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

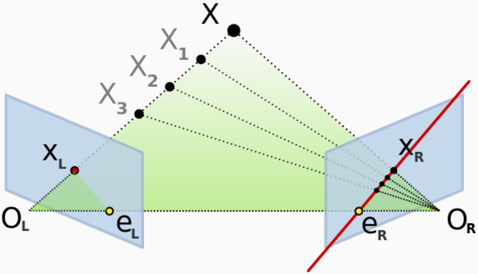
Motivation

Applications

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

Foundations

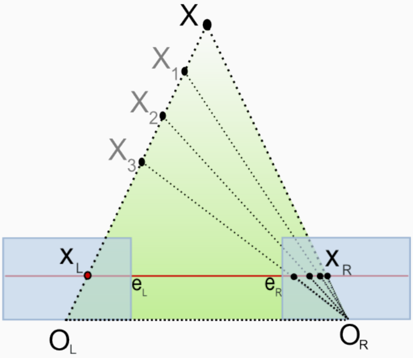
Epipolar geometry



Epipolar geometry[1]

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

Epipolar geometry

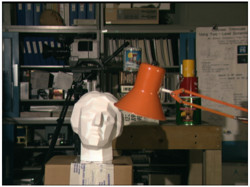


Epipolar geometry after image rectification[2]

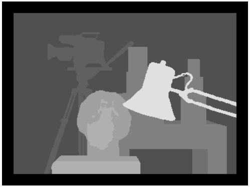
[2] Source (accessed 02/2016): `https://en.wikipedia.org`.

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 [1].

Classification

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

Classification

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

Energy function

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d)$$

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

- Penalty
- Cost for having dissimilarities
- Optimum $= 0$

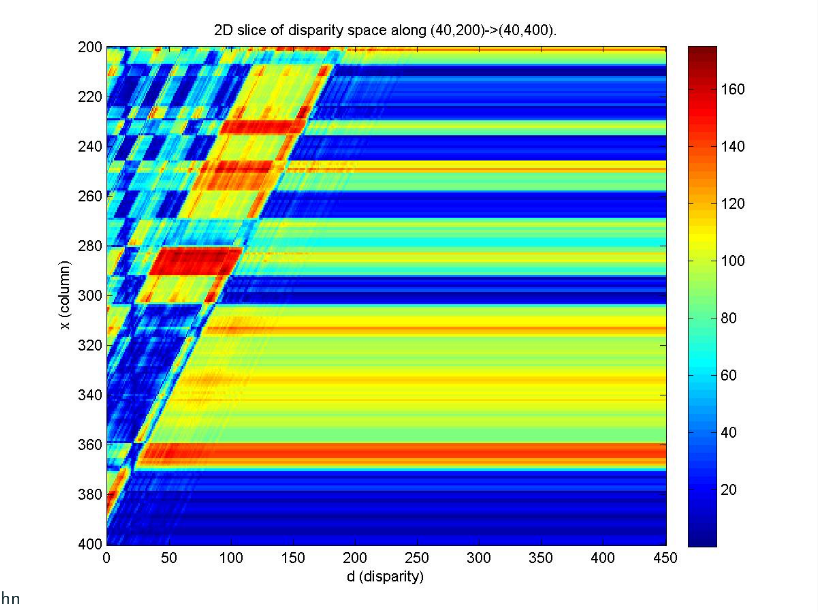
Step 1: Matching cost

- Penalty
- Cost for having dissimilarities
- Optimum $= 0$

Sum of absolute differences

$$\text{SAD} = \sum_{i,j \in U} |I_1(x_L + i, y_L + j) - I_2(x_R + i, y_R + j)|$$

Step 2: Disparity space image



Simple block-matching

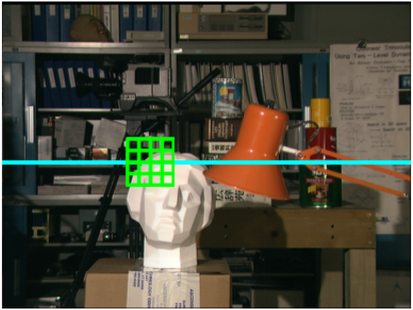
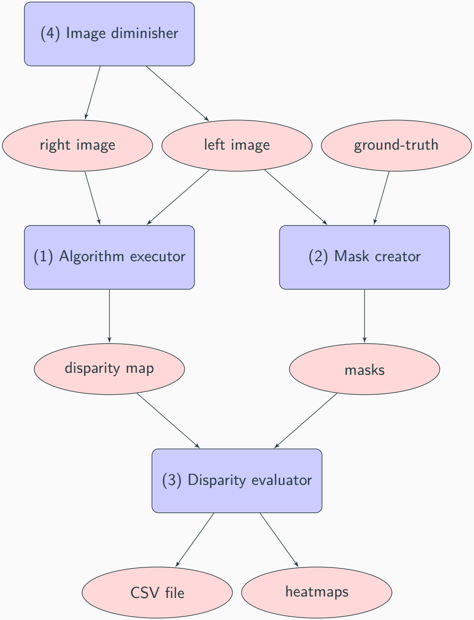


