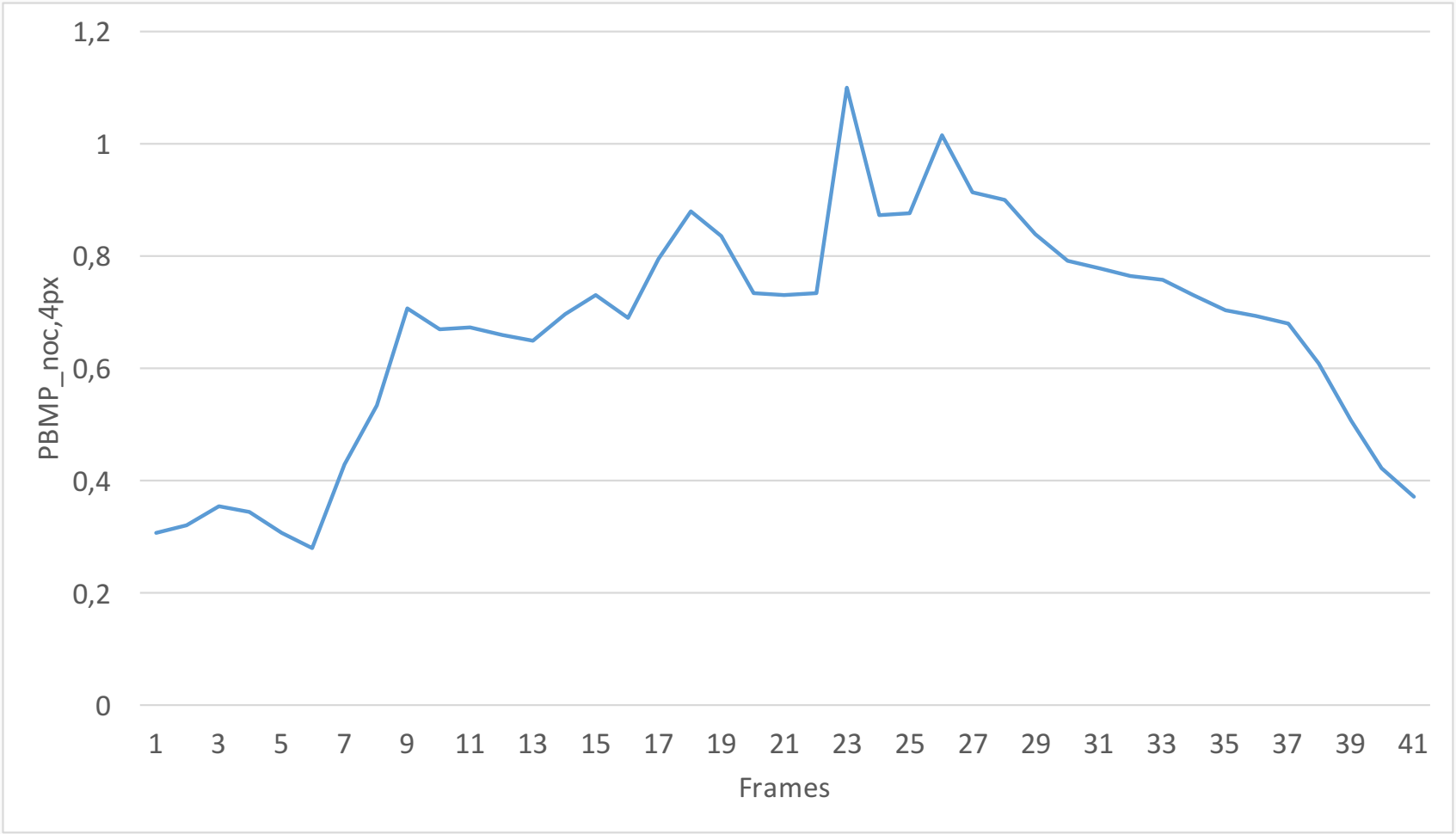
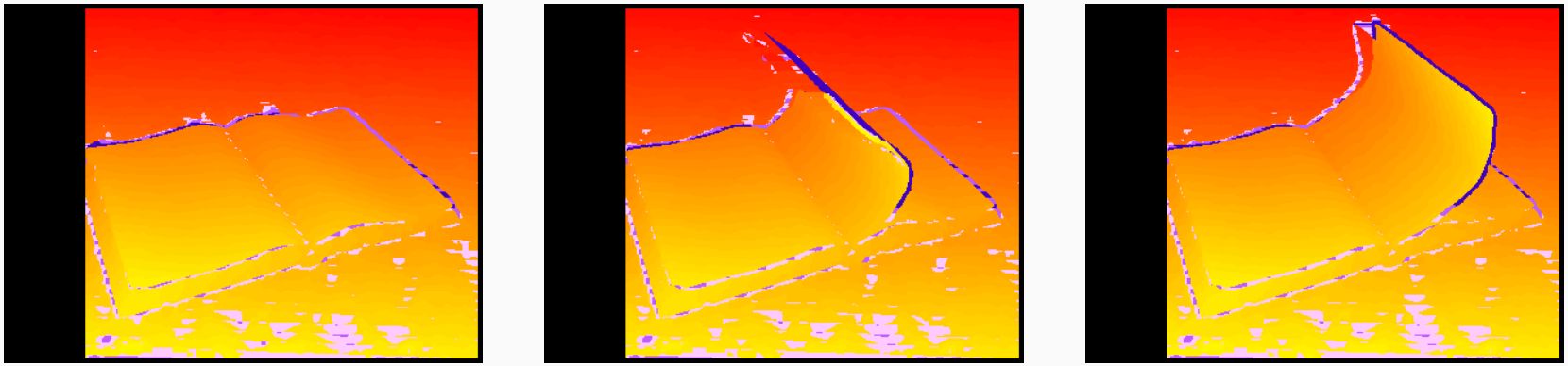


