3rd IEEE Global Conference on Signal & Information Processing
Orlando, Florida, USA December 14-16 2015

General Symposium:
WbGS-L: Image Enhancement and Calibration Techniques – 1.

Fast Depth Estimation Using Non-iterative Local Optimization for Super Multi-view Images

T. Senoh, K. Wakunami, H. Sasaki, R. Oi and K. Yamamoto

Dec. 16, 2015

National Institute of Information and Communications Technology

Contents

- Introduction

  Depth estimation ⇒ super multi-view coding and free navigation

  Problem ⇒ depth estimation speed and quality

- Fast depth estimation algorithm

  3-view matching ⇒ against occlusion

  Non-iterative depth optimization:

  - Depth continuity weight ⇒ against pseudo matching
  - zig-zag scan ⇒ for speed and depth uniformity
  - edge adaptation ⇒ for depth reliability

- Experimental results

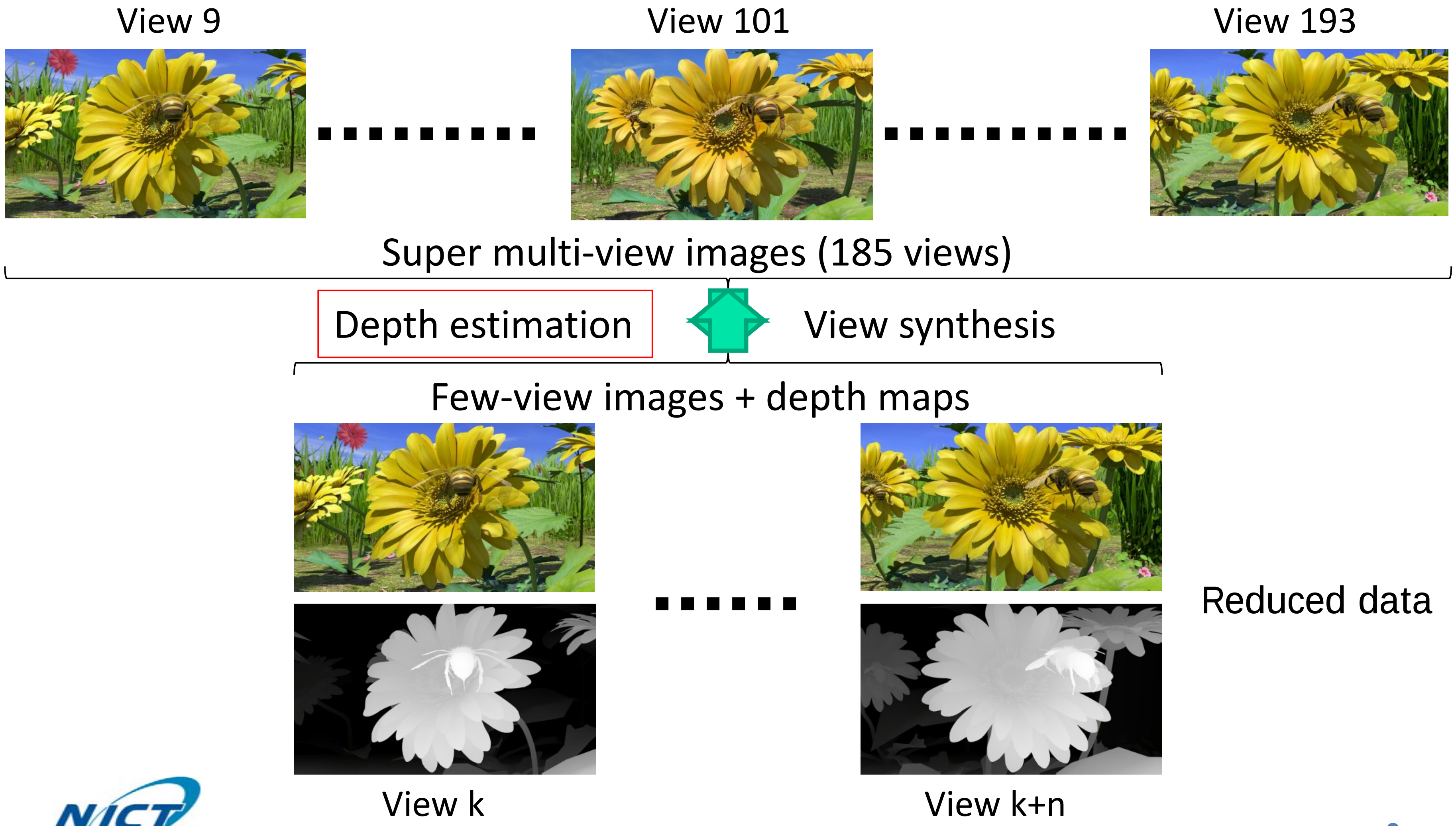
  CG image：68-100 times fast / synthesized image gain = +0.3 to -0.3dB

  Camera image：111 times fast / synthesized image loss = -0.7to -1.0dB

- Conclusion

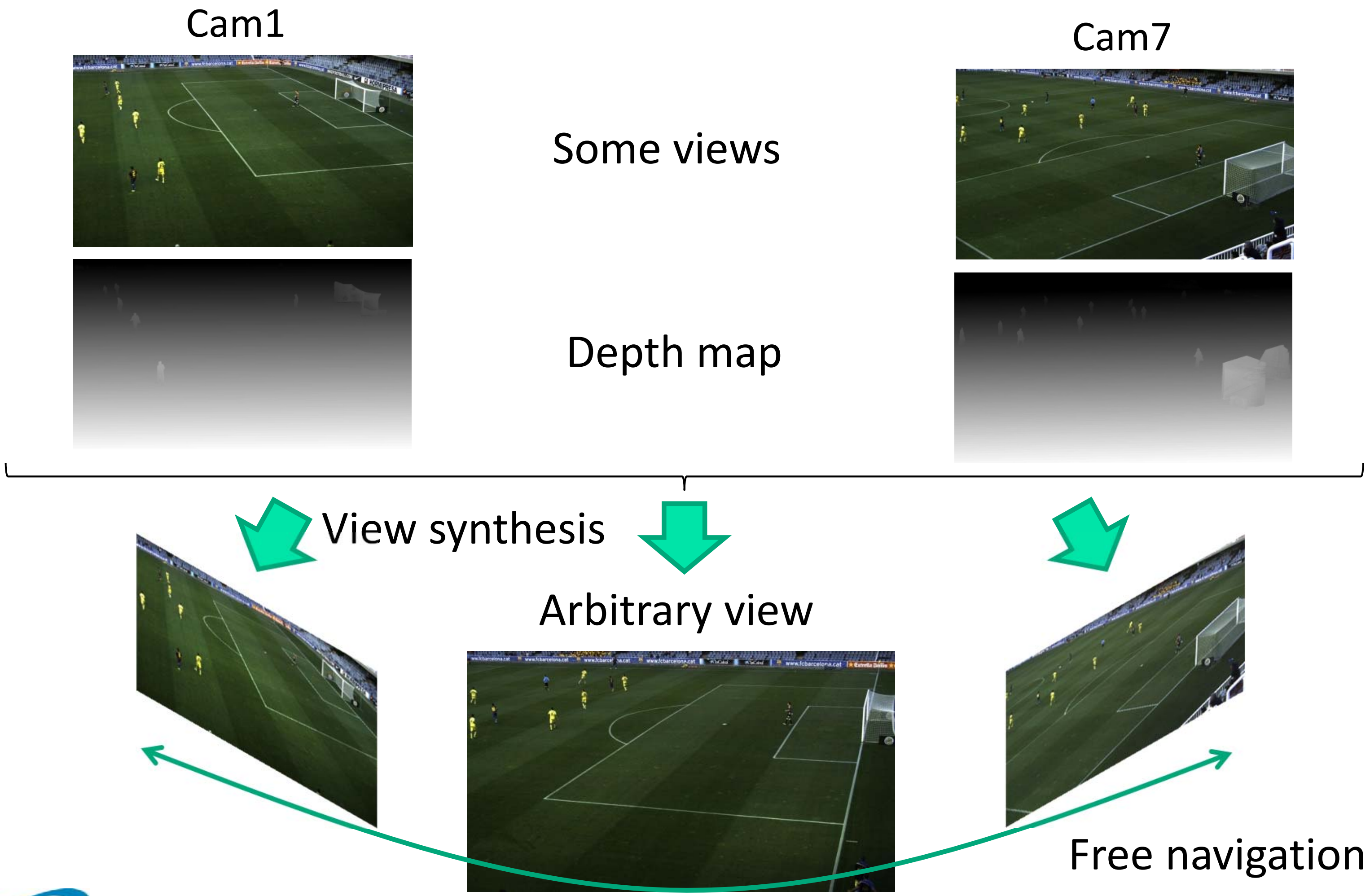
Super multi-view image coding

- If depth maps are available, number of views can be reduced.