Illustration of block matching along a scanline.

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

```
step ← (wSize − 1)/2
C ← CreateMatrix(Cols(I_L), Rows(I_L), d_max)
for t ← 0 to Images(I_L) do
    leftImage ← I_L(t)
    rightImage ← I_L(t)
    for y ← 0 + step to Rows(I_L(0)) − step do
        for x ← 0 + step to Cols(I_L(0)) − step − d_max do
            for d ← 0 to d_max do
                rect_L ← Rect{x − step, y − step, wSize, wSize}
                rect_R ← Rect{x + d − step, y − step, wSize, wSize}
                window_L ← leftImage(rect_L)
                window_R ← rightImage(rect_R)
                C(x, y, t, d) ← MatchingCost(window_L, window_R)
return C
```

Spatiotemporal stereo matcher (2)

Algorithm 2: GetDisparityMap

Input: C, t

Output: *DisparityMap*

$DisparityMap \leftarrow \textsc{CreateMatrix}(\textsc{Cols}(C), \textsc{Rows}(C))$

for $t \leftarrow 0$ ***to*** $\textsc{Frames}(C)$ **do**

  for $y \leftarrow 0$ ***to*** $\textsc{Rows}(C)$ **do**

    for $x \leftarrow 0$ ***to*** $\textsc{Cols}(C)$ **do**

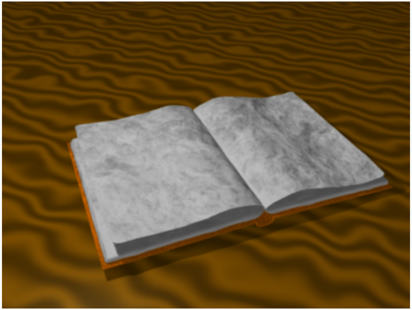
      $Cost \leftarrow \frac{1}{4} C(x, y, f_0) + \frac{2}{4} C(x, y, f_1) + \frac{1}{4} C(x, y, f_2)$

      $DisparityMap(x, y) \leftarrow \textsc{BestMatch}(Cost)$

return *DisparityMap*

Masking modes

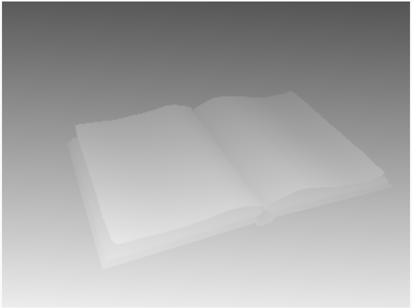
- Non-occluded mask
- Depth-discontinuity mask
- Textureless mask
- Saliency mask



Frame of book sequence

Masking modes

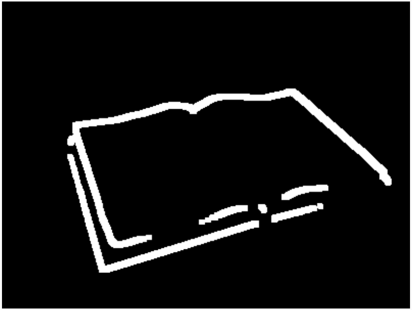
- Non-occluded mask
- Depth-discontinuity mask
- Textureless mask
- Saliency mask



Ground-truth disparity map

Masking modes

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



Depth-discontinuity at object borders

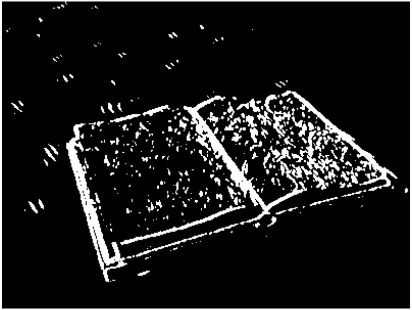
Masking modes

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

Non-occluded mask

Masking modes

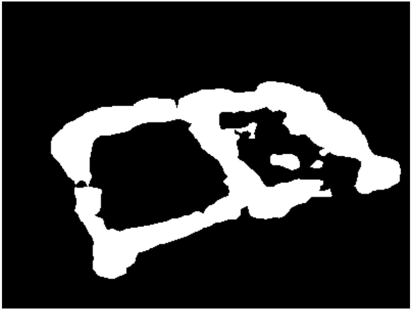
- Non-occluded mask
- Depth-discontinuity mask
- Textureless mask
- Saliency mask