Chart of general outliers in the book sequence.

Results (5) - Outliers example



(a) Frame 1 **(b)** Frame 23 **(c)** Frame 26

Examples for general outliers in the book sequence. The disparity maps are computed with the (3) ELAS algorithm.

Results (6) - Video compression

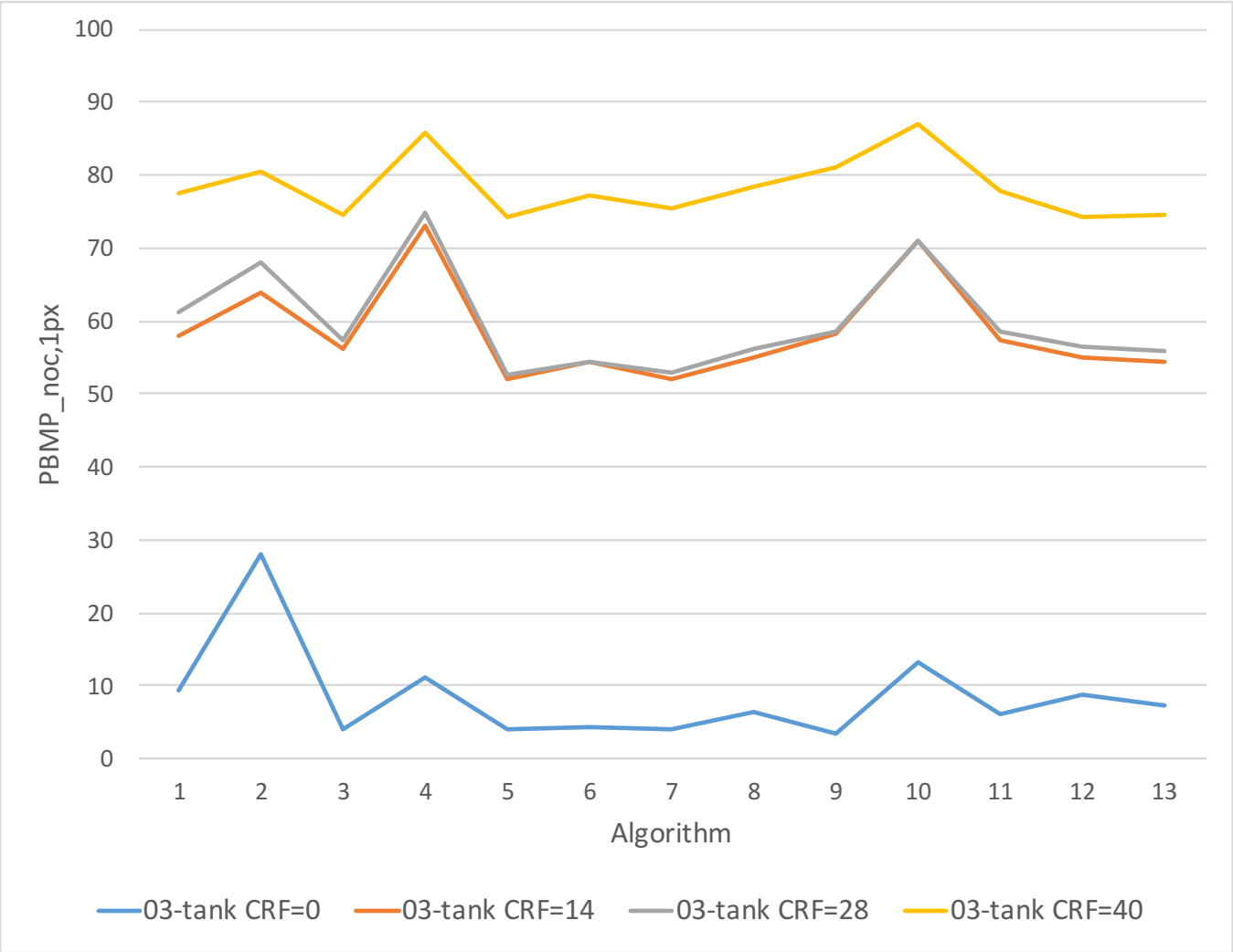