Free navigation

- If depth maps are available, arbitrary views can be synthesized.

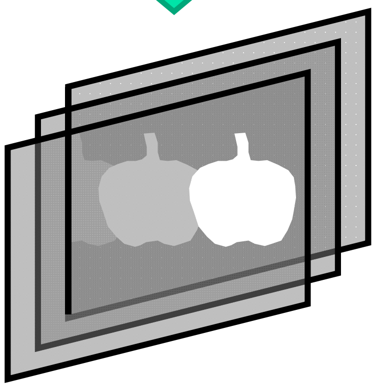


Depth estimation from multi-view image

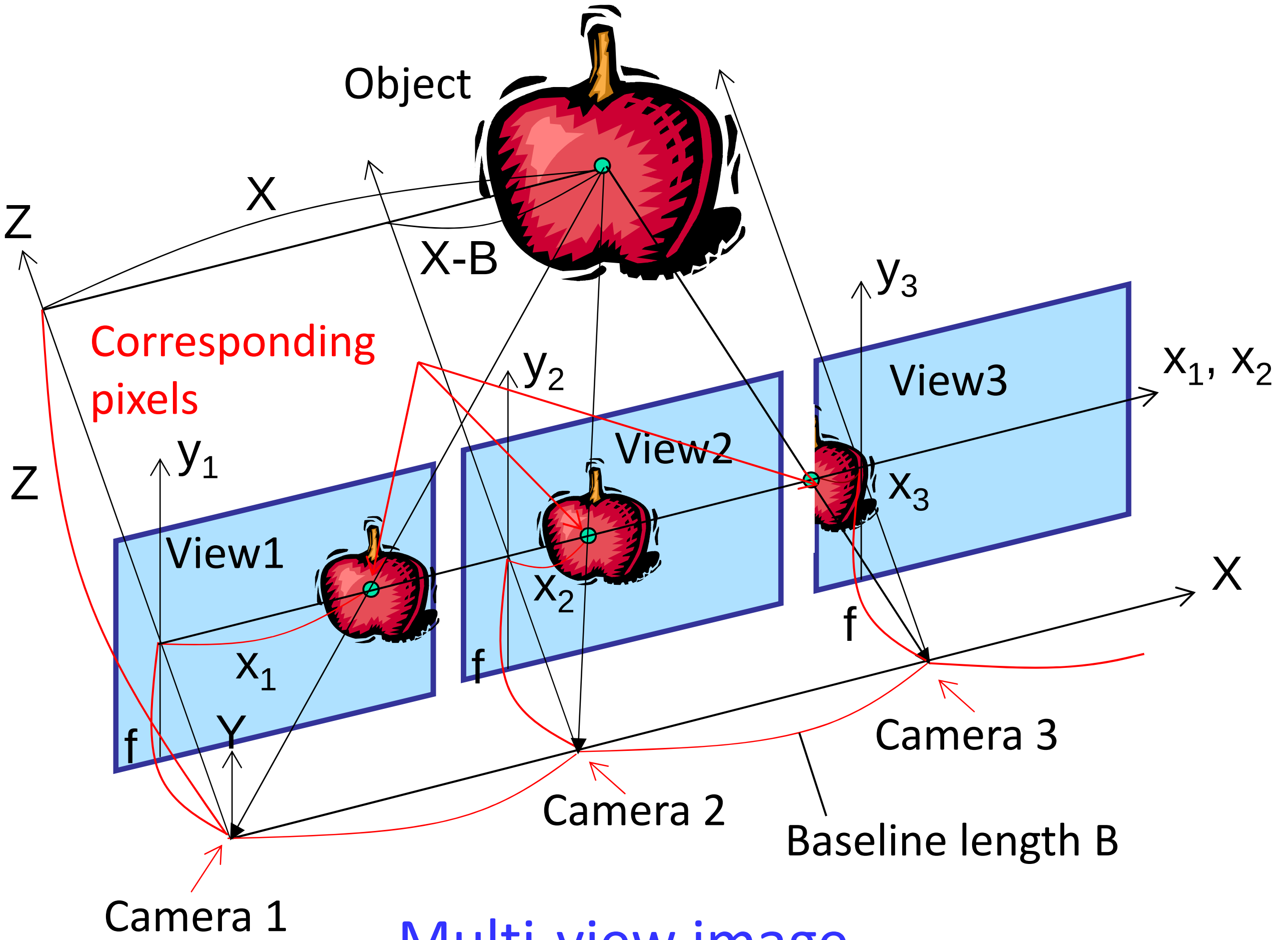
- Depth is the distance between corresponding pixels (disparity).
- Corresponding pixels are searched by stereo matching.