Textureless regions

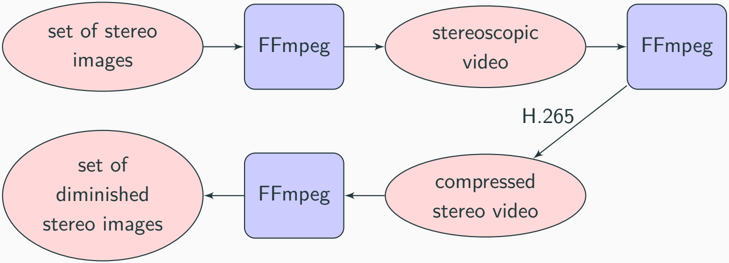
Masking modes

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



Salient pixels

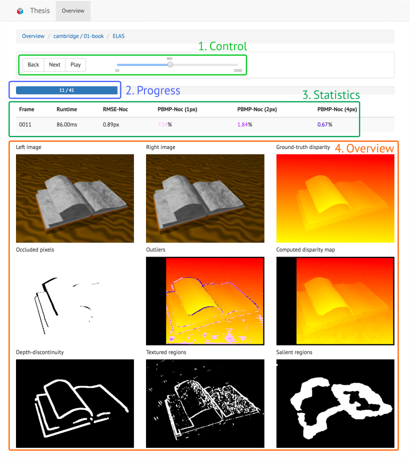
Image diminisher



Flow of FFmpeg as image diminisher.

Web viewer

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



Detail view

Demo

Evaluation

Overview

- Targeting videos (mean)
- Looking for outliers
- Masking modes
- Video compression

Datasets

- Cambridge
- SVDDD (high-resolution)

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**

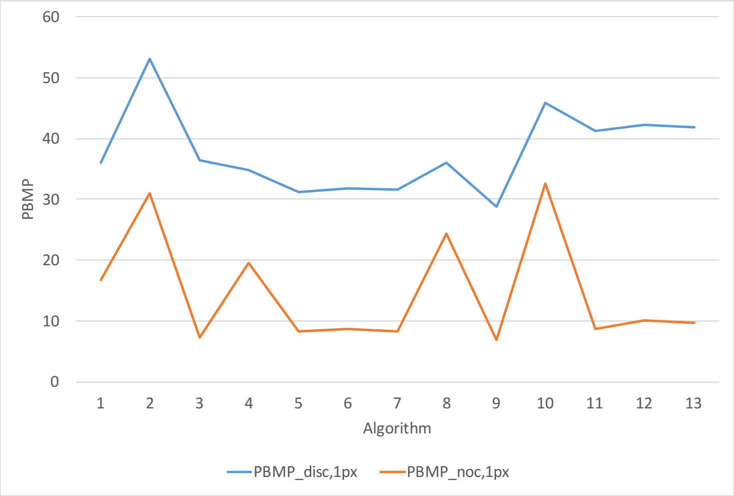
  $\sqrt{\frac{1}{n}\sum_{x,y=0}(d_a(x,y) - d_e(x,y))^2}$

Results

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

Result table for comparison of own implementation

Results



Depth-discontinuity mask applied on the book sequence.

Results

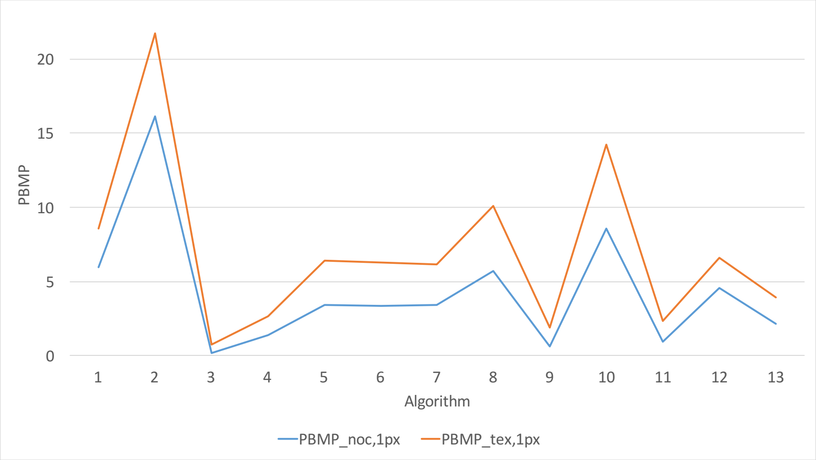


Chart of textureless region mask applied on the tunnel sequence.