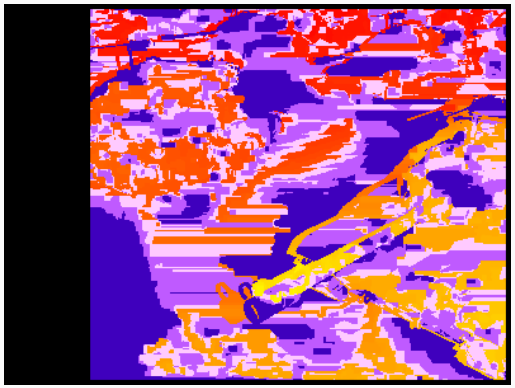
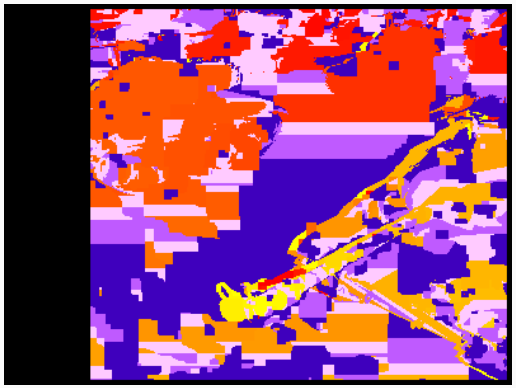


Chart of the impact of different CRF values for H.265 video compression on the result of disparity algorithms focusing on $PBMP_{noc,1px}$.

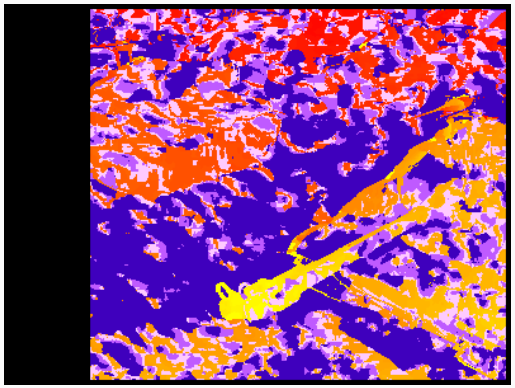
Results (7) - Video compression examples



