Stereo — 2D to 3D on an FPGA

Brian Axelrod, Sheena Nie, Amartya Shankha Biswas

Massachusetts Institute of Technology

baxelrod, xnie, asbiswas

November 10, 2015

What is Stereo

- Monocular

What is Stereo

- Monocular
- Binocular

What is Stereo

- Monocular
- Binocular
    - two perspectives

What is Stereo

- Monocular
- Binocular
  - two perspectives
- Algorithmic Idea

What is Stereo

- Monocular
- Binocular
  - two perspectives
- Algorithmic Idea
- Computationally Difficult

What is Stereo

- Monocular
- Binocular
  - two perspectives
- Algorithmic Idea
- Computationally Difficult
  - 2x HD camera

What is Stereo

- Monocular
- Binocular
  - two perspectives
- Algorithmic Idea
- Computationally Difficult
  - 2x HD camera
  - 4 million pixels

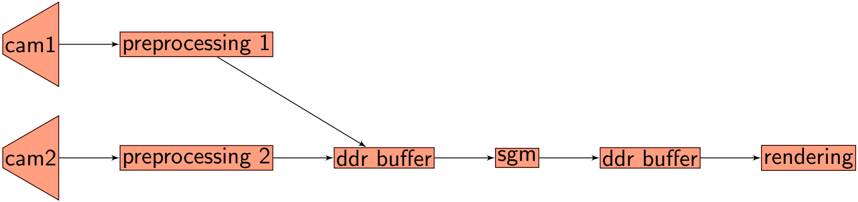
What is Stereo

- Monocular
- Binocular
  - two perspectives
- Algorithmic Idea
- Computationally Difficult
  - 2x HD camera
  - 4 million pixels
  - 4.3 Trillion possible correspondences

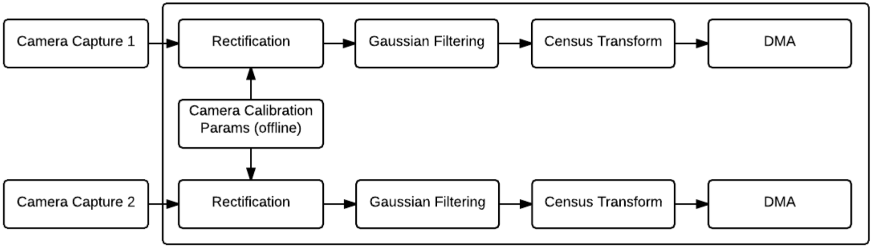
What is Stereo

- Monocular
- Binocular
  - two perspectives
- Algorithmic Idea
- Computationally Difficult
  - 2x HD camera
  - 4 million pixels
  - 4.3 Trillion possible correspondences
  - Even smart algorithms require a lot of power

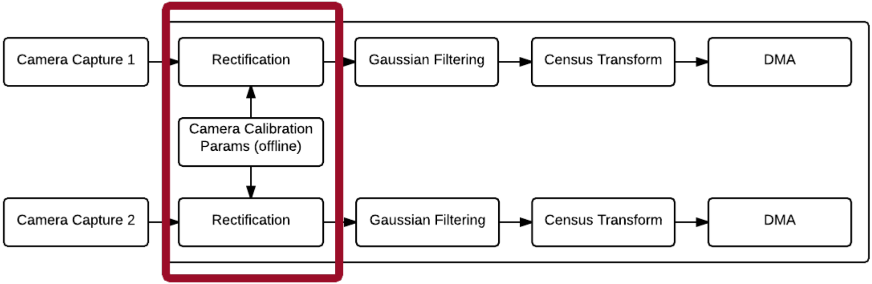
Overall System Layout



Camera Pipeline



Camera Pipeline



Why Rectification?

- Intrinsic optical distortion

Why Rectification?

- Intrinsic optical distortion
- Improper alignment of the cameras (rotation, translation)

Rectification

- Calibrate cameras offline (MATLAB)

- Calibrate cameras offline (MATLAB)
- Acquire rotation, translation matrix coefficients

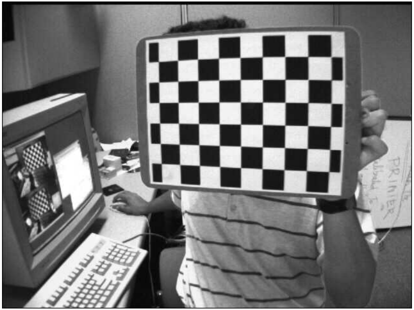
Rectification

- Calibrate cameras offline (MATLAB)
- Acquire rotation, translation matrix coefficients
- Acquire intrinsic distortion parameters

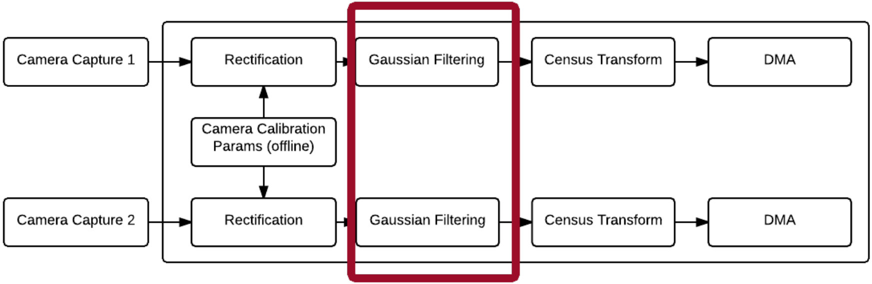
Rectification

- Calibrate cameras offline (MATLAB)
- Acquire rotation, translation matrix coefficients
- Acquire intrinsic distortion parameters
- Apply the matrix transformations to streamed images (real-time)

Rectification: Example



Camera Pipeline



Gaussian Filtering

Original image

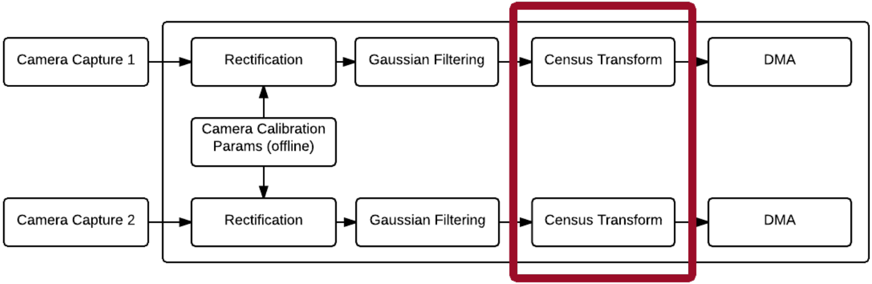
Gaussian Blur applied

Gaussian Filtering

- Reduce noise in the image
- Convolution: weighted sum of surrounding pixels
- Separate horizontal and vertical passes

