Subpixel accurate refinement of disparity maps using stereo correspondences

Matthias Demant

Lehrstuhl für Mustererkennung,
Universität Freiburg

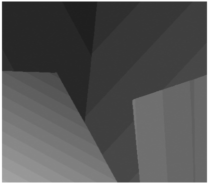
Outline

Outline

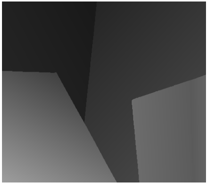
Problem statement

(a) Original image

(b) Reference image



(c) Integer disparity map



(d) Subpixel accurate disparity map

Figure: A typical example for integer disparities. The disparity map is traversed by stripes, the **staircase effect**.

The standard stereo configuration

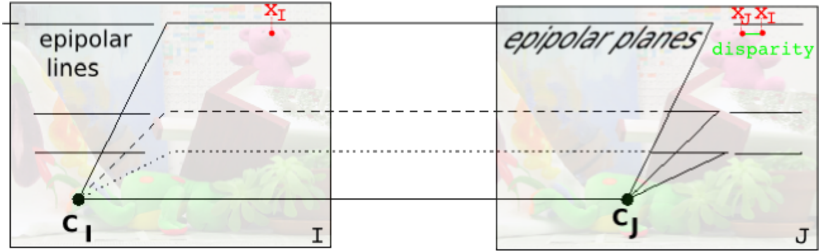


Figure: The standard stereo geometry system. Camera centers differ by a pure translation.

Standard stereo case

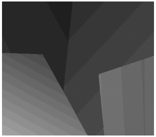
- Corresponding points have identical y-coordinates.
- Disparity = offset between corresponding points along scanline
- Depth is inverse proportional to disparity.

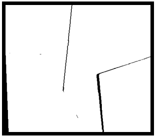
General approach with two steps

1. Pixel accurate disparity estimation
    - Required: Cost Volume
    - Search disparity with minimum costs

    ⇒ Disparity map (+- 2 pixel)

    ⇒ Occlusion handling





2. Subpixel refinement

Outline

Disparity Space Image

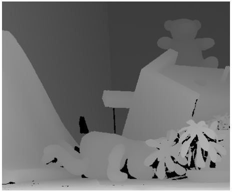
Wanted: Disparity map $\mathbf{d}(x, y)$
with $I_L(x, y) \approx I_R(x - \mathbf{d}(x, y), y)$

Cost Volume: $\mathrm{DSI}(x, y, d) = (I_L - I_R(x - d, y))^2$

Aggregation: with block, Gaussian, .. in 2D or 3D
reduce noise + environment informations

Start correspondence search

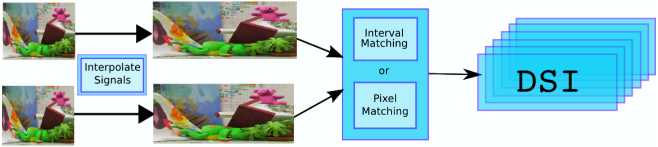
Assumptions: ordering constraint
smoothness constraint
...



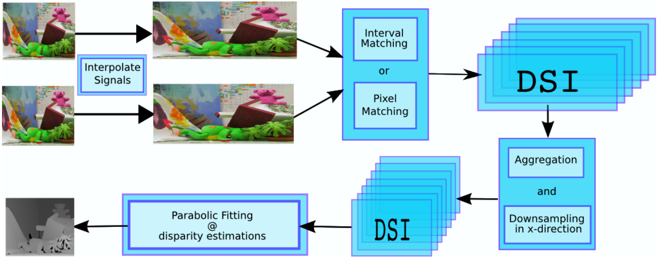
Continuous Disparity Space Image

Image sensor $\downarrow$	Integrates intensities over 1 Pixel
Sampling insensitive measure	Compare Intervals $[x - 0.5, x, x + 0.5]$ instead of Pixels
Wish:	Continuous DSI
Parabolic fitting	Continuous values after correspondence search
Sampling theorem	Samples of 0.5 pixel in x- and d-direction necessary to properly reconstruct the DSI

Parabolic Fitting Algorithm



Parabolic Fitting Algorithm



Parabolic Fitting Algorithm

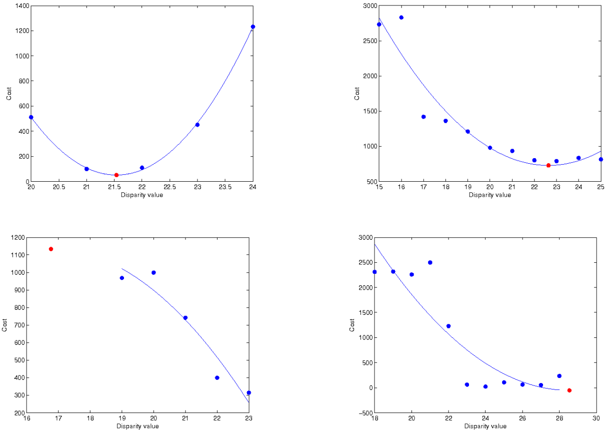
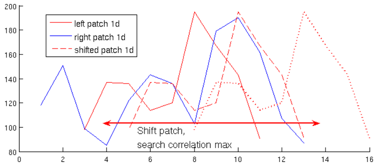


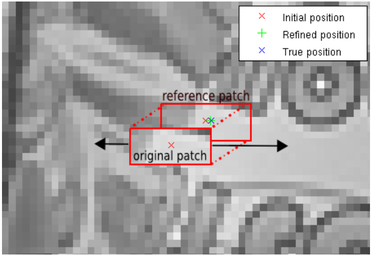
Figure: Fitting a parabola to 5 or 11 points around a value in the estimated cost volume (DSI). The red point specifies the calculated minimum. The images at the bottom are examples for unrefined values. The values are out of range of the fitted points or maxima.