Depth = x_1-x_2 = x_2-x_3 = fB/Z

Stereo matching



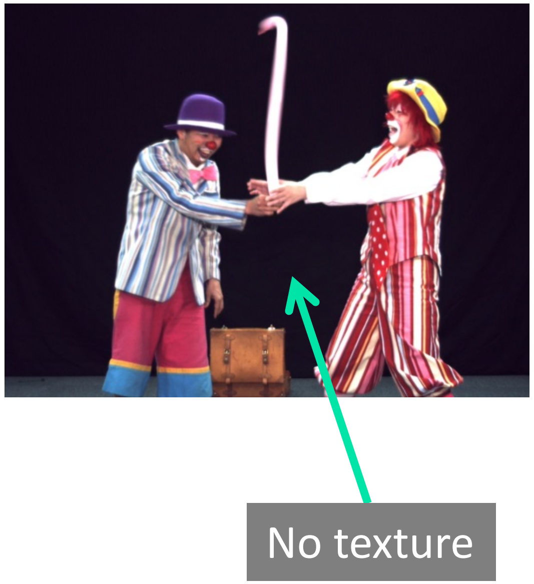
Depth map

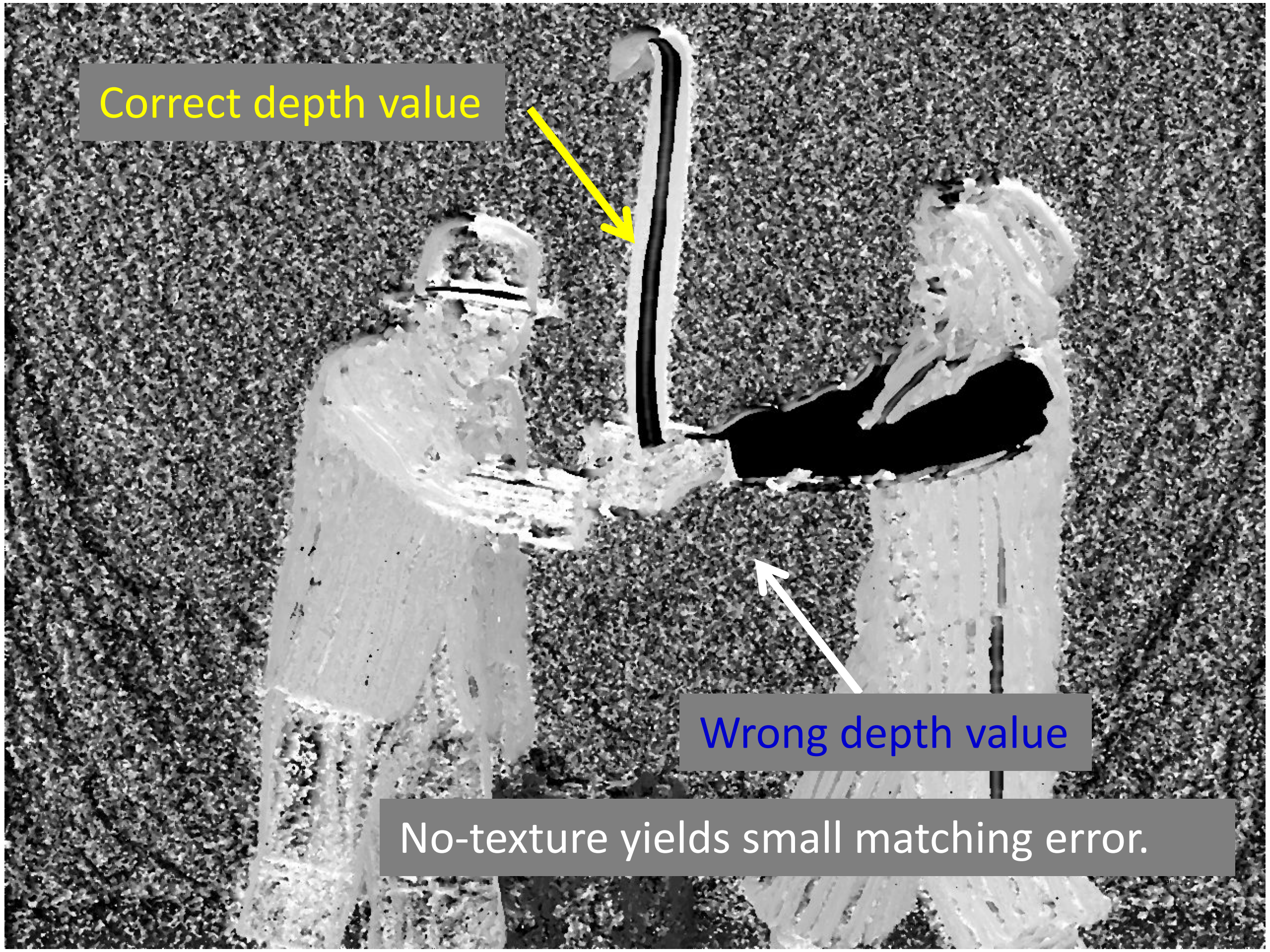


Multi-view image

Depth map by stereo matching only

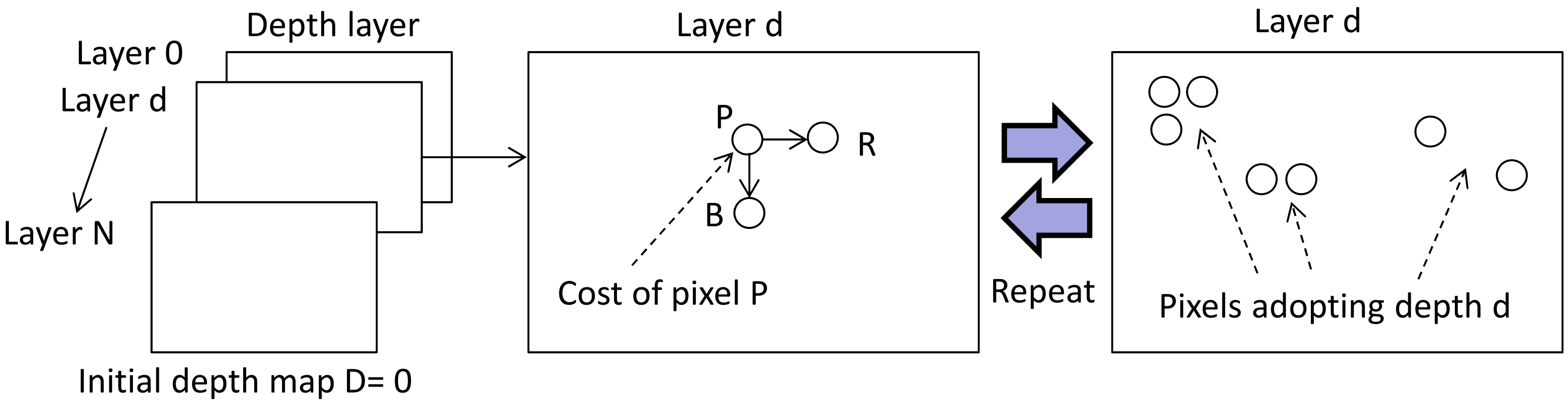
- Problem : Depth errors caused by pseudo matching.
- Point : Depth at texture edge is correct. ⇒ Propagate correct depth value.





Graph-cuts smoothing

- Graph-cuts algorithm repeats cost evaluation until the cost converges.
- ⇒ slow, un-predictable estimation time.



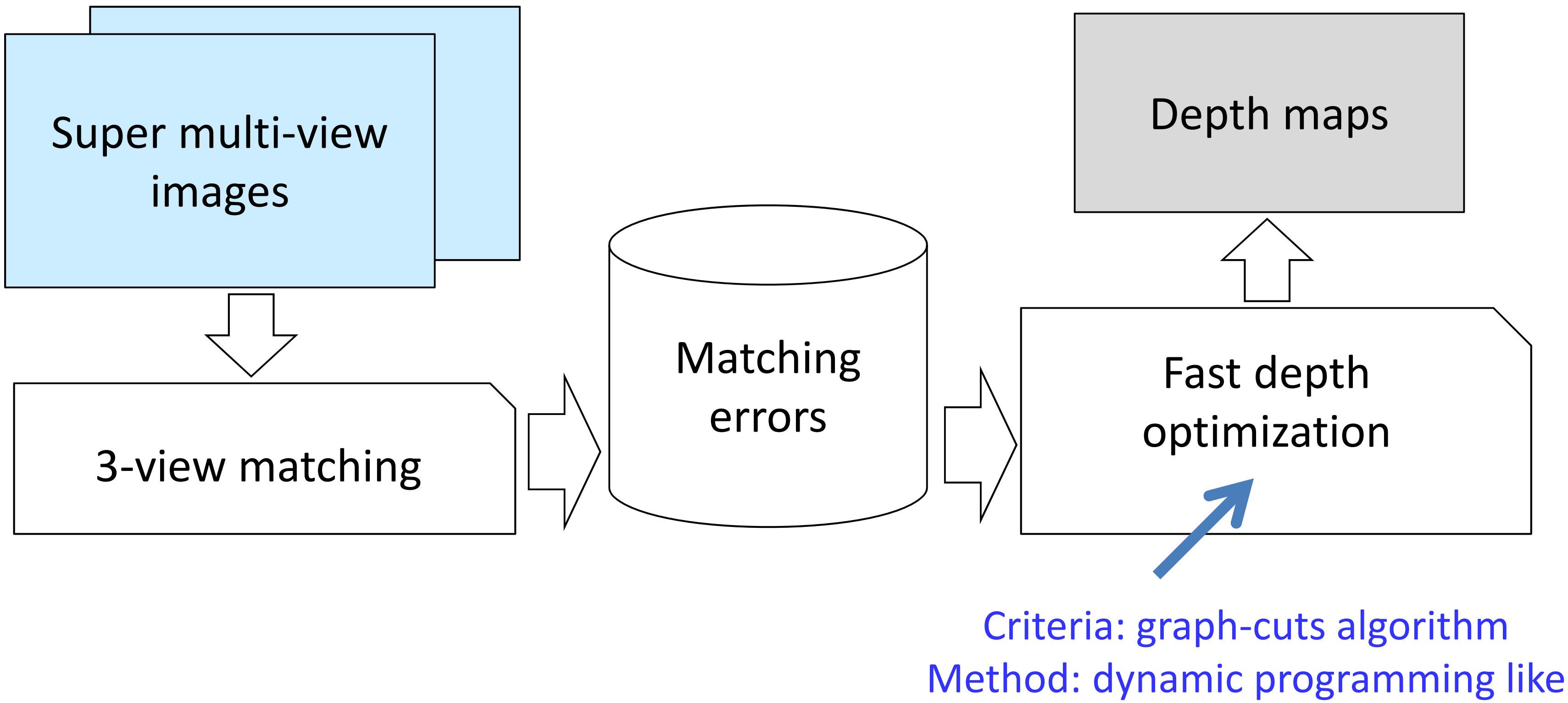
New cost $$C(d,P) = \underbrace{E(d,P)}_{\text{Matching error}} + \underbrace{\lambda(|D(R)-d| + |D(B)-d|)}_{\text{Depth continuity}}$$