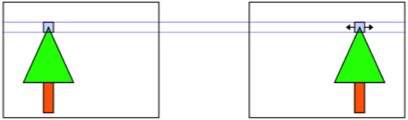
$$s = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Camera Pipeline



Disparity

- A shift to the left of an image feature when viewed in the right image
- Disparity cost: associated matching cost between two pixels

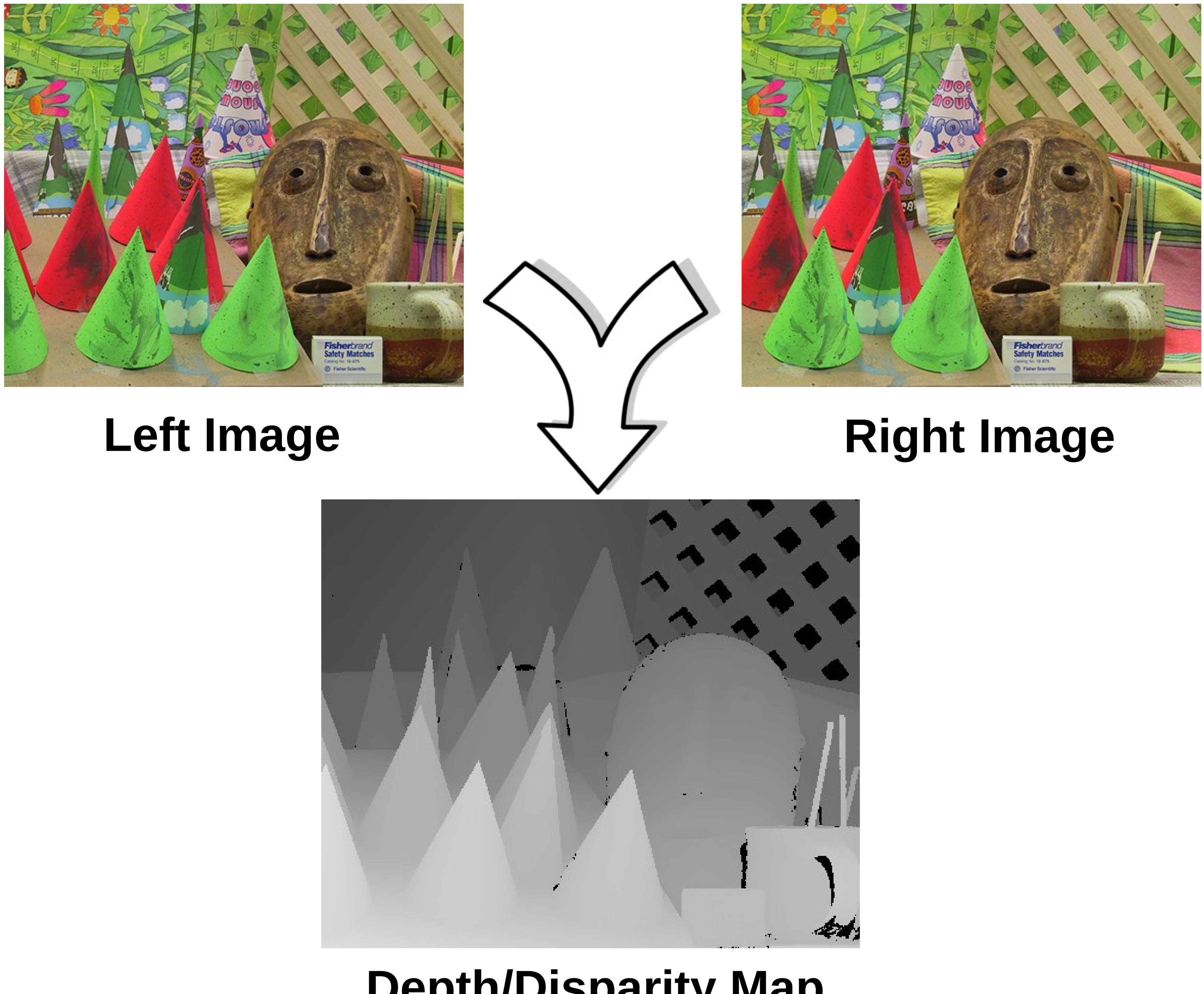


Census Transform

- Used to compute disparity matching costs
- 5x5 window for each pixel to represent the information from the surroundings of the pixel
- produces a bit stream

Left Image

Right Image


Left Image
Right Image
Depth/Disparity Map

Minimize Global Energy

Minimize Global Energy

$$
\begin{aligned}
E(\boldsymbol{D}) \; &= \sum_{\boldsymbol{p}} C(\boldsymbol{p}, D_{\boldsymbol{p}}) \\
&+ \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} \mathcal{P}_1 \cdot T\left[|D_{\boldsymbol{p}} - D_{\boldsymbol{q}}| = 1\right] \\
&+ \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} \mathcal{P}_2 \cdot T\left[|D_{\boldsymbol{p}} - D_{\boldsymbol{q}}| > 1\right]
\end{aligned}
$$

Minimize Global Energy

$E(\boldsymbol{D})$ *D is a disparity image*

Minimize Global Energy

$$E(\boldsymbol{D}) = \sum_{\boldsymbol{p}} C(\boldsymbol{p}, D_{\boldsymbol{p}})$$

Similarity Costs for Each Pixel

Minimize Global Energy

$$E(\boldsymbol{D}) = \sum_{\boldsymbol{p}} C(\boldsymbol{p}, D_{\boldsymbol{p}}) + \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} T\left[|D_{\boldsymbol{p}} - D_{\boldsymbol{q}}| = 1\right]$$

Small Disparity Change (Smoothness)

Minimize Global Energy

$$E(\boldsymbol{D}) = \sum_{\boldsymbol{p}} C(\boldsymbol{p}, D_{\boldsymbol{p}}) + \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} \mathcal{P}_1 \cdot T\left[|D_{\boldsymbol{p}} - D_{\boldsymbol{q}}| = 1\right]$$