Outline

Patch Correlation Algorithm



(a) 1D patch

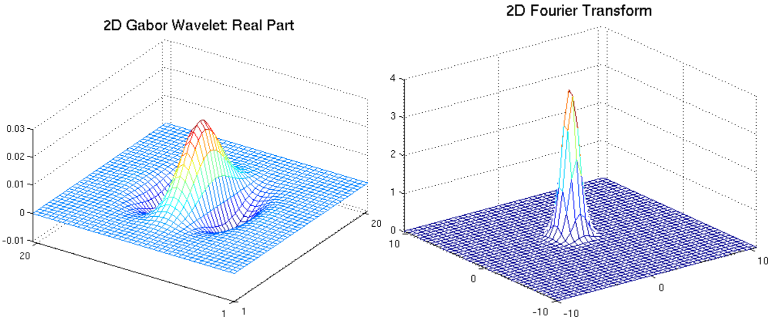


(b) 2D patch

Figure: Shifting the original patch over the reference patch.

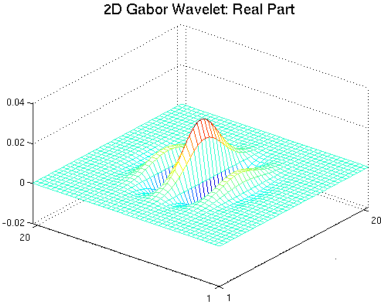
- Pixel accurate disparity map $\Rightarrow$ initial correspondence.
- Compare patch in left image with corresponding patch in right image.
- Shift and compare. The signals stay constant.

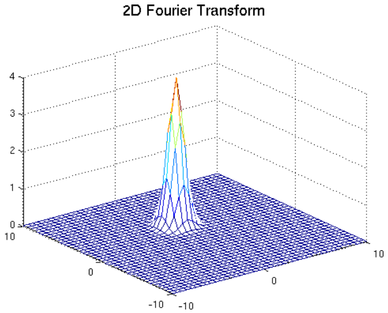
The Gabor Wavelet



- Gaussian multiplied with complex exponential
- Frequency domain: Gaussian at modulation frequency of complex exponential
- Acts like a bandpass filter
- Change modulation frequency → change position in frequency domain
- Windowing + expansion in frequency domain

The Gabor Wavelet





- Gaussian multiplied with complex exponential
- Frequency domain: Gaussian at modulation frequency of complex exponential
- Acts like a bandpass filter
- Change modulation frequency → change position in frequency domain
- Windowing + expansion in frequency domain

$t \leftrightarrow t'$ corresponding points
P_t Patch in Image I at position t,
$Q_{t'}$ Patch in Image J at position t'

1. Convolve images with a limited number of gabor filters

$\Rightarrow$ Patches expanded in frequency domain

$$a_t(k) = \int P_t(x) e^{-ikx} dx = \int I(x-t) \underbrace{g(x)e^{-ikx}}_{\text{“gabor” wavelet}} dx$$

2. Correlate and shift patches according to shift theorem

$$\langle P_{t-\delta t}, Q_{t'} \rangle \approx \sum_k a_t(k) e^{-ik\delta t} \overline{b_{t'}(k)} = \sum_k c_t(k) e^{-ik\delta t}$$

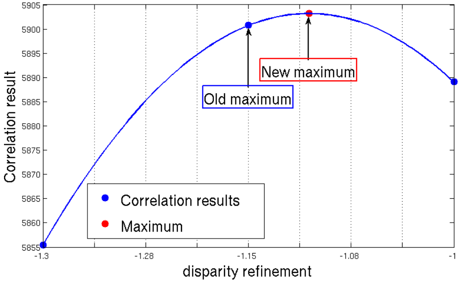
3. Find maximum of the correlation function
4. Confidence estimation with normalized correlation

Maximum Detection

Correlation Function

$$\langle P_{t-\delta t}, Q_{t'} \rangle \approx \sum_k a_t(k) e^{-ik\delta t} \overline{b_{t'}(k)} = \sum_k c_t(k) e^{-ik\delta t}$$

1. Discrete Fourier Transform + Parabolic Refinement
2. Iterative Techniques
   - Gradient Ascent
   - Newton Rhapson
   - Modified Newton Rhapson



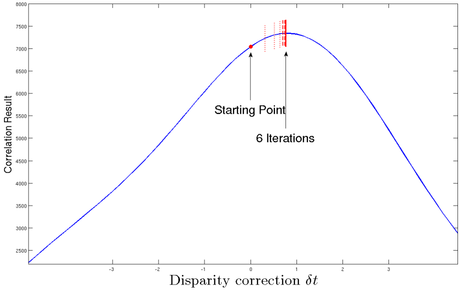
Maximum Detection

Correlation Function

$$\langle P_{t-\delta t}, Q_{t'} \rangle \approx \sum_k a_t(k) e^{-ik\delta t} \overline{b_{t'}(k)} = \sum_k c_t(k) e^{-ik\delta t}$$

1. Discrete Fourier Transform + Parabolic Refinement

2. Iterative Techniques
   - Gradient Ascent
   - Newton Rhapson
   - Modified Newton Rhapson



Overview

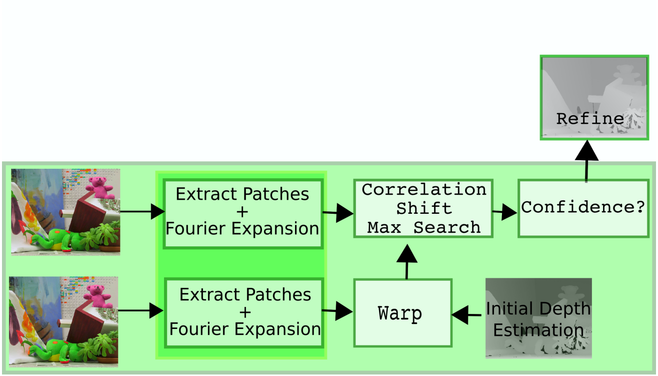


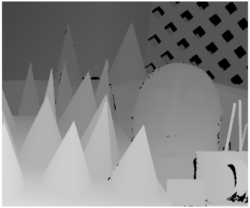
Figure: The Patch Correlation Algorithm

Outline

Real Data

(a) Original image

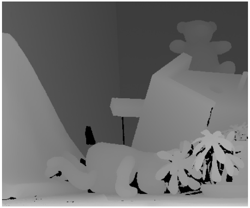
(b) Reference image



(c) Ground truth

(d) Original image

(e) Reference image



(f) Ground truth

Figure: The 'Cones' and 'Teddy' images with disparity map.