Old cost $$C(D,P) = E(D,P) + \lambda(|D(R)-D(P)| + |D(B)-D(P)|)$$

Criteria $$\text{if } \sum_P C(d,P) < \sum_P C(D,P), \rightarrow \textit{replace } D \textit{ with } d$$

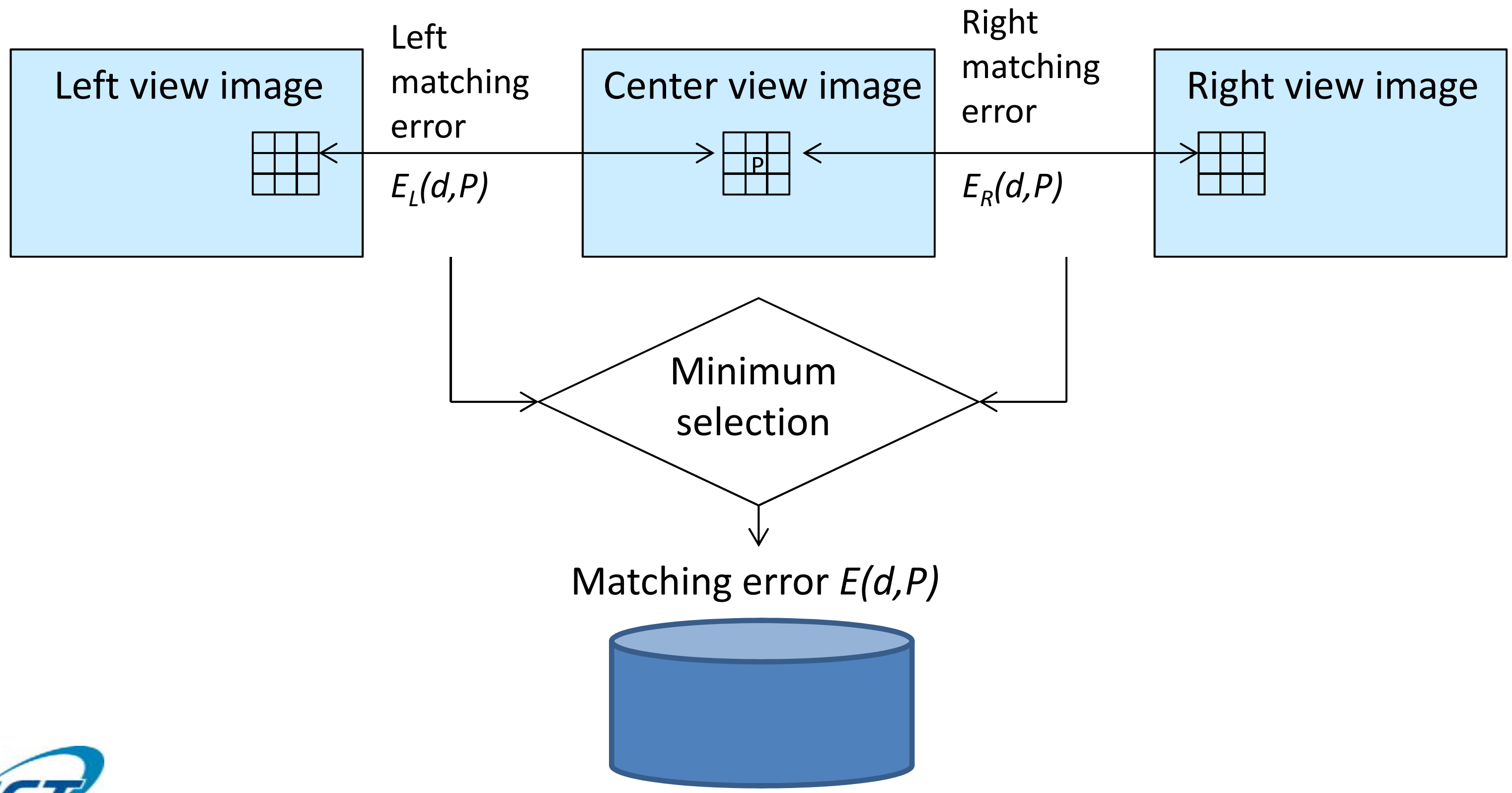
Fast depth estimation proposal

- Accelerate graph-cuts algorithm by means of dynamic programming like.



3-view matching error aggregation

- By changing depth value *d* from minimum to maximum, at each pixel *P*,
- Record smaller left or right matching error (E_L or E_R). ⇒ against occlusion
- 3x3 pixel matching ⇒ against pseudo matching

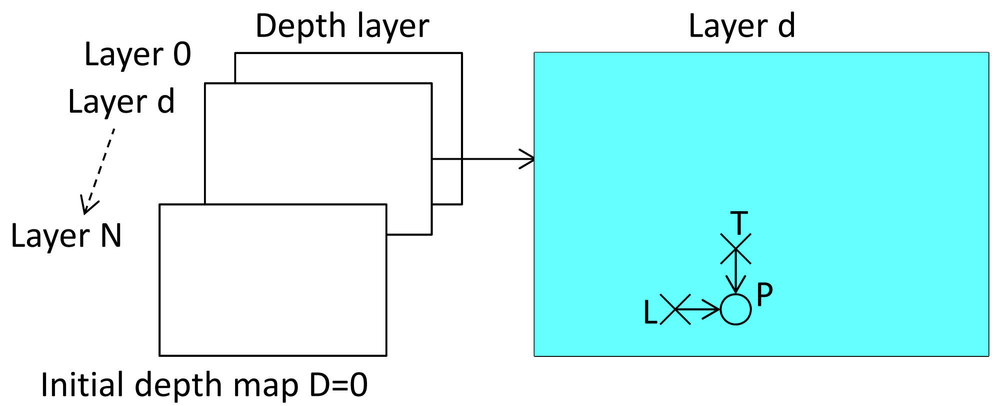


Non-iterative cost evaluation

- Scan from depth layer 0 to layer N,
- In depth layer d, start from center line ⇒ get depth of important object.
- Zig-zag scan pixels. ⇒ Reduce depth over-propagation.
- Use adopted depth values at pixel *T* and *L* for the next pixel *P*.
- Evaluate *C(d, P):* Cost of current depth *d*

  C(D,P): Cost of already determined depth *D*

  If *C(d, P)*＜*C(D, P)*, adopt depth *d* for the pixel *P* temporally.
- When lower half lines are end, scan upper half lines.
- After one-layer scan, repeat scan in reverse direction. ⇒ smooth depth map.