Small Disparity Change (Smoothness)
Small Penalty P_1

Minimize Global Energy

$$E(\boldsymbol{D}) = \sum_{\boldsymbol{p}} C(\boldsymbol{p}, D_{\boldsymbol{p}}) + \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} \mathcal{P}_1 \cdot T\left[|D_{\boldsymbol{p}} - D_{\boldsymbol{q}}| = 1\right] + \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} T\left[|D_{\boldsymbol{p}} - D_{\boldsymbol{q}}| > 1\right]$$

Object Boundary

Minimize Global Energy

$$E(\boldsymbol{D}) = \sum_{\boldsymbol{p}} C(\boldsymbol{p}, D_{\boldsymbol{p}}) + \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} \mathcal{P}_1 \cdot T\left[|D_{\boldsymbol{p}} - D_{\boldsymbol{q}}| = 1\right] + \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} \mathcal{P}_2 \cdot T\left[|D_{\boldsymbol{p}} - D_{\boldsymbol{q}}| > 1\right]$$

Object Boundary : Large Penalty P_2

2D Global Minimization

$$
\begin{aligned}
E(\boldsymbol{D}) \quad &= \sum_{\boldsymbol{p}} C(\boldsymbol{p}, D_{\boldsymbol{p}}) \\
&+ \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} \mathcal{P}_1 \cdot T\left[|D_{\boldsymbol{p}} - D_{\boldsymbol{p}}| = 1 \right] \\
&+ \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} \mathcal{P}_2 \cdot T\left[|D_{\boldsymbol{p}} - D_{\boldsymbol{p}}| > 1 \right]
\end{aligned}
$$

2D Global Minimization

$$
\begin{aligned}
E(\boldsymbol{D}) \quad &= \sum_{\boldsymbol{p}} C(\boldsymbol{p}, D_{\boldsymbol{p}}) \\
&+ \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} \mathcal{P}_1 \cdot T\left[|D_{\boldsymbol{p}} - D_{\boldsymbol{p}}| = 1\right] \\
&+ \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} \mathcal{P}_2 \cdot T\left[|D_{\boldsymbol{p}} - D_{\boldsymbol{p}}| > 1\right]
\end{aligned}
$$

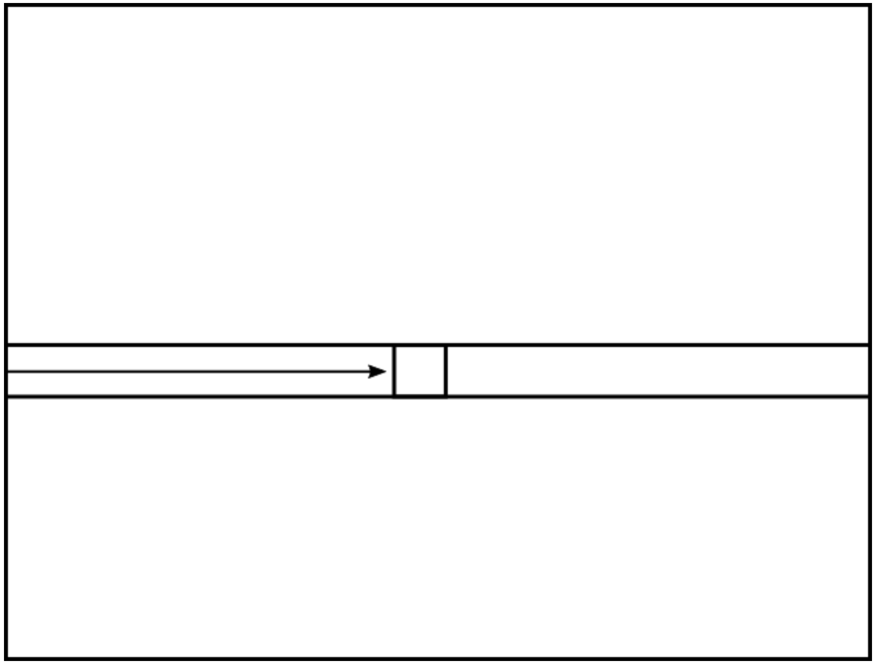
NP-Complete Problem

1D Optimization

- Dynamic Programming

1D Optimization

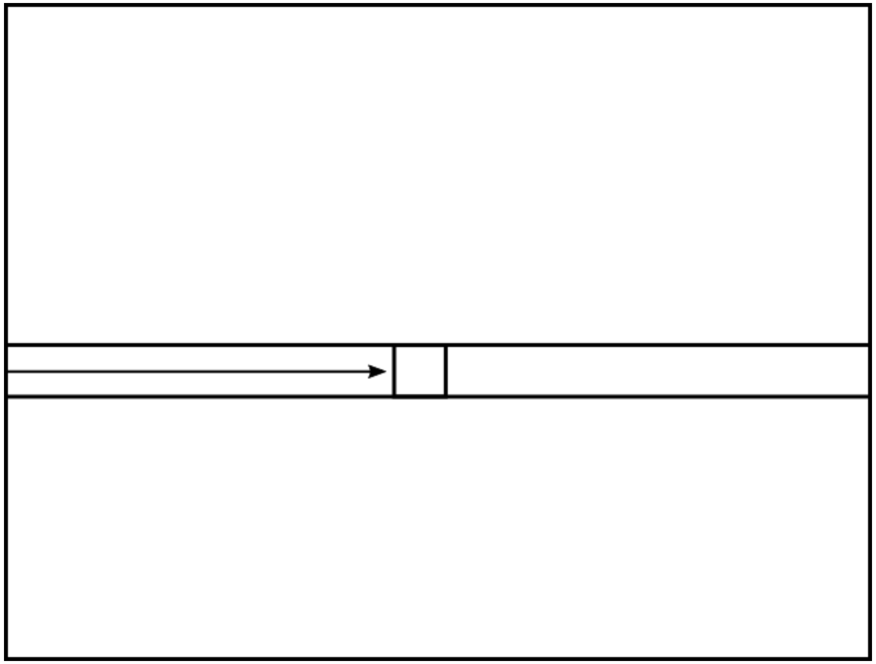
- Dynamic Programming
- Minimize Cost along Horizontal Lines