Starting Point

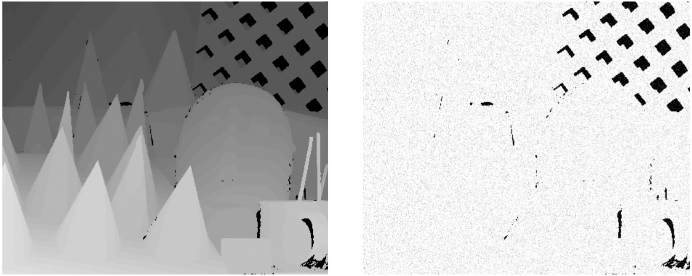
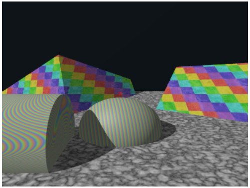


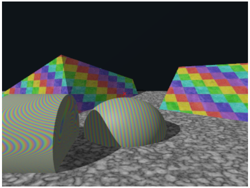
Figure: The integer disparity map (left) with added Gaussian noise (right). Our starting point of the refinement process.

- Integer disparity map + Gaussian noise $\sigma = 0.7$: Values in $[-2, +2]$
- Range $[+0.5, +1.5]$ with little Gaussian noise
- Middlebury stereo benchmark: Far behind last place.

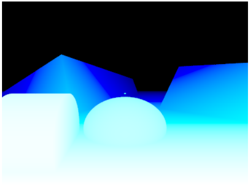
PovRay Scenes



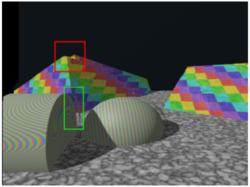
(a) Original image



(b) Reference image



(c) Depth map with 3×256 values



(d) Synthesized View

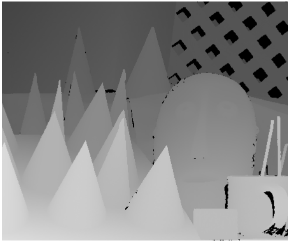
Figure: A povray scene with 3×256 depth values. Due to unpredictable discontinuities in the depth map the synthesized view in figure 8(d) is not correct (red box).

Quality Metrics

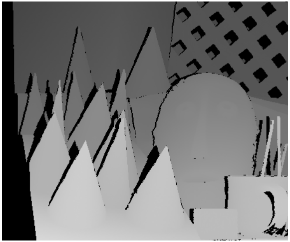
1. **RMS** (root-mean-square) error between the computed disparity map and the ground truth map
2. **BMP** Percentage of bad matching pixels
3. **RMS** error between synthetic views

(a) Original image (b) Forward warped image (c) Reference image

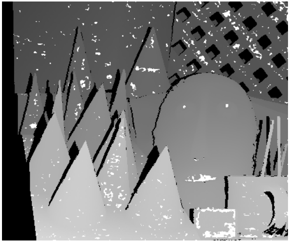
Figure: Using the disparity map to create different views. 9(b) The reference image in coordinates of the original image.



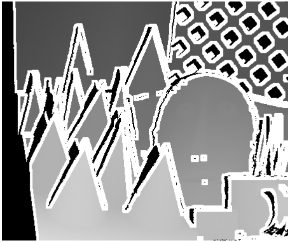
(a) Ground truth disparity map



(b) Occluded regions (black)



(c) Textureless regions (white)



(d) Discontinuity regions (white)

Outline

Parabolic Fitting

- Best results with Interval Matching
- Gaussian 2D aggregation (size 3,5 Pixel)
- Parabola fitted through 9 points

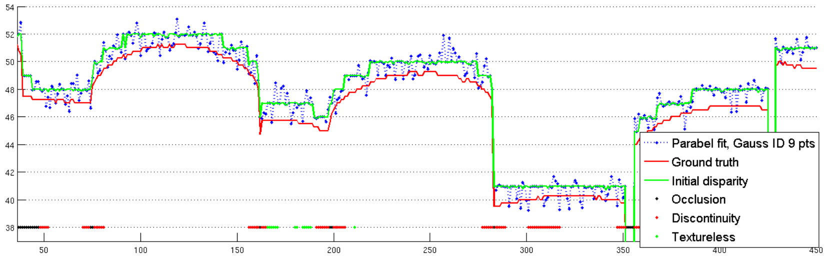
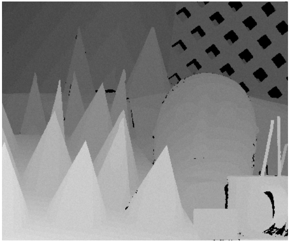


Figure: A complex scene, the refined line was achieved with a parabolic fitting to 9 points. The DSI was aggregated with a Gaussian 2D window using interval differences **ID**.

Results in different image regions

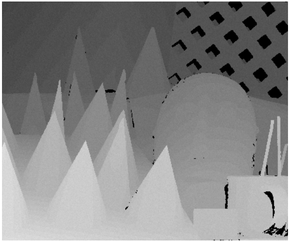


(a) Initial disp.map

(b) Result

(c) Result (Gradient)

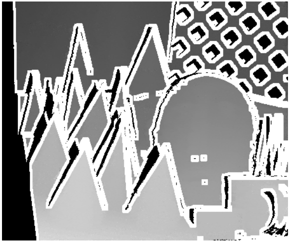
Results in different image regions



(f) Initial disp.map

(g) Result

(h) Result (Gradient)



Results - "Cones" line profile

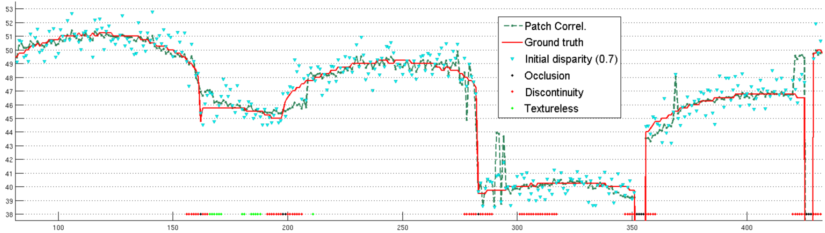


Figure: The refined line with the patch correlation algorithm.