Cost function

- Cost = matching error + weight × depth continuity

$$\underbrace{C(d,P)}_{\text{New cost}} = \underbrace{E(d,P)}_{\text{Matching error}} + \overset{\text{Horizontal weight}}{\lambda_H} \, |\underbrace{D(L)-d}_{\text{Horizontal continuity}}| + \overset{\text{Vertical weight}}{\lambda_V} \, |\underbrace{D(T)-d}_{\text{Vertical continuity}}| + \overset{\text{Hysteresis weight}}{\delta} \, |\underbrace{D(P)-d}_{\text{Hysteresis}}|$$

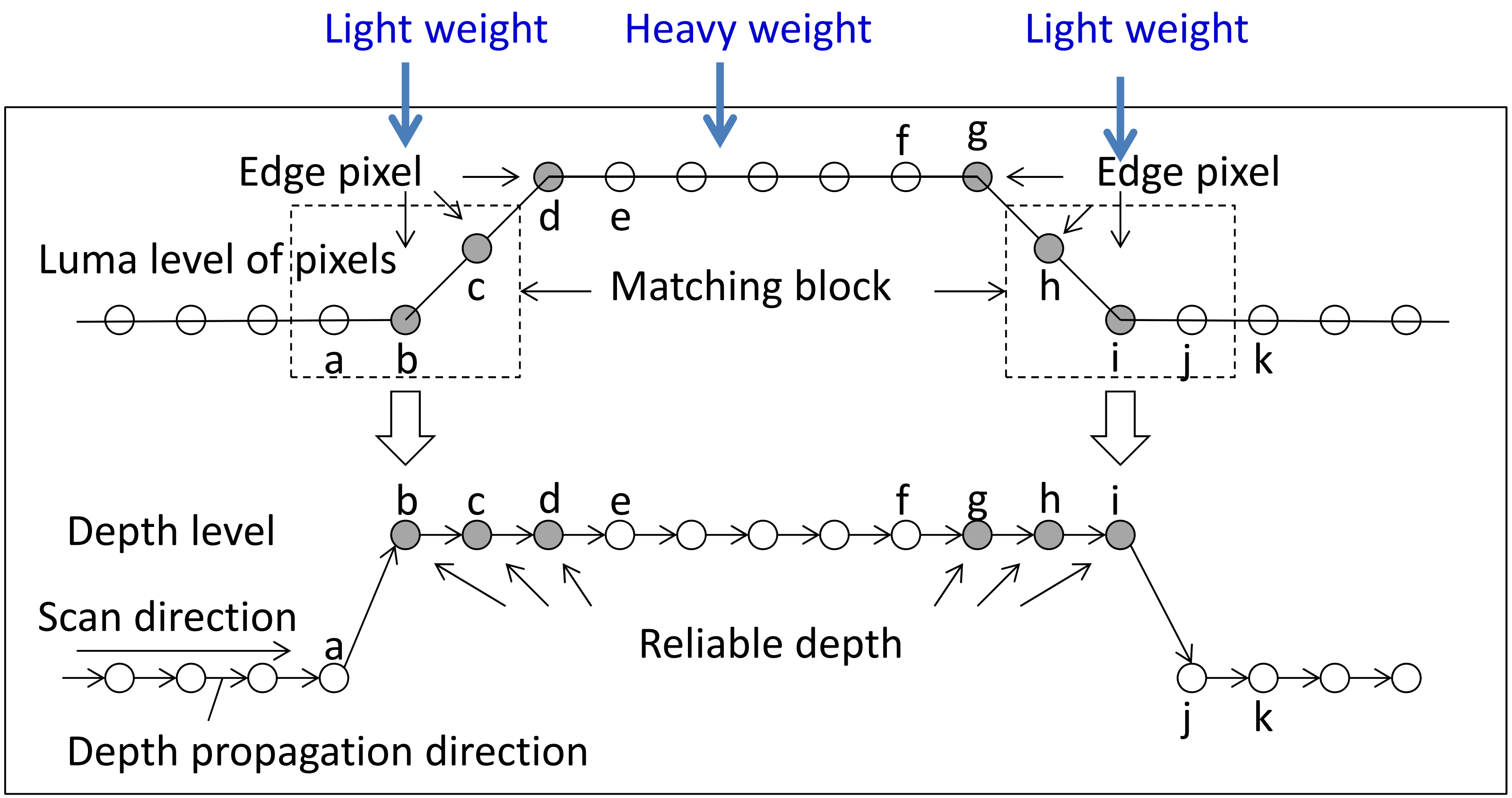
$$\underbrace{C(D,P)}_{\text{Previous cost}} = \underbrace{E(D,P)}_{\text{Matching error}} + \lambda_H \, |\underbrace{D(L)-D(P)}_{\text{Horizontal continuity}}| + \lambda_H \, |\underbrace{D(T)-D(P)}_{\text{Vertical continuity}}|$$

$$\text{if } \underbrace{C(d,P) < C(D,P),}_{\text{No summation}} \rightarrow \text{adopt } \underbrace{d}_{\text{New depth}}$$

Correct depth propagation

- Propagate depth value at large luminance change to other area.

  ⇒ Change depth continuity weight when edge exists.



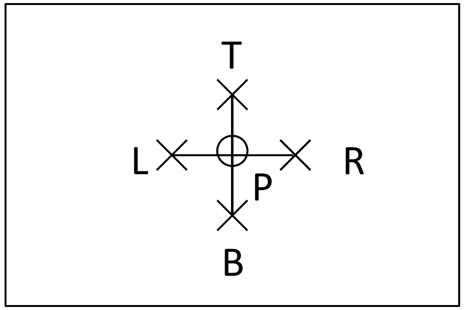
Edge adaptation

- If luma difference between left pixel L and right pixel R, or top pixel T and bottom pixel B > threshold,
- ⇒ edge ⇒ Multiply reduction coefficient ρ to depth continuity weight λ.
- Not using center pixel ⇒ increases detection sensitivity for blurred edges.

$$if\ \left|I(L)-I(R)\right|>th\quad \Rightarrow horizontal\ \ edge\quad \Rightarrow \lambda_H=\lambda_H\times\rho$$

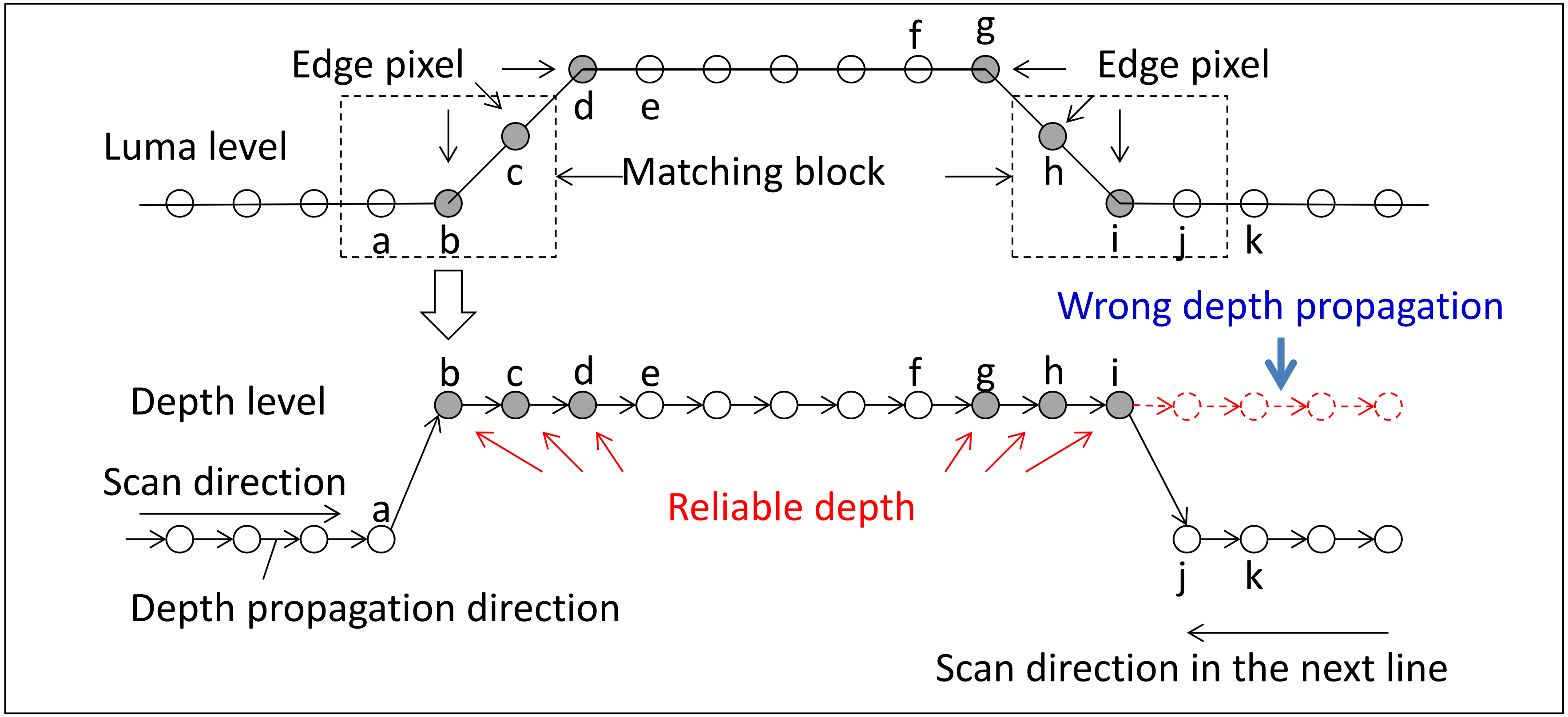
$$if\ \left|I(T)-I(B)\right|>th\quad \Rightarrow vertical\ \ edge\quad \Rightarrow \lambda_V=\lambda_V\times\rho$$