1D Optimization

- Dynamic Programming
- Minimize Cost along Horizontal Lines

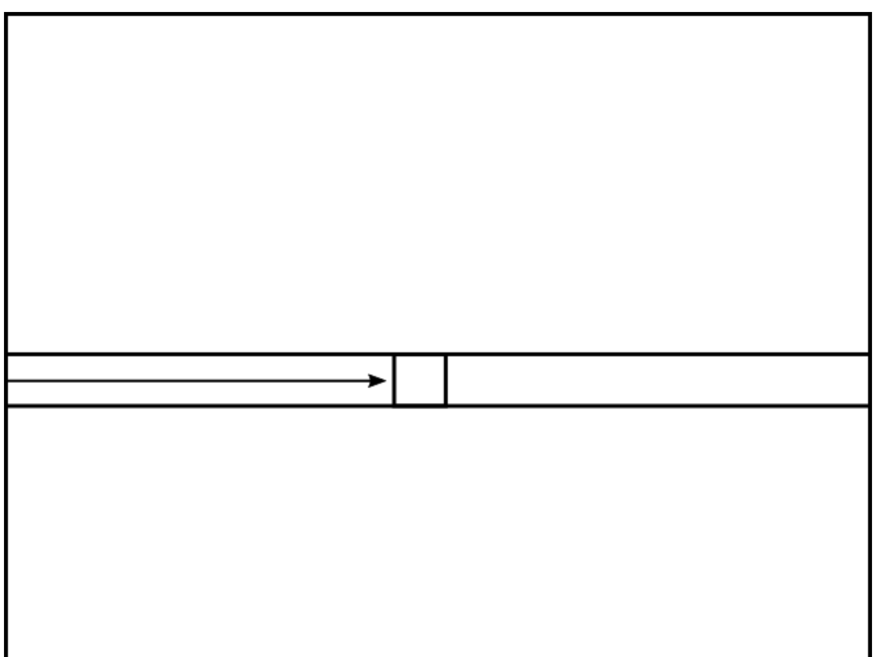
$$\begin{aligned} S(\boldsymbol{p}, d) &= C(\boldsymbol{p}, d) \\ &+ \min \{ \quad S(\boldsymbol{p}-1, d), \\ &\qquad S(\boldsymbol{p}-1, d-1) + \mathcal{P}_1, \\ &\qquad S(\boldsymbol{p}-1, d+1) + \mathcal{P}_1, \\ &\qquad \min_i S(\boldsymbol{p}-1, i) + \mathcal{P}_2 \quad \} \end{aligned}$$



1D Optimization

- Dynamic Programming
- Minimize Cost along Horizontal Lines

$$
\begin{aligned}
S(\boldsymbol{p}, d) &= C(\boldsymbol{p}, d) \\
&\quad + \min \{ \quad S(\boldsymbol{p}-1, d), \\
&\qquad\qquad S(\boldsymbol{p}-1, d-1) + \mathcal{P}_1, \\
&\qquad\qquad S(\boldsymbol{p}-1, d+1) + \mathcal{P}_1, \\
&\qquad\qquad \min_i S(\boldsymbol{p}-1, i) + \mathcal{P}_2 \quad \}
\end{aligned}
$$



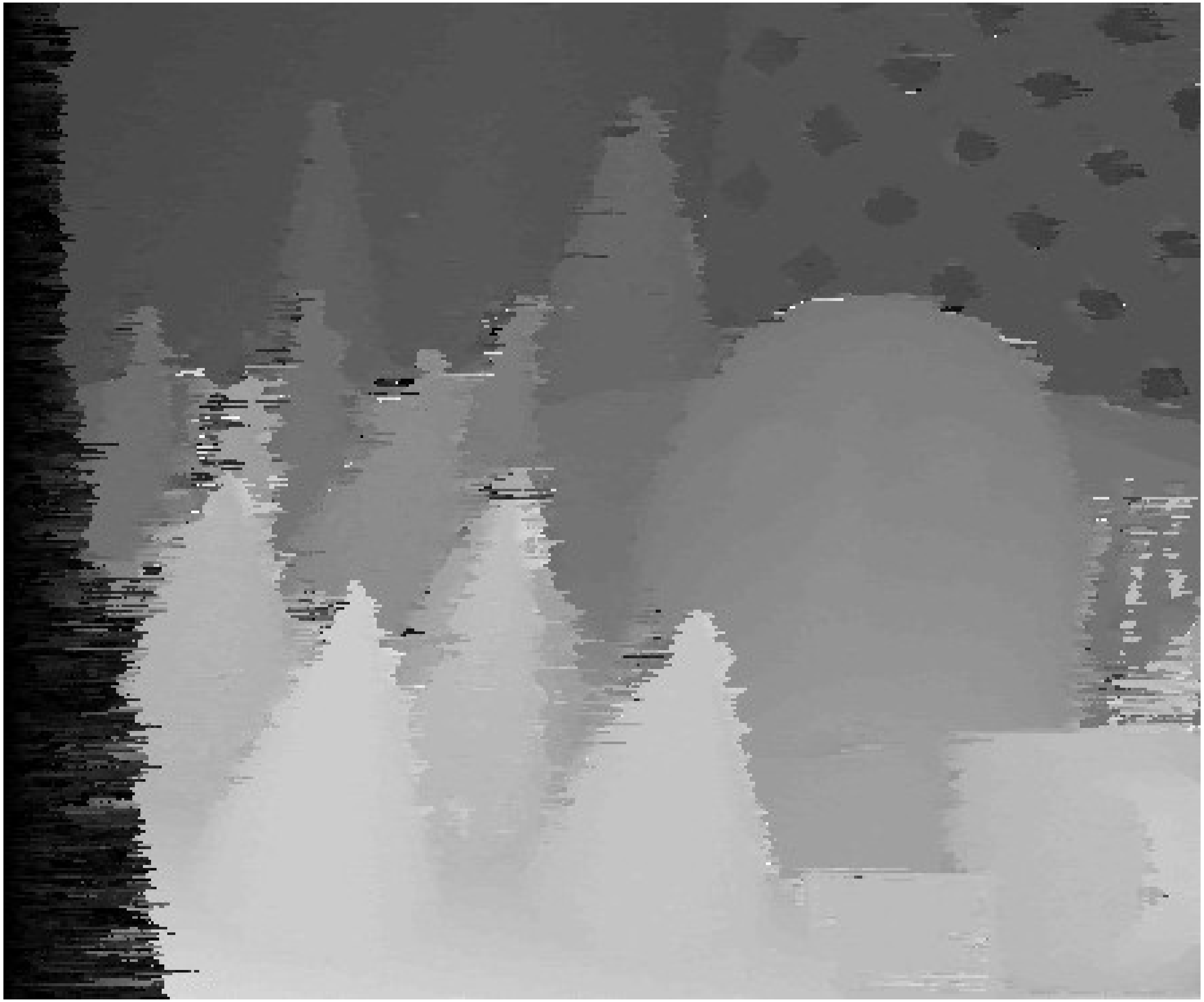
$$D_{\boldsymbol{p}} = \text{argmin}_d S(\boldsymbol{p}, d)$$