Results - "Cones" line profile

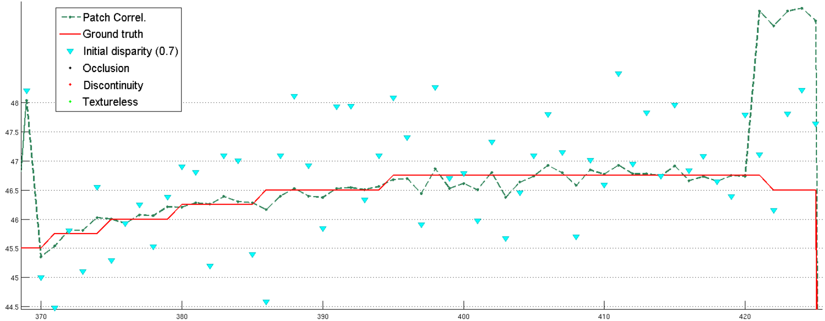
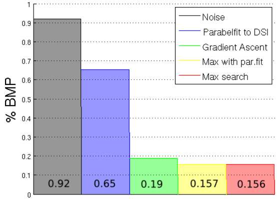
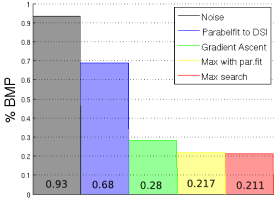


Figure: The refined line with the patch correlation algorithm.

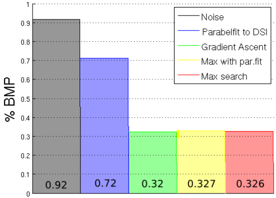
Results in different image regions



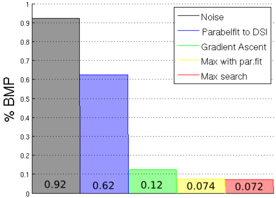
(a) $\overline{\mathcal{O}}$ Not occluded regions



(b) $\mathcal{T}$ Textureless regions



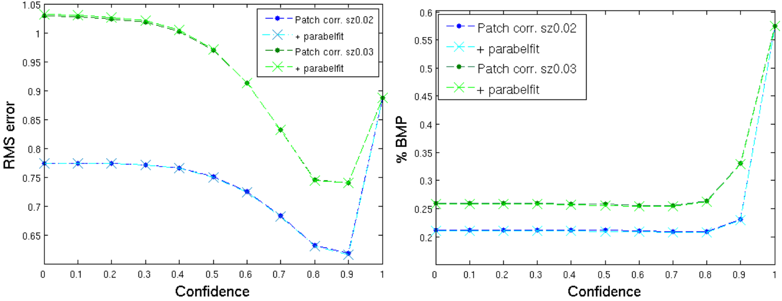
(c) $\mathcal{D}$ Discontinuity regions



(d) $\overline{\mathcal{D}}$ Non-discontinuity regions

Figure: Percentage of bad matching pixels with an error threshold of 0.7 pixel.

Patch Correlation - confidence and window size



(a) RMS

(b) %BMP, Threshold 0.5 Pixel

Figure: Changed Confidence and window size. Initial disparity map with gaussian noise: $\sigma = 0.7$

Iterative Techniques

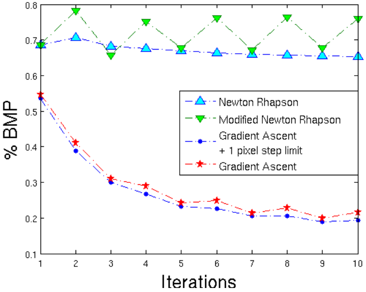


Figure: Bad matching pixels and iterations. The convergence of the different iterative techniques. The gradient ascent method converges in most cases in the global maximum.