Results

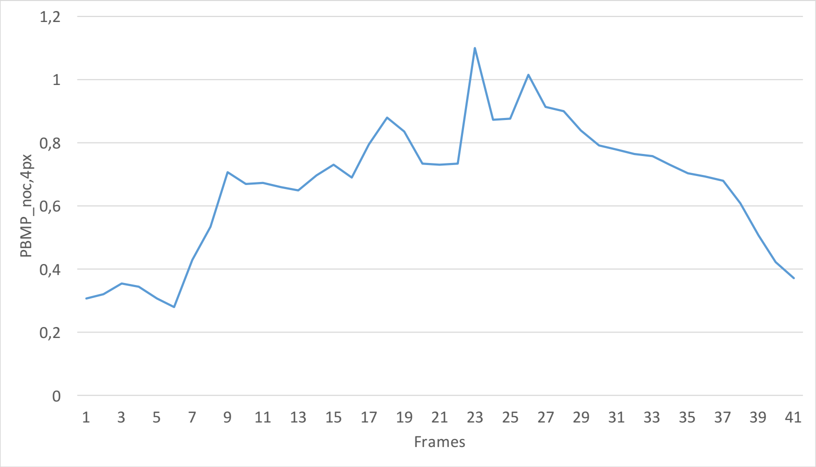
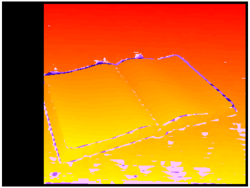
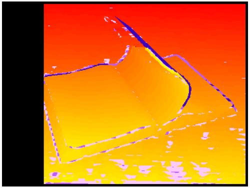


Chart of general outliers in a sequence.

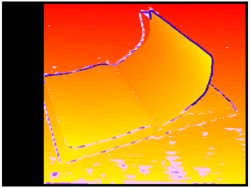
Results



(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

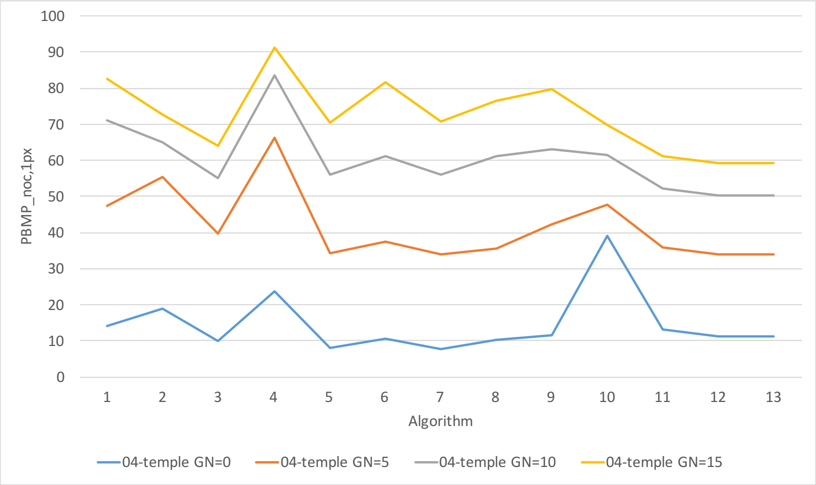
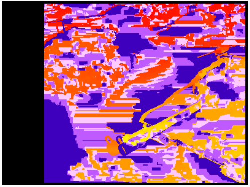
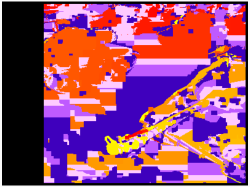


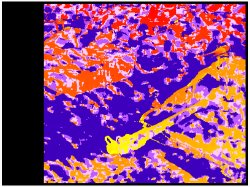
Chart of the impact of different σ^2 values for additive Gaussian noise on the result of disparity algorithms focusing on $P_{noc,1px}$.