Streaking Issues

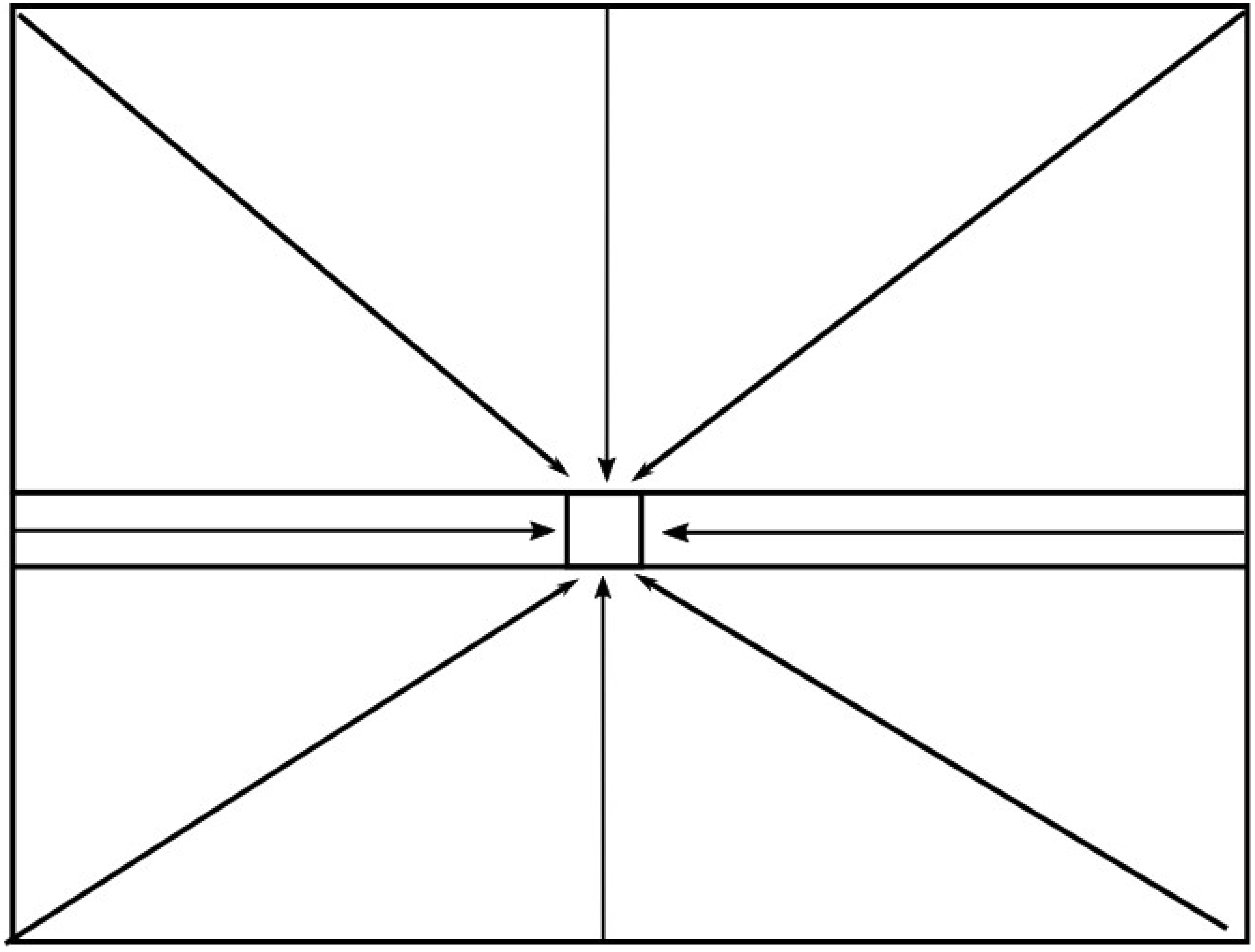
Left Image

Right Image

Streaking Issues

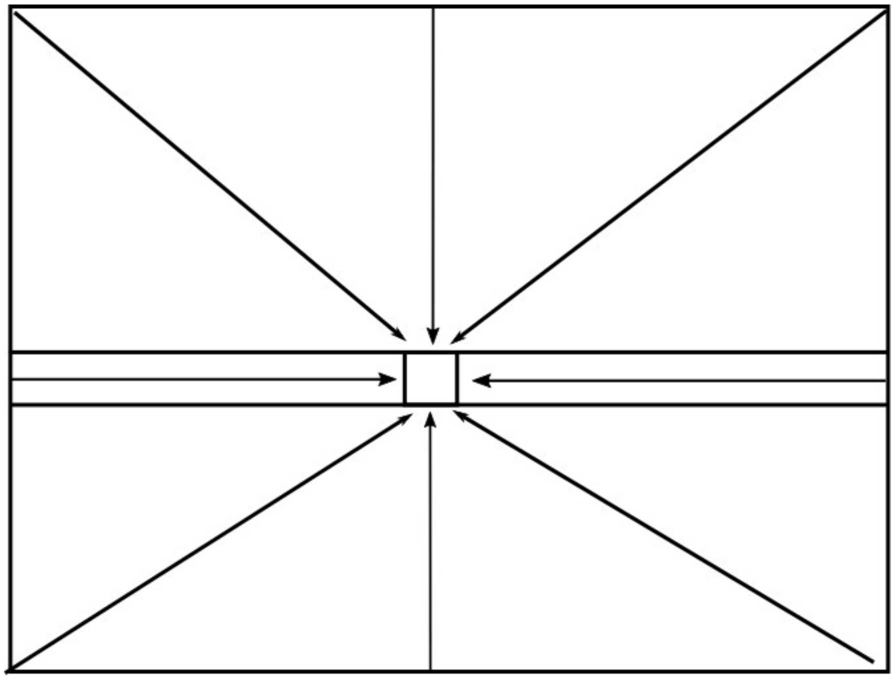


Use Multiple Directions



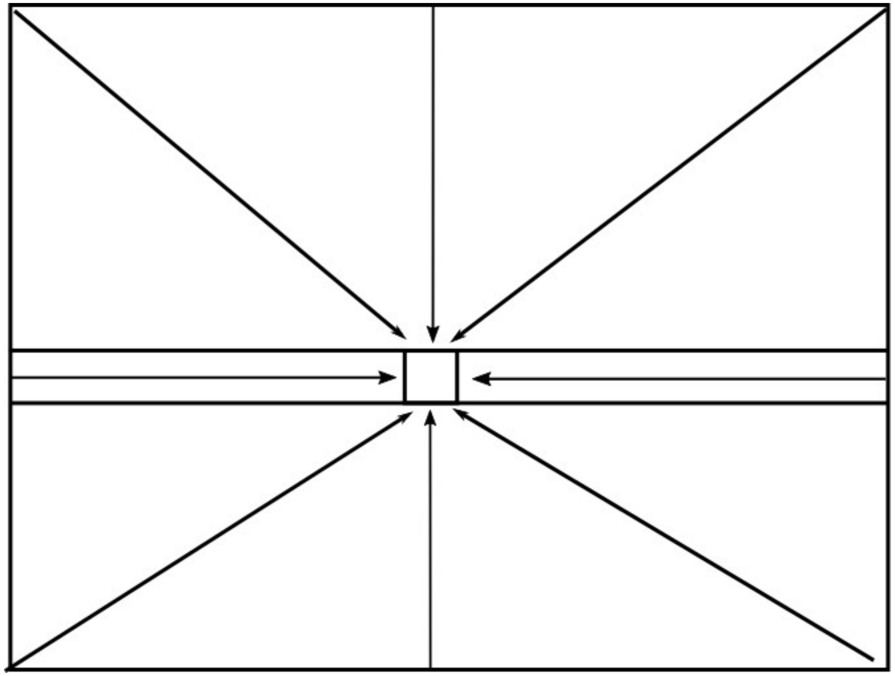
Use Multiple Directions

$$
\begin{aligned}
L_r(\boldsymbol{p}, d) \quad = C(\boldsymbol{p}, d) \\
+ \min \{ \quad & L_r(\boldsymbol{p} - r, d), \\
& L_r(\boldsymbol{p} - r, d - 1) + \mathcal{P}_1, \\
& L_r(\boldsymbol{p} - r, d + 1) + \mathcal{P}_1, \\
& \min_i L_r(\boldsymbol{p} - r, i) + \mathcal{P}_2 \qquad \}
\end{aligned}
$$



Use Multiple Directions

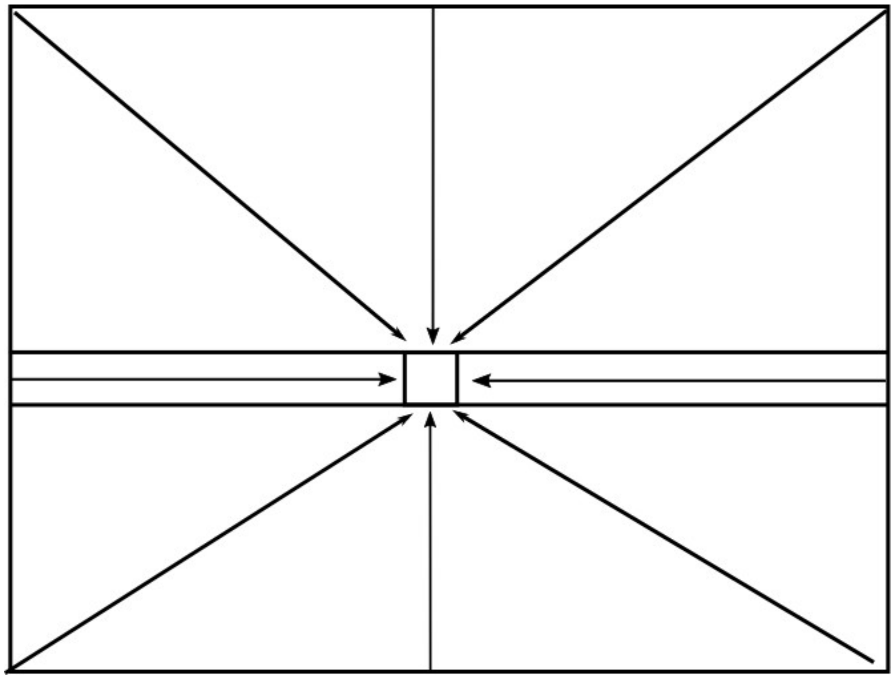
$$L_r(\boldsymbol{p}, d) = C(\boldsymbol{p}, d) + \min\{ L_r(\boldsymbol{p} - r, d), L_r(\boldsymbol{p} - r, d-1) + \mathcal{P}_1, L_r(\boldsymbol{p} - r, d+1) + \mathcal{P}_1, \min_i L_r(\boldsymbol{p} - r, i) + \mathcal{P}_2 \}$$



r *varies over 8 directions*

Use Multiple Directions

$$
\begin{aligned}
L_r(\boldsymbol{p}, d) \quad = C(\boldsymbol{p}, d) \\
+ \min \{ \quad & L_r(\boldsymbol{p} - r, d), \\
& L_r(\boldsymbol{p} - r, d-1) + \mathcal{P}_1, \\
& L_r(\boldsymbol{p} - r, d+1) + \mathcal{P}_1, \\
& \min_i L_r(\boldsymbol{p} - r, i) + \mathcal{P}_2 \quad \}
\end{aligned}
$$