(a) (3) ELAS outliers



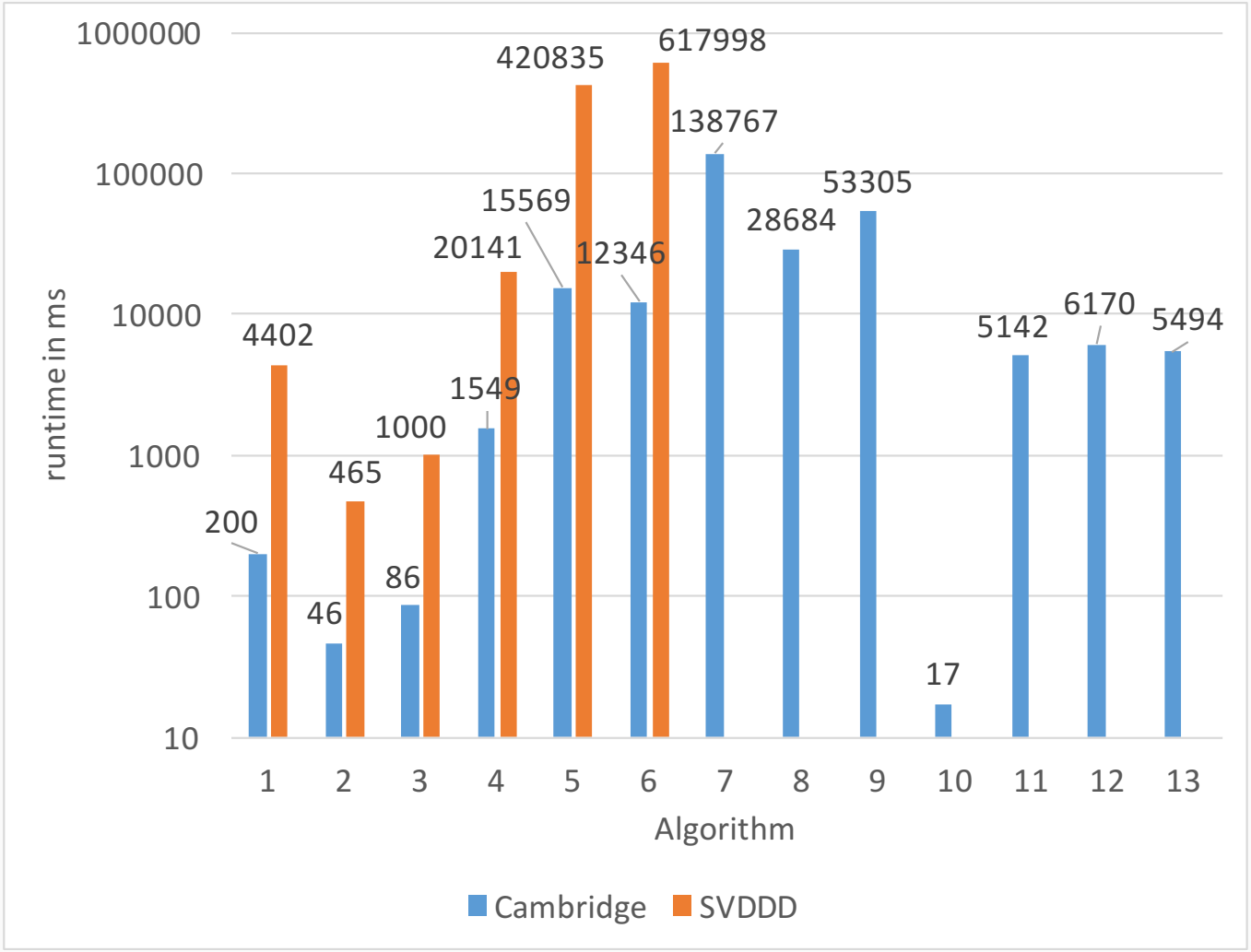
(b) (5) MRF GC Swap outliers



(c) (13) SNSM STW

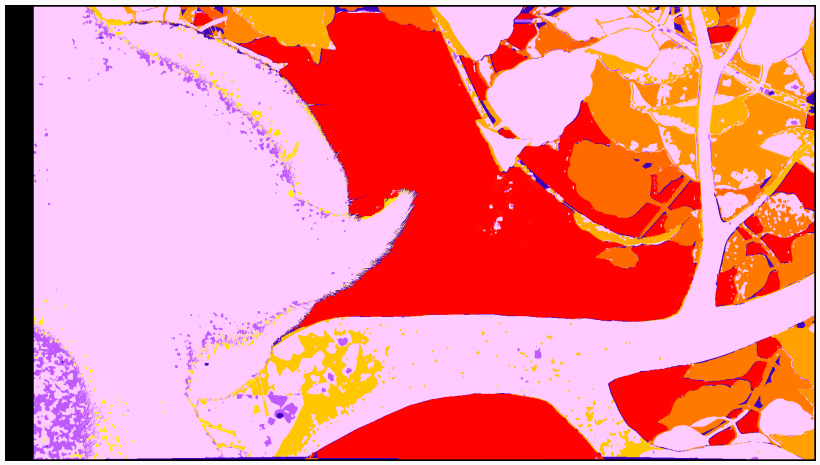
Example of computed disparity maps with