Center view image



Depth level after falling edge

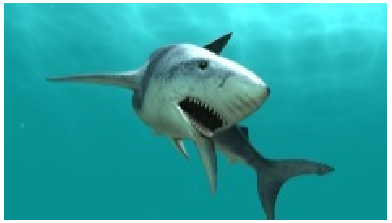
- If background is less textured, preventing foreground depth propagation is difficult.
- To ease such propagation, change the scan direction line by line.



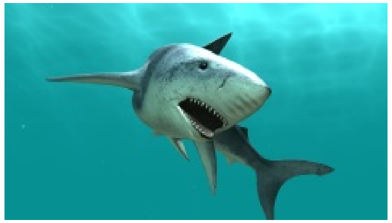
Test image for experiment

CG image:

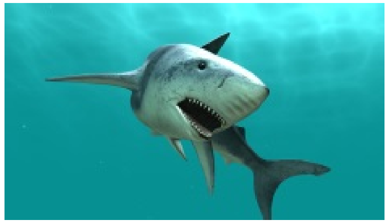
Shark



View99



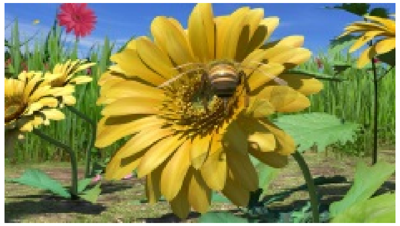
View100



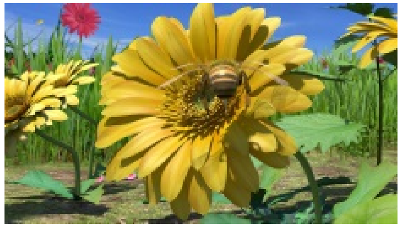
View101

(1920 × 1088)

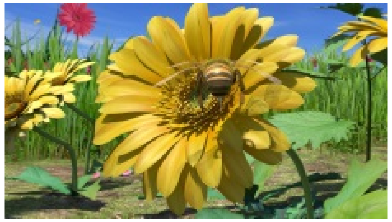
Bee



View11



View10



View9

(1920 × 1088)

Camera image:

Champagne Tower

View34

View35

View36

(1280 × 960)

Pantomime

View49

View50

View51

(1280 × 960)

NICT

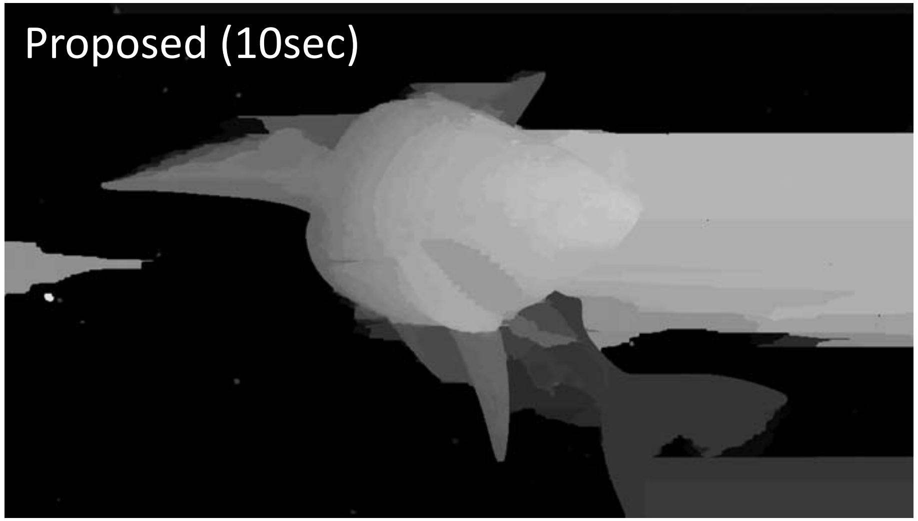
Fast depth estimation parameter

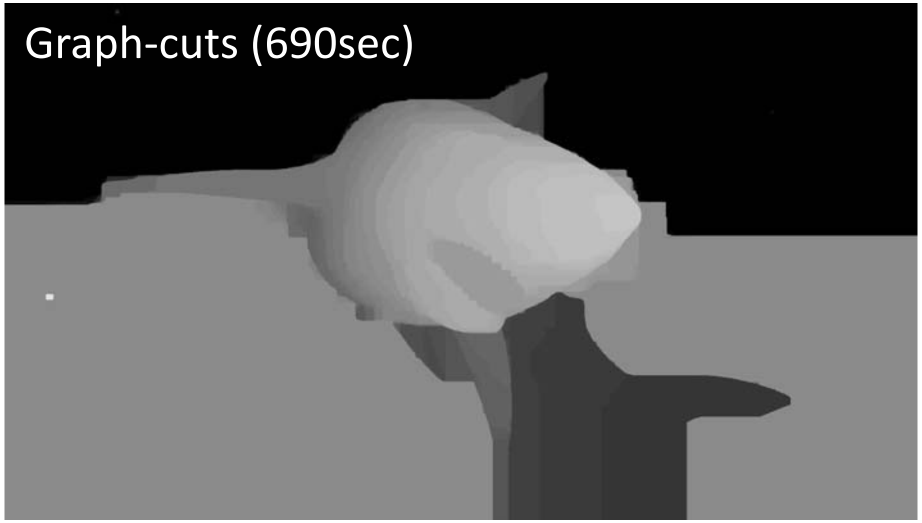
- Parameters are currently determined empirically.

Item	Shark	Bee	Champagne	Pantomime
Min Disparity Search Range	-7	-10	0	0
Max Disparity Search Range	5	3	28	15
Search Level	4	4	4	4
Smoothing Coef 1 (λ)	1.0	1.0	1.0	2.0
Smoothing Coef 2 (ρ)	0.1	0.1	0.1	0.1
Smoothing Coef 3 (δ)	0.02	0.02	0.01	0.04
δ in case luma <20	0.5	0.5	0.25	1.0
Threshold (th)	10	30	30	20

Shark estimated depth

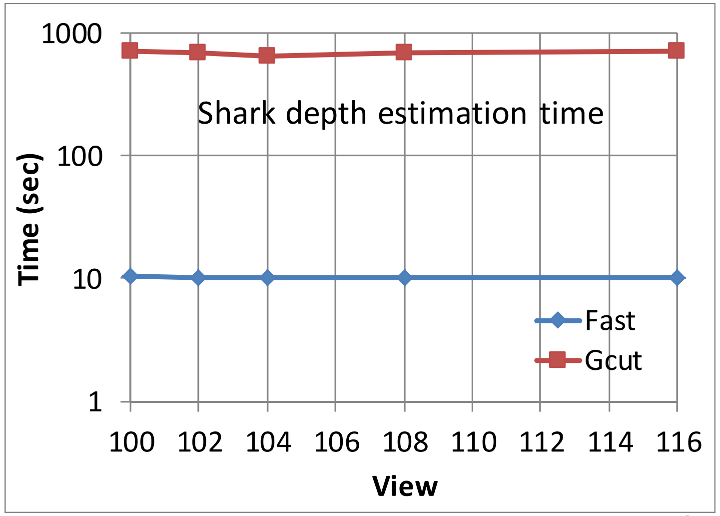
- Depth estimation speed of proposed is 68-times faster than of Graph-Cuts.