r *varies over 8 directions*

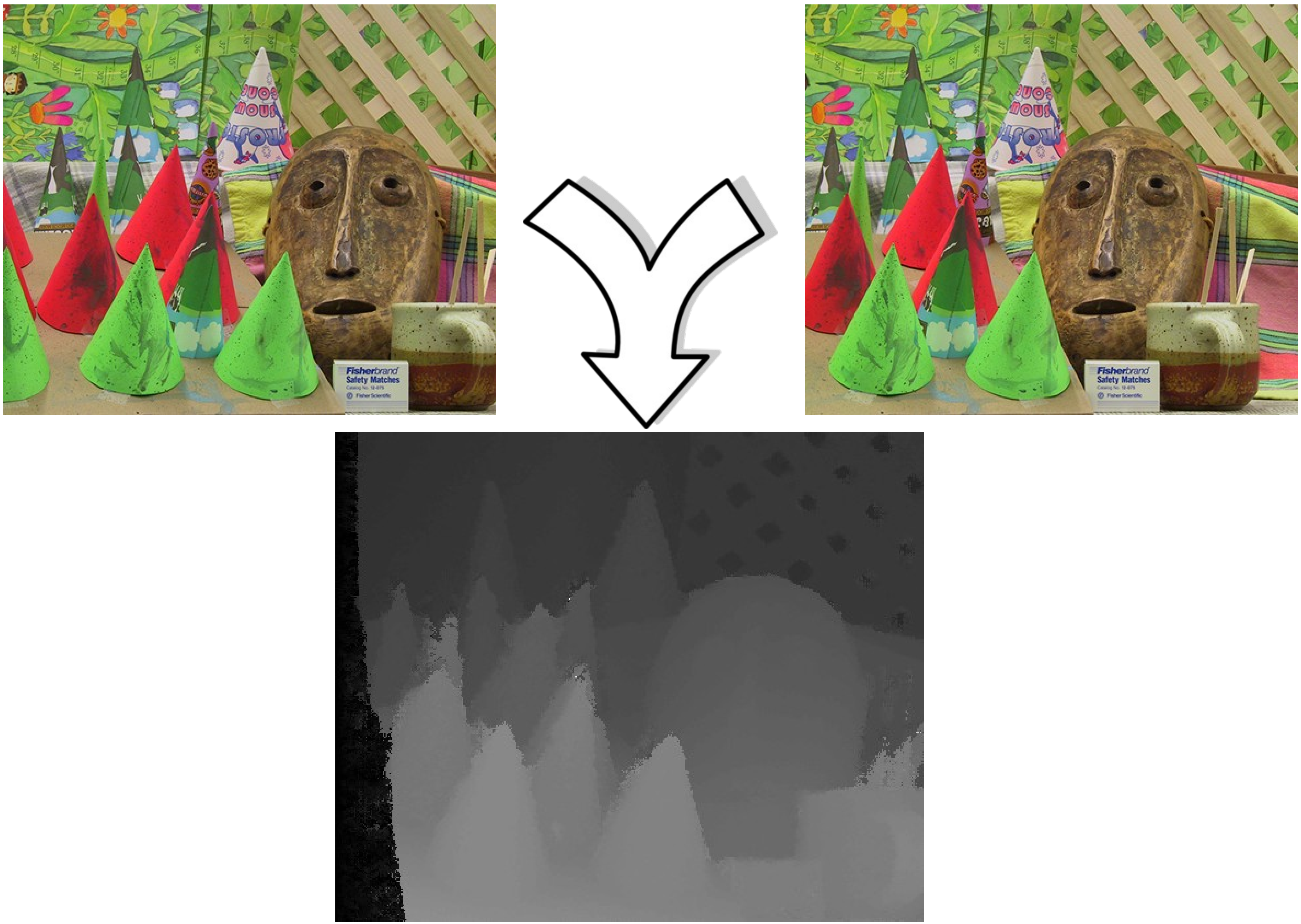
Total Cost : $S(p, d) = \sum_r L_r(\boldsymbol{p}, d)$

Semi Global Matching

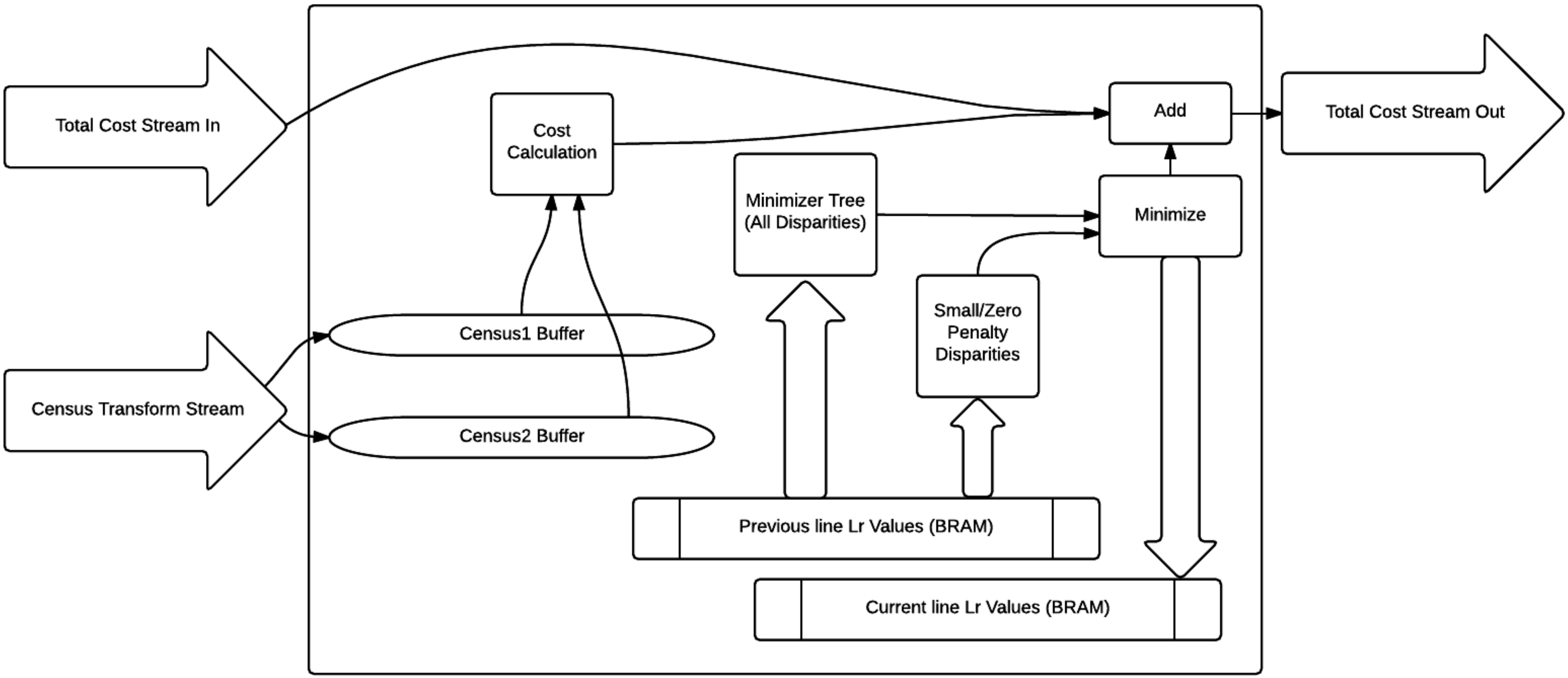
Left Image

Right Image

No Streaking!!!