Middlebury Ranking

Error Threshold = 0.75		Sort by nonocc			Sort by all						Sort by disc		
Algorithm	Avg. Rank	Tsukuba (ground truth) nonocc	Tsukuba all	Tsukuba disc	Venus (ground truth) nonocc	Venus all	Venus disc	Teddy (ground truth) nonocc	Teddy all	Teddy disc	Cones (ground truth) nonocc	Cones all	Cones disc
SubPixDoubleBP [30]	2.9	**4.91** 1	**5.62** 1	11.0 2	**0.22** 1	**0.57** 1	3.03 2	**5.46** 1	11.1 3	13.6 3	4.69 7	10.6 7	11.0 6
C-SemiGlob [19]	4.3	6.21 6	6.90 4	12.9 4	0.46 3	0.85 4	4.42 11	6.39 4	13.8 7	16.0 4	**3.29** 1	9.21 2	9.38 2
AdaptOvrSegBP [33]	5.4	5.98 3	6.56 2	**9.09** 1	0.44 2	0.58 2	**2.53** 1	8.78 9	13.7 6	19.3 9	5.34 13	10.9 10	11.3 7
AdaptingBP [17]	7.8	14.0 18	14.4 18	13.5 6	1.16 12	1.36 10	4.34 9	7.16 6	10.6 2	16.6 6	3.46 3	9.34 3	**9.19** 1
ImproveSubPix [25]	8.1	5.98 4	6.69 3	13.4 5	1.53 16	2.19 17	9.60 19	8.43 8	14.6 11	19.3 8	3.40 2	**8.98** 1	9.55 3
OverSegmBP [26]	8.2	7.75 10	8.17 8	13.8 7	0.86 8	1.11 5	6.03 14	8.89 10	14.6 12	19.5 10	4.02 5	9.97 5	10.4 4
DoubleBP2 [35]	8.8	18.7 22	19.1 21	15.8 11	0.82 5	1.18 6	3.65 5	5.82 3	11.2 4	13.4 2	4.76 8	10.8 9	11.5 9
DoubleBP [15]	9.9	18.7 23	19.1 22	15.8 12	0.83 7	1.34 9	3.84 6	5.77 2	11.4 5	**13.4** 1	4.80 10	11.3 12	11.6 10
Segm+visib [4]	12.2	12.7 14	12.9 12	15.8 13	2.64 24	3.08 23	10.2 21	7.04 5	**8.71** 1	16.3 5	4.86 11	9.86 4	12.2 13
SemiGlob [6]	12.5	7.34 8	8.37 9	18.4 21	1.96 22	2.60 20	12.8 26	7.47 7	14.4 9	19.6 11	3.61 4	10.7 8	10.4 5
PlaneFitBP [32]	14.1	12.7 15	13.6 15	16.2 16	0.83 6	1.26 7	3.24 3	9.40 15	15.3 15	19.2 7	6.97 25	13.8 22	15.2 23
DistinctSM [27]	14.5	13.5 17	14.3 17	16.2 15	0.92 9	1.31 8	4.38 10	9.42 16	15.7 16	22.0 15	6.28 20	12.4 19	12.1 12
SymBP+occ [7]	15.3	20.7 31	21.6 31	19.5 24	0.48 4	0.70 3	3.44 4	9.03 11	14.0 8	22.3 20	5.85 17	12.1 16	12.8 15
SO+borders [29]	17.4	19.1 25	19.5 23	20.0 26	1.67 21	2.05 16	4.63 12	9.28 14	15.2 14	20.2 12	5.62 15	11.8 14	13.1 17
AdaptWeight [12]	18.0	18.1 21	18.8 20	18.6 22	1.32 14	1.90 14	8.27 17	10.1 20	16.3 18	23.5 23	6.17 19	12.1 17	12.1 11
GenModel [20]	18.3	5.28 2	7.47 6	15.8 14	2.04 23	3.45 24	17.7 30	9.19 13	17.4 22	23.1 22	5.97 18	16.3 27	13.4 19
InteriorPtLP [34]	18.6	19.2 26	19.6 24	18.1 19	1.66 20	2.21 18	14.5 27	9.76 17	14.5 10	22.6 21	5.54 14	11.6 13	12.7 14
SegTreeDP [22]	19.0	25.4 38	26.0 36	24.6 34	1.29 13	1.53 11	4.21 7	12.3 26	18.4 26	22.0 14	4.78 9	10.1 6	11.4 8
RegionTreeDP [18]	19.6	21.0 32	21.1 30	18.3 20	1.09 11	1.57 12	4.21 7	10.0 19	15.1 13	20.8 13	9.80 28	15.5 25	16.1 25
SegmentSupport [28]	19.6	19.8 29	20.2 26	20.2 27	1.42 15	1.92 15	4.81 13	10.8 22	17.1 20	22.2 19	5.82 16	12.1 15	13.2 18
GC+occ [2]	19.8	6.10 5	7.11 5	14.6 8	3.20 26	3.80 26	8.40 18	14.7 27	21.3 27	24.4 25	6.81 23	14.0 24	15.8 24
TensorVoting [9]	19.8	16.8 19	17.7 19	15.2 9	1.66 19	2.37 19	12.4 25	11.1 23	18.4 25	26.8 30	5.18 12	12.4 18	14.0 20
EnhancedBP [24]	20.9	22.9 35	23.5 35	19.7 25	1.01 10	1.65 13	6.67 15	9.94 18	15.9 17	22.1 18	6.87 24	13.0 20	14.1 21
MultiCamGC [3]	21.2	6.56 7	7.55 7	15.7 10	6.33 32	6.75 32	6.67 15	15.5 29	21.7 29	26.8 29	6.35 21	13.3 21	14.8 22
CostRelax [11]	21.2	19.5 27	20.6 28	29.5 36	1.65 18	2.73 21	19.4 31	9.18 12	17.1 21	26.3 28	4.37 6	10.9 11	12.9 16
RealtimeBP [21]	23.5	19.9 30	21.6 32	22.2 28	1.61 17	2.82 22	11.0 23	11.2 24	16.4 19	22.1 16	6.66 22	13.9 23	16.9 26
Layered [5]	24.2	13.0 16	13.5 14	18.7 23	4.71 30	5.33 29	10.6 22	12.0 25	17.9 24	23.7 24	9.17 27	17.3 28	18.3 28
GC [1d]	25.8	7.71 9	9.82 11	17.4 17	3.96 28	5.60 30	11.3 24	21.3 37	29.4 37	29.3 31	9.84 29	20.1 30	18.3 27
[illegible]R METHOD	27.1	8.54 11	[illegible] 10	[illegible] 3	13.1 [illegible]	[illegible] 37	[illegible] 37	16.1 30	[illegible] 30	[illegible] 32	12.5 31	21.5 31	29.8 35
RealTimeGPU [14]	28.0	24.2 36	26.0 37	24.9 35	3.30 27	4.43 27	22.0 32	10.2 21	17.8 23	22.1 17	8.68 26	16.2 26	20.6 29
ReliabilityDP [13]	28.2	19.0 24	20.7 29	17.5 18	3.98 29	5.20 28	15.9 29	14.7 28	21.6 28	25.1 26	15.7 35	22.6 34	23.5 31
TreeDP [8]	30.6	22.4 34	23.1 33	22.3 29	2.86 25	3.60 25	10.2 20	21.3 36	28.9 36	33.7 35	13.7 32	21.7 32	23.3 30
PhaseBased [31]	31.0	11.2 12	13.4 13	23.9 32	7.81 34	9.28 34	27.9 36	20.4 35	28.5 35	32.7 34	15.9 36	25.2 37	27.5 34
DP [1b]	32.0	19.6 28	20.6 27	22.8 30	13.6 39	14.5 38	24.1 33	19.2 32	26.3 32	25.6 27	13.8 33	22.1 33	25.7 32
PhaseDiff [23]	32.8	11.5 13	13.6 16	24.5 33	9.48 35	10.9 35	27.6 35	24.9 38	32.5 38	34.9 36	24.1 39	32.4 39	32.9 37
SO [1c]	33.8	17.9 20	19.8 25	23.4 31	13.1 37	14.5 39	25.2 34	25.6 39	33.3 39	31.3 33	16.0 37	25.6 38	26.8 33
SSD+MF [1a]	34.6	28.5 39	30.0 39	38.1 38	4.98 31	6.43 31	14.5 28	19.8 33	27.8 34	38.2 37	13.8 34	22.8 35	31.2 36
STICA [16]	35.2	24.3 37	26.1 38	44.8 39	9.65 36	11.0 36	42.6 39	18.2 31	25.8 31	42.8 38	11.6 30	19.6 29	33.1 38
Infection [10]	35.6	21.9 33	23.3 34	37.0 37	6.50 33	7.67 33	33.6 38	20.1 34	27.7 33	49.2 39	16.5 38	23.9 36	42.1 39