Shark: 1920 × 1088 pixels

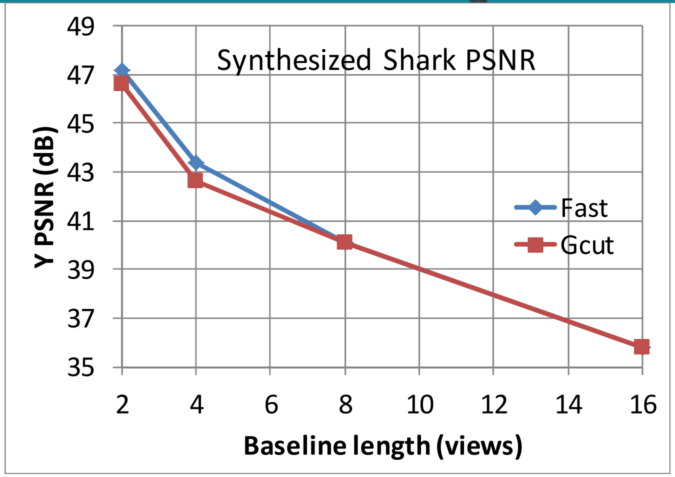
	Proposed	Graph-cuts
Continuity weight	λ=1.0	λ=1.0
Edge weight	ρ=0.1	-
Hysteresis weight	δ=0.02	-
Edge threshold	th=10	-



Synthesized Shark from the depth

- Center view was synthesized from estimated left and right depth maps.
- Average PSNR is +0.3dB better than Graph-cuts result.

Baseline length between left and right views :
2 views
BL = 7mm × 2 views = 14 mm

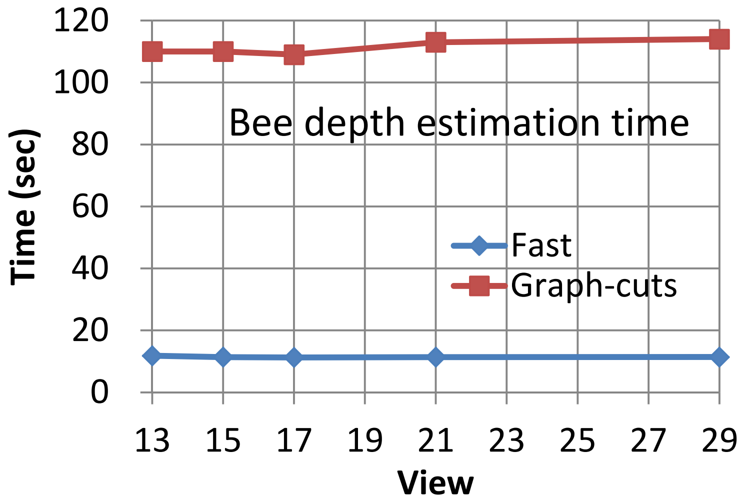


Bee estimated depth

- Depth estimation speed of proposed is 100 times faster than of Graph-Cuts.

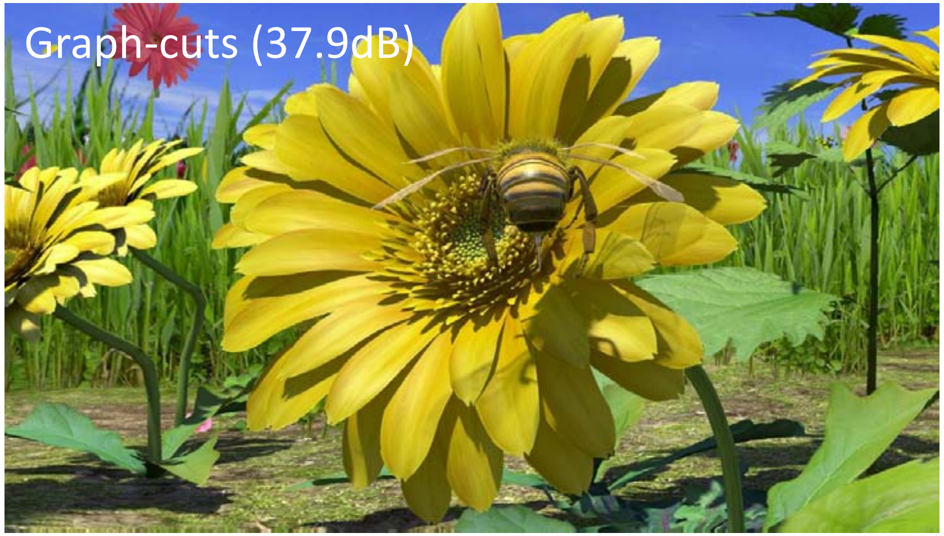
Bee: 1920×1088 pixels

	Proposed	Graph-cuts
Continuity weight	λ=1.0	λ=1.0
Edge weight	ρ=0.1	-
Hysteresis weight	δ=0.02	-
Edge threshold	th=30	-

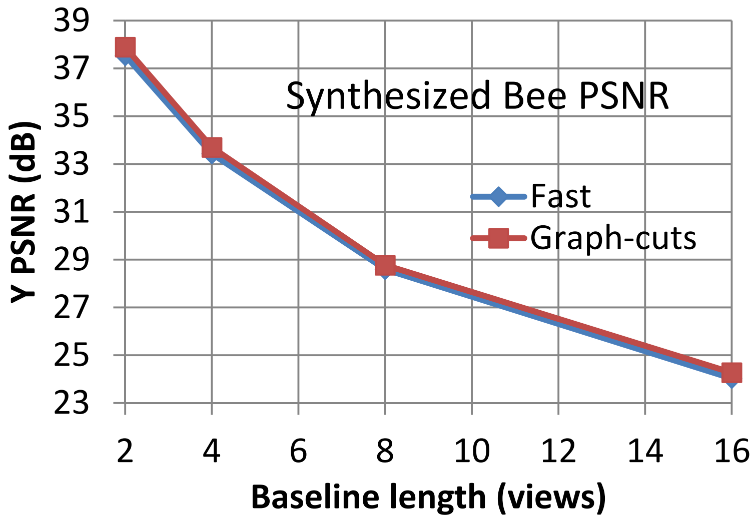


Synthesized Bee from the depth

- Average PSNR is about -0.3dB lower than Graph-cuts result.

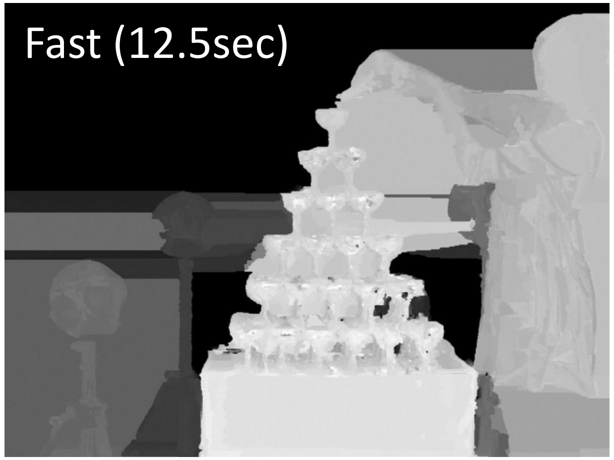


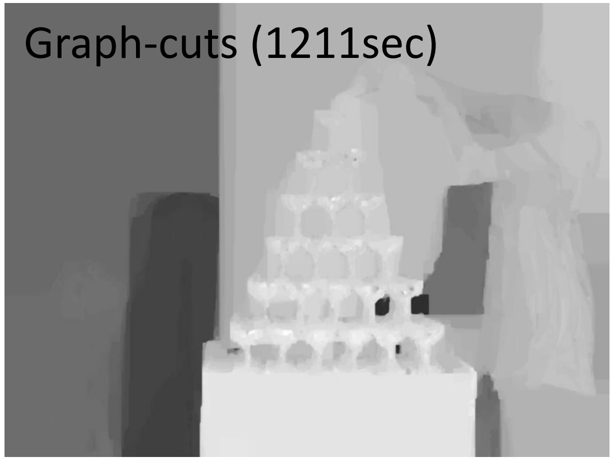
Baseline length between left and right views :
2 views
BL = 3.7mm × 2 views = 7.4 mm