video compression. CRF is set to 40. Frame 23 of the tanks scene.

Results (8) - Runtime

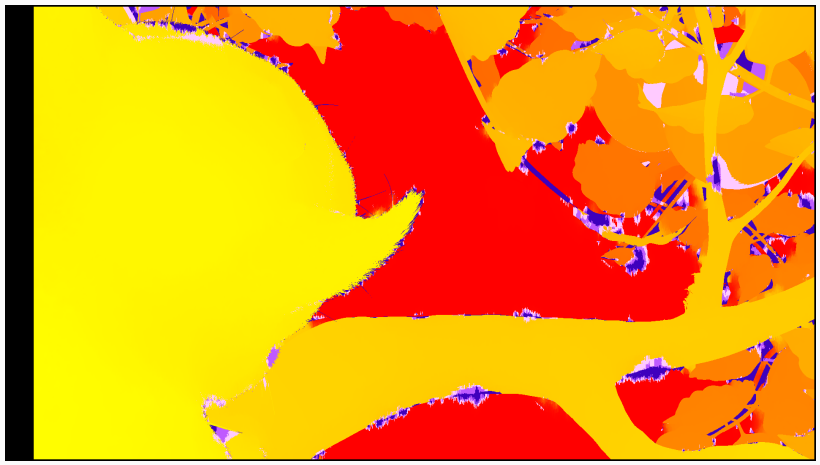


Comparison of the runtime of different disparity algorithms.