29 von 39

Figure: A screenshot of the Middlebury ranking with an error threshold of 0.75.

Outline

Conclusion and Outlook

Initial estimation should be replaced by real results.

Synthetic data showed unpredictable errors. Applicable with small disparity ranges

Parabolic fitting to DSI is appropriated for small errors < 1 pixel. Intervall Matching better

Patch correlation: 75% closer than 0.5 Pixels Errors in discontinuity regions.

Scene information: Add semi-global character using informations about

- sharp areas or
- scene geometry.

Blurring

Assumption: Well-known scene geometry

Question: How can we emphasize small structures?

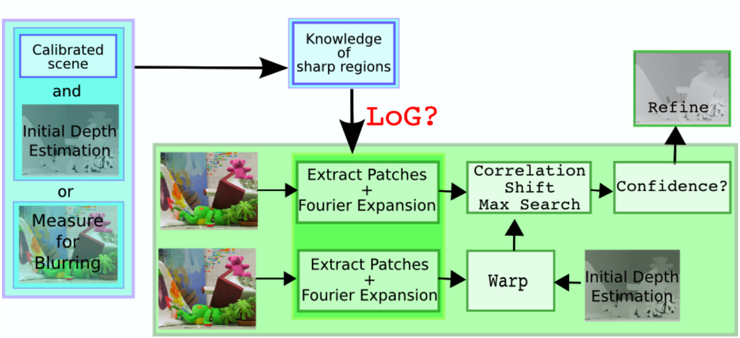
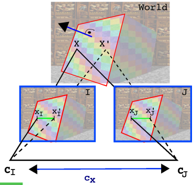
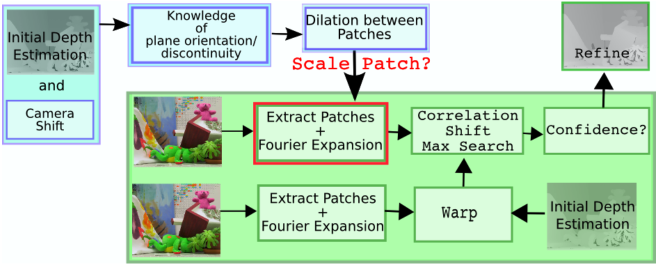


Figure: With a well known scene setup blurred image regions can be detected (blue). In sharp areas the high frequencies can be emphasized applying the Laplacian of Gaussian. The Patch Correlation algorithm (green) can be extended with little additional costs.

Dilation between patches

Points on planes + camera shift + plane orientation
$\Rightarrow$
dilation between pairs of corresponding points (blue)





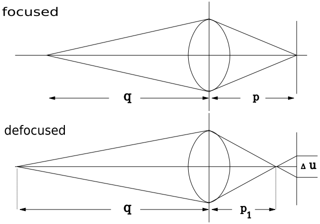
The Patch Correlation algorithm (green) extracts both patches. One of the extracted patches can be scaled according to the dilation.

Thank you for your attention.

Model of the Point-spread function

Assumption: Well-known scene geometry

Question: How and when can we emphasize small structures?



Point spread function: The brightness of the blurred area can be approximated by a two dimensional Gaussian distribution.

- Brightness distribution depends on distance.
- A high σ value affects like a low-pass filter.
- Emphasize small stuctures in sharp areas: LoG
- Easy extandable to patch correlation algorithm

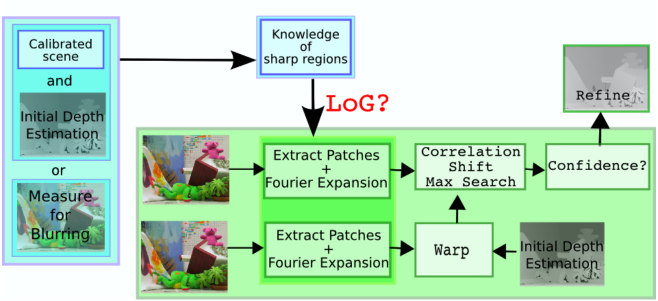


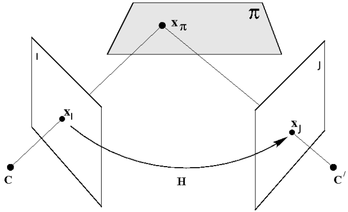
Figure: With a well known scene setup blurred image regions can be detected (blue). In sharp areas the high frequencies can be emphasized applying the Laplacian of Gaussian. The Patch Correlation algorithm (green) can be extended with little additional costs.

Homography induced by a plane

Assumption: Standard stereo configuration

Question: How can we classify transformations between corresponding points on same world plane?

$$\mathbf{x}_J = \mathbf{H}_{JI}(\pi)\mathbf{x}_I$$



- Back-trace ray $\mathbf{C} \rightarrow \mathbf{x}$ via $\mathbf{P}_I^+$ to π
- Project $\mathbf{x}_\pi$ via $\mathbf{P}_J$ to J

$\mathbf{H}_{JI}(\pi) = \mathbf{K}_J \left(I + \mathbf{c}'\mathbf{p}^T \right) \mathbf{K}_I^{-1}$.

with $\mathbf{c}' = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}^T$

and $\pi = \begin{pmatrix} \mathbf{p} & 1 \end{pmatrix}^T$

The calibration matrices $\mathbf{K}_I, \mathbf{K}_I^{-1}$ and $\mathbf{K}_J$ are affine transformations.
The center part $\left(I + \mathbf{c}\mathbf{p}^T \right)$ is affine, if $c_z = 0$.