(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

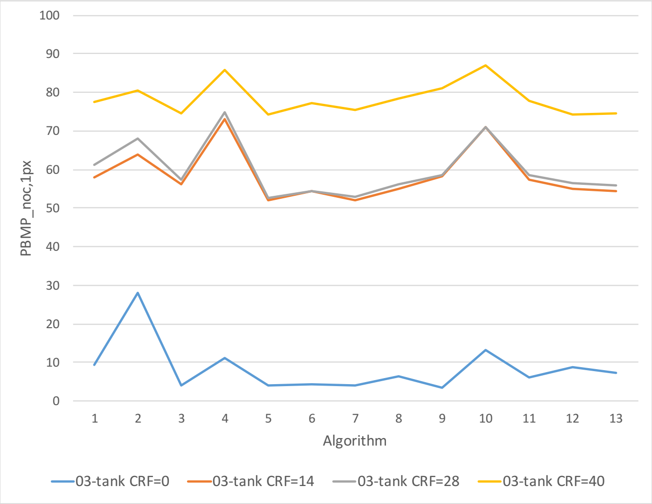
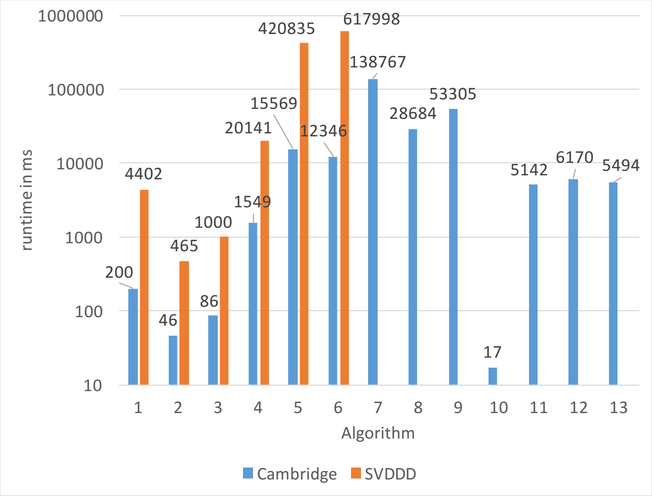


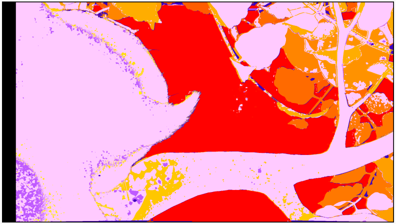
Chart of the impact of different CRF values for H.265 video compression on the result of disparity algorithms focusing on $\text{PBMP}_{noc,1px}$.

Results

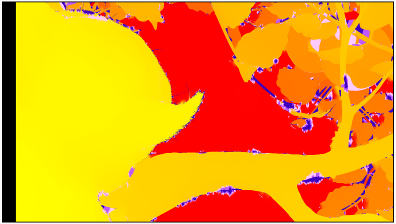


Comparison of the runtime of different disparity algorithms with both datasets, Cambridge and SVDDD

Results



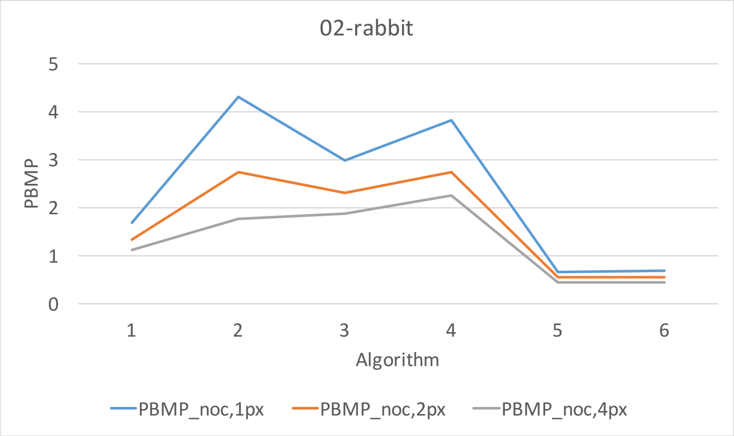
(a) Negative disparity



(b) Only positive disparity

Comparison of computed disparity maps regarding negative disparity.

Results



Performance of SVDDD rabbit scene

Results

	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 ($\text{PBMP}_{noc,1px}$)

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?

References I

S. Martull, M. Peris, and K. Fukui.
Realistic CG stereo image dataset with ground truth disparity maps.
In *ICPR workshop TrakMark2012*, volume 111, pages 117–118, 2012.