Results (9) - SVDDD



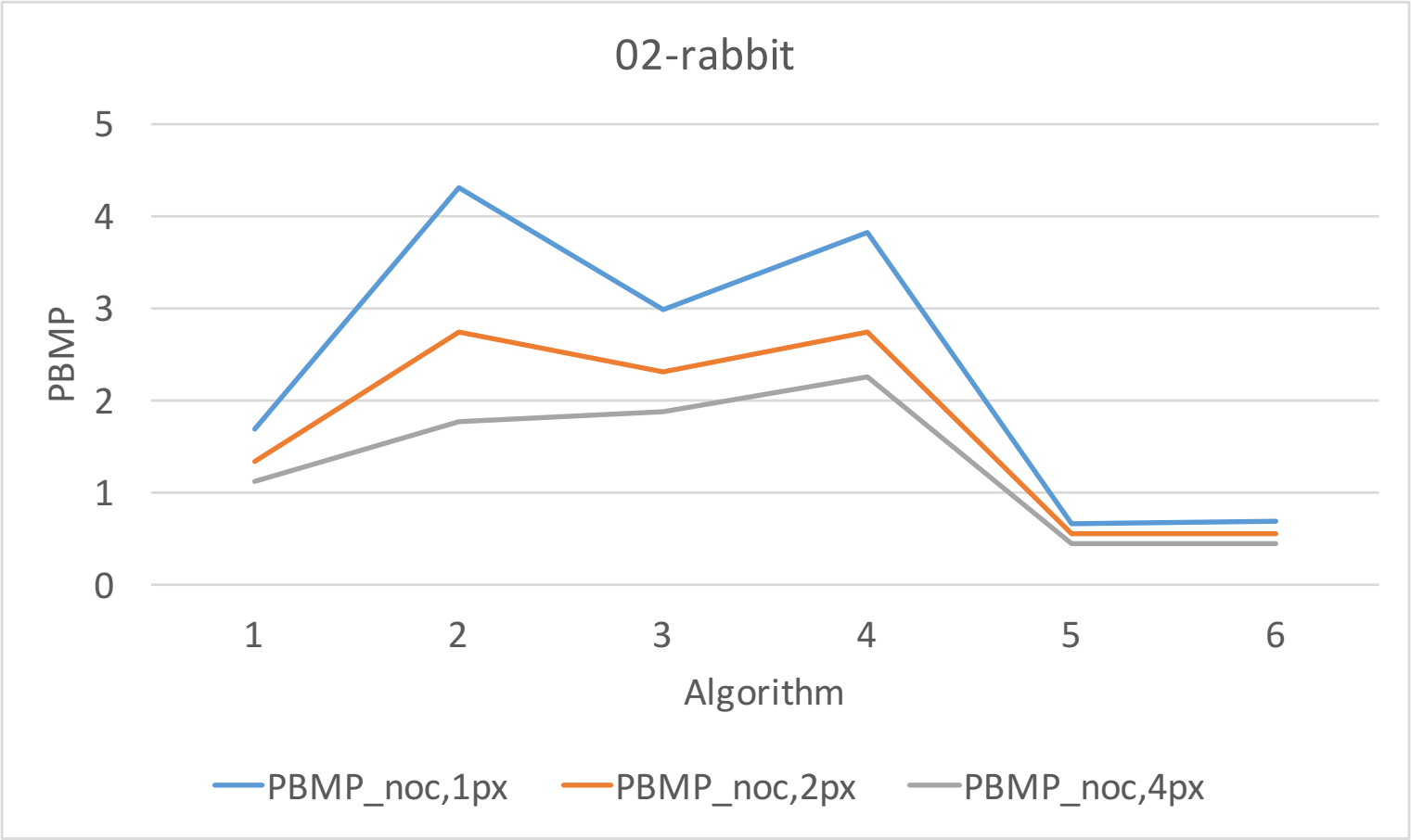
(a) Negative disparity



(b) Only positive disparity

Comparison of computed disparity maps regarding negative disparity.

Results (10) - SVDDD performance



Performance of SVDDD rabbit scene

Results (11) - SVDDD performance

	1	2	3	4	5	6
02-rabbit-neg	58.62%	61.51%	59.99%	60.58%	57.12%	57.13%
02-rabbit	1.68%	4.31%	2.98%	3.82%	0.65%	0.68%
03-apple	1.69%	4.10%	3.11%	3.44%	0.63%	0.65%
∅ (w/o neg)	1.69%	4.21%	3.05%	3.63%	0.64%	0.67%

Result table for general performance of SVDDD ($PBMP_{noc,1px}$)

Conclusion and outlook

Conclusion

- Surprise candidate ELAS
- Camera noise model
- SVDDD dataset
- Salient mask varies a bit
- Immense runtime differences
- Possible outliers in a scene

Contributions

- Generic Disparity Interface
- Evaluation Engine
- Mask creator
- Image diminisher
- Web result viewer
- Benchmark results
- Skeleton for stereo matcher
- Spatiotemporal stereo matcher

Outlook

- Motion saliency
- Enhancement of spatiotemporal matcher
- Holistic evaluation suite for modern disparity algorithm comparison
- Multi-view datasets
- High-resolution datasets
- Optical flow regarding spatiotemporal consistency
- Humans depth experience with neuronal networks

Questions?