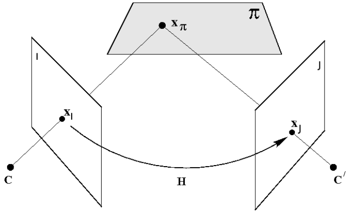
In the standard stereo case $\mathbf{H}_{JI}(\pi)$ is an affine transformation ($c_y = c_z = 0$).

Homography induced by a plane

Assumption: Standard stereo configuration

Question: How can we classify transformations between corresponding points on same world plane?

$$\mathbf{x}_J = \mathbf{H}_{JI}(\pi)\mathbf{x}_I$$



- Back-trace ray $\mathbf{C} \rightarrow \mathbf{x}$ via $\mathbf{P}_I^+$ to π
- Project $\mathbf{x}_\pi$ via $\mathbf{P}_J$ to J

$\mathbf{H}_{JI}(\pi) = \mathbf{K}_J \left(I + \mathbf{c}'\mathbf{p}^T \right) \mathbf{K}_I^{-1}$.

with $\mathbf{c}' = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}^T$

and $\pi = \begin{pmatrix} \mathbf{p} & 1 \end{pmatrix}^T$

The calibration matrices $\mathbf{K}_I, \mathbf{K}_I^{-1}$ and $\mathbf{K}_J$ are affine transformations.
The center part $\left(I + \mathbf{cp}^T \right)$ is affine, if $c_z = 0$.

In the standard stereo case $\mathbf{H}_{JI}(\pi)$ is an affine transformation ($c_y = c_z = 0$).

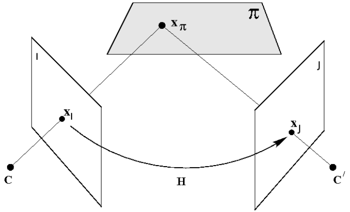
Homography induced by a plane

Assumption: Standard stereo configuration

Question: How can we classify transformations between corresponding points on same world plane?

$$\mathbf{x}_J = \mathbf{H}_{JI}(\pi)\mathbf{x}_I$$

- Back-trace ray $\mathbf{C} \rightarrow \mathbf{x}$ via $\mathbf{P}_I^+$ to π
- Project $\mathbf{x}_\pi$ via $\mathbf{P}_J$ to J



$$\mathbf{H}_{JI}(\pi) = \mathbf{K}_J \left(I + \mathbf{c}'\mathbf{p}^T \right) \mathbf{K}_I^{-1}.$$

with $\mathbf{c}' = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}^T$

and $\pi = \begin{pmatrix} \mathbf{p} & 1 \end{pmatrix}^T$

The calibration matrices $\mathbf{K}_I, \mathbf{K}_I^{-1}$ and $\mathbf{K}_J$ are affine transformations.

The center part $\left(I + \mathbf{c}\mathbf{p}^T \right)$ is affine, if $c_z = 0$.

In the standard stereo case $\mathbf{H}_{JI}(\pi)$ is an affine transformation $(c_y = c_z = 0)$.

Homography induced by a plane

Assumption: Standard stereo configuration

Question: How can we classify transformations between corresponding points on same world plane?

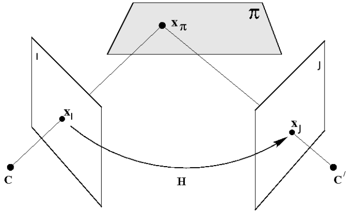
$$\mathbf{x}_J = \mathbf{H}_{JI}(\pi)\mathbf{x}_I$$

- Back-trace ray $\mathbf{C} \rightarrow \mathbf{x}$ via $\mathbf{P}_I^+$ to π
- Project $\mathbf{x}_\pi$ via $\mathbf{P}_J$ to J

$$\mathbf{H}_{JI}(\pi) = \mathbf{K}_J \left(I + \mathbf{c}'\mathbf{p}^T \right) \mathbf{K}_I^{-1}.$$

with $\mathbf{c}' = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}^T$

and $\pi = \begin{pmatrix} \mathbf{p} & 1 \end{pmatrix}^T$



The calibration matrices $\mathbf{K}_I, \mathbf{K}_I^{-1}$ and $\mathbf{K}_J$ are affine transformations.
The center part $\left(I + \mathbf{c}\mathbf{p}^T \right)$ is affine, if $c_z = 0$.

In the standard stereo case $\mathbf{H}_{JI}(\pi)$ is an affine transformation ($c_y = c_z = 0$).

Homography induced by a plane

Assumption: Standard stereo configuration

Question: How can we classify transformations between corresponding points on same world plane?

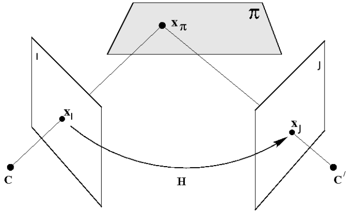
$$\mathbf{x}_J = \mathbf{H}_{JI}(\pi)\mathbf{x}_I$$

- Back-trace ray $\mathbf{C} \rightarrow \mathbf{x}$ via $\mathbf{P}_I^+$ to π
- Project $\mathbf{x}_\pi$ via $\mathbf{P}_J$ to J

$\mathbf{H}_{JI}(\pi) = \mathbf{K}_J \left(I + \mathbf{c}'\mathbf{p}^T\right) \mathbf{K}_I^{-1}$.

with $\mathbf{c}' = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}^T$

and $\pi = \begin{pmatrix} \mathbf{p} & 1 \end{pmatrix}^T$



The calibration matrices $\mathbf{K}_I, \mathbf{K}_I^{-1}$ and $\mathbf{K}_J$ are affine transformations.
The center part $\left(I + \mathbf{cp}^T\right)$ is affine, if $c_z = 0$.