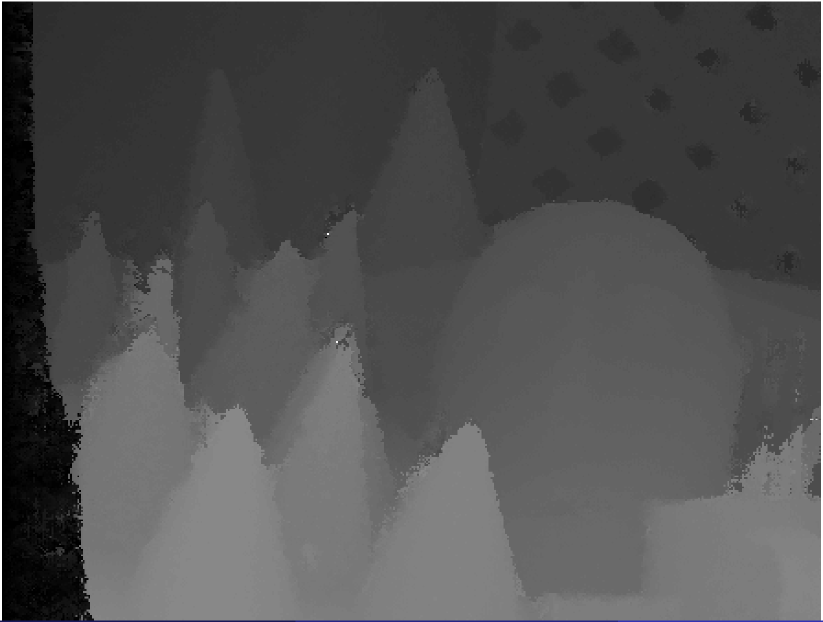
Streaming Design



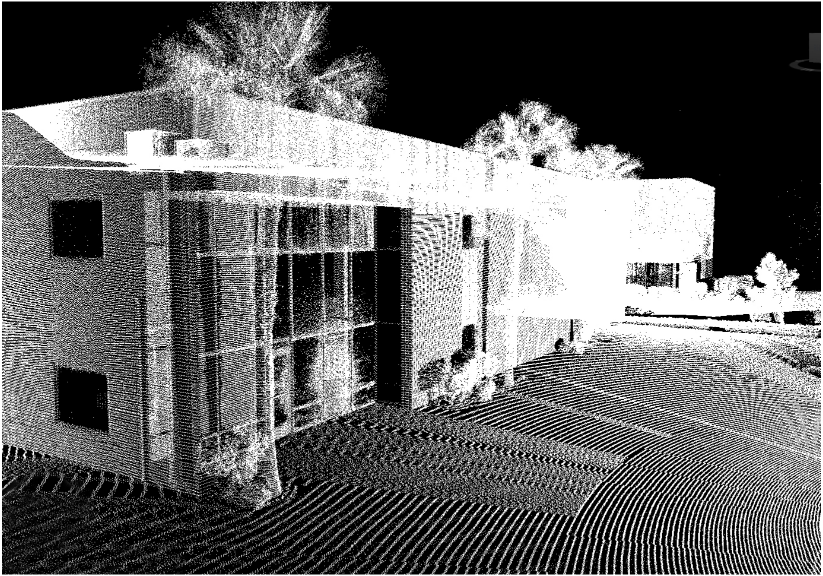
$$
\begin{aligned}
L_r(\boldsymbol{p}, d) \quad &= C(\boldsymbol{p}, d) \\
&+ \min \{ \quad L_r(\boldsymbol{p} - r, d), \\
&\qquad L_r(\boldsymbol{p} - r, d - 1) + \mathcal{P}_1, \\
&\qquad L_r(\boldsymbol{p} - r, d + 1) + \mathcal{P}_1, \\
&\qquad \min_i L_r(\boldsymbol{p} - r, i) + \mathcal{P}_2 \qquad \}
\end{aligned}
$$

Total Cost : $S(p, d) = \sum_r L_r(\boldsymbol{p}, d)$

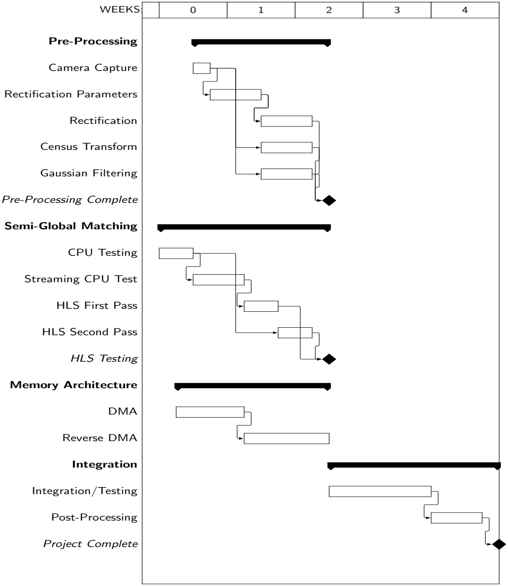
Depth Map Rendering



Point Cloud Rendering



Timeline


WEEKS
0
1
2
3
4
Pre-Processing
Camera Capture
Rectification Parameters
Rectification
Census Transform
Gaussian Filtering
Pre-Processing Complete
Semi-Global Matching
CPU Testing
Streaming CPU Test
HLS First Pass
HLS Second Pass
HLS Testing
Memory Architecture
DMA
Reverse DMA
Integration
Integration/Testing
Post-Processing
Project Complete