In the standard stereo case $\mathbf{H}_{JI}(\pi)$ is an affine transformation ($c_y = c_z = 0$).

Homography induced by a plane

Assumption: Standard stereo configuration

Question: How can we classify transformations between corresponding points on same world plane?

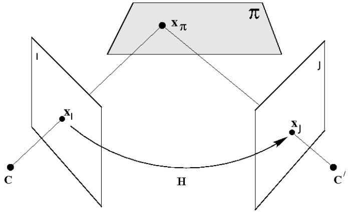
$$\mathbf{x}_J = \mathbf{H}_{JI}(\boldsymbol{\pi})\mathbf{x}_I$$

- Back-trace ray $\mathbf{C} \rightarrow \mathbf{x}$ via $\mathbf{P}_I^+$ to $\boldsymbol{\pi}$
- Project $\mathbf{x}_\pi$ via $\mathbf{P}_J$ to J

$\mathbf{H}_{JI}(\boldsymbol{\pi}) = \mathbf{K}_J \left(I + \mathbf{c}'\mathbf{p}^T\right) \mathbf{K}_I^{-1}$.

with $\mathbf{c}' = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}^T$

and $\boldsymbol{\pi} = \begin{pmatrix} \mathbf{p} & 1 \end{pmatrix}^T$



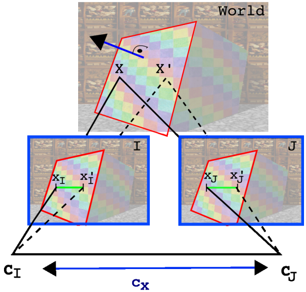
The calibration matrices $\mathbf{K}_I, \mathbf{K}_I^{-1}$ and $\mathbf{K}_J$ are affine transformations. The center part $\left(I + \mathbf{c}\mathbf{p}^T\right)$ is affine, if $c_z = 0$.

In the standard stereo case $\mathbf{H}_{JI}(\boldsymbol{\pi})$ is an affine transformation ($c_y = c_z = 0$).

Scene planes, disparities and patches

Definition: Iso-disparity surfaces have identical size in both images.
Assumption: Points on same world plane, simplified calibration matrices
Question: How does texture change, depending on plane orientation?

$$\mathbf{x}_J = \mathbf{H}_{JI}(\pi)\mathbf{x}_I$$
$$\mathbf{x}_J - \mathbf{x}'_J = \mathbf{H}_{JI}(\pi)\left(\mathbf{x}_I - \mathbf{x}'_I\right)$$



For corresponding points:

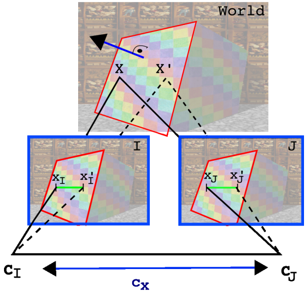
$$x_J - x'_J = (1 + c_x p_x)(x_I - x'_I)$$

Iso-disparity surfaces are fronto parallel planes. All patches on these surfaces keep the same patch size in both images.

Scene planes, disparities and patches

Definition: Iso-disparity surfaces have identical size in both images.
Assumption: Points on same world plane, simplified calibration matrices
Question: How does texture change, depending on plane orientation?

$$\mathbf{x}_J = \mathbf{H}_{JI}(\boldsymbol{\pi})\mathbf{x}_I$$
$$\mathbf{x}_J - \mathbf{x}'_J = \mathbf{H}_{JI}(\boldsymbol{\pi})\left(\mathbf{x}_I - \mathbf{x}'_I\right)$$

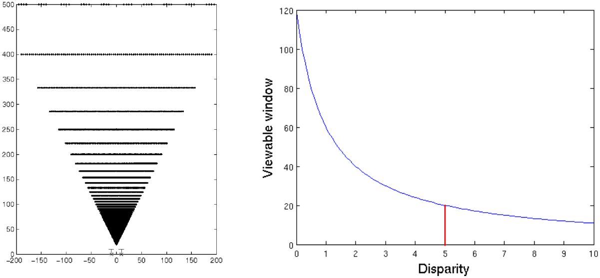


For corresponding points:

$$x_J - x'_J = (1 + c_x p_x)(x_I - x'_I)$$

Iso-disparity surfaces are fronto parallel planes. All patches on these surfaces keep the same patch size in both images.

The viewable window on fronto-parallel planes



(a) Two cameras and iso-disparity surfaces

(b) Relation between disparity and the size of the viewable window.

Figure: The viewable window decreases with increasing disparity.

Isodisparity surfaces and focal length

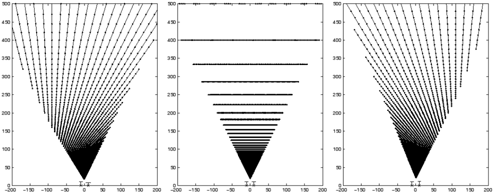


Figure: Iso-disparity surfaces in the standard stereo configuration with different focal length. The center figure represents equal focal length.

Applicability in a well-known camera setup

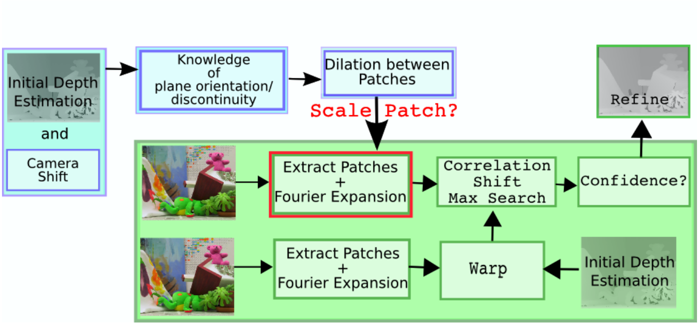


Figure: With a known camera shift and the initial disparity map a dilation between patches can be corrected (blue). The Patch Correlation algorithm (green) extracts both patches. One of the extracted patches can be scaled according to the dilation.