Champagne Tower estimated depth

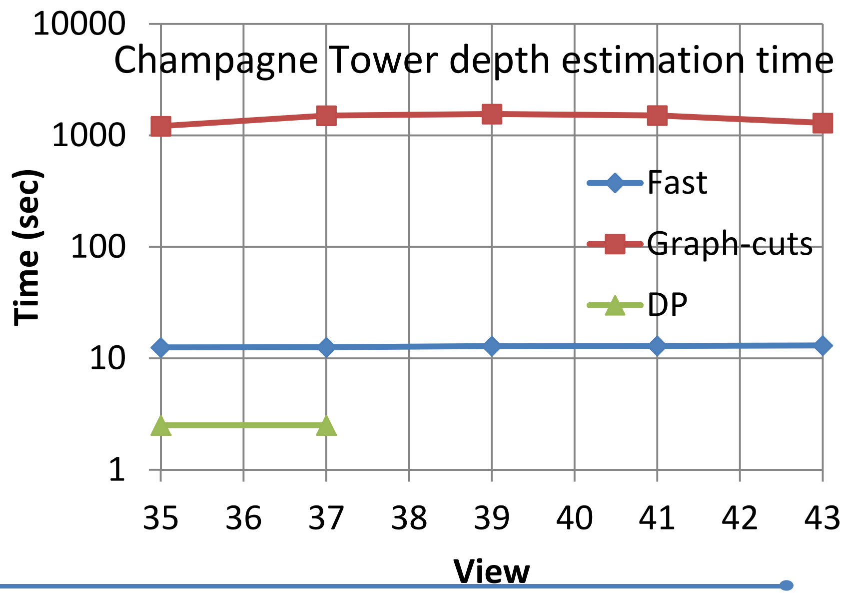
- Depth estimation speed is about 110-times faster than Graph-Cuts.





Champagne Tower: 1280×960 pixels

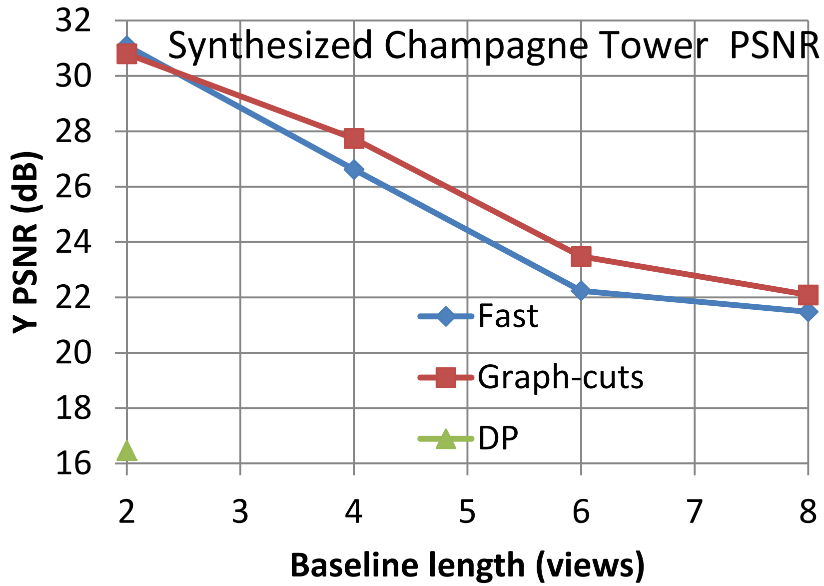
	Proposed	Graph-cuts
Continuity weight	λ=1.0	λ=1.0
Edge weight	ρ=0.1	-
Hysteresis weight	δ=0.01 (δ=0.25/luma<20)	-
Edge threshold	th=30	-



Synthesized Champagne Tower

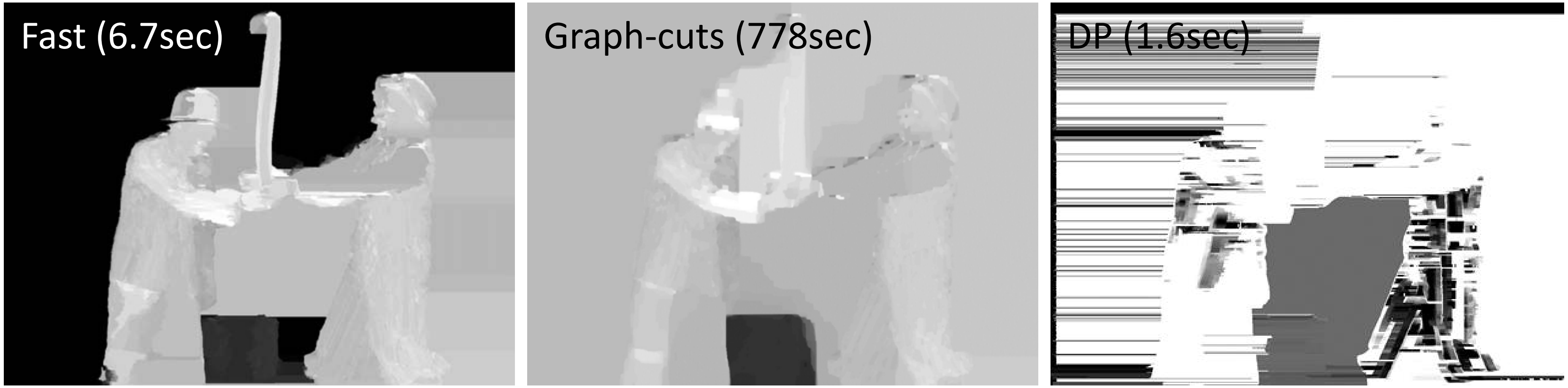
- Average PSNR loss is about -0.7dB compared to Graph-cuts result.

Baseline length between left and right views :
2 views
BL = 50mm × 2 views = 100 mm



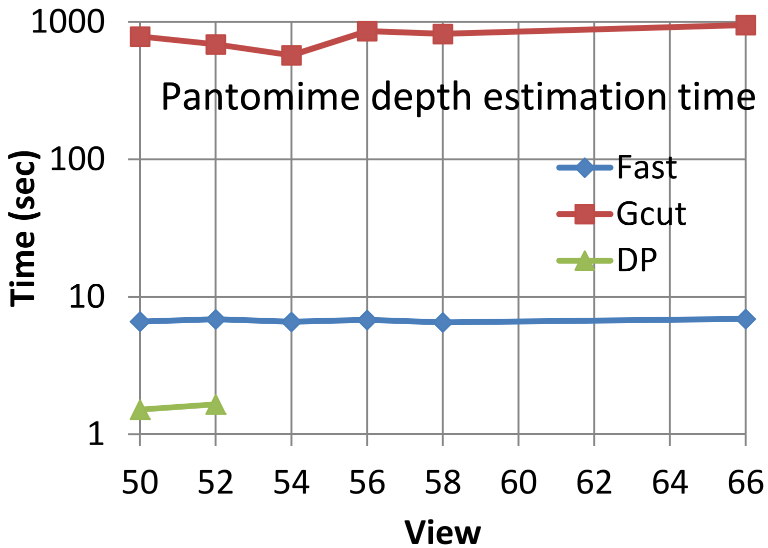
Pantomime estimated depth

- Depth estimation speed is about 116-times faster than Graph-Cuts.



Pantomime: 1280 × 960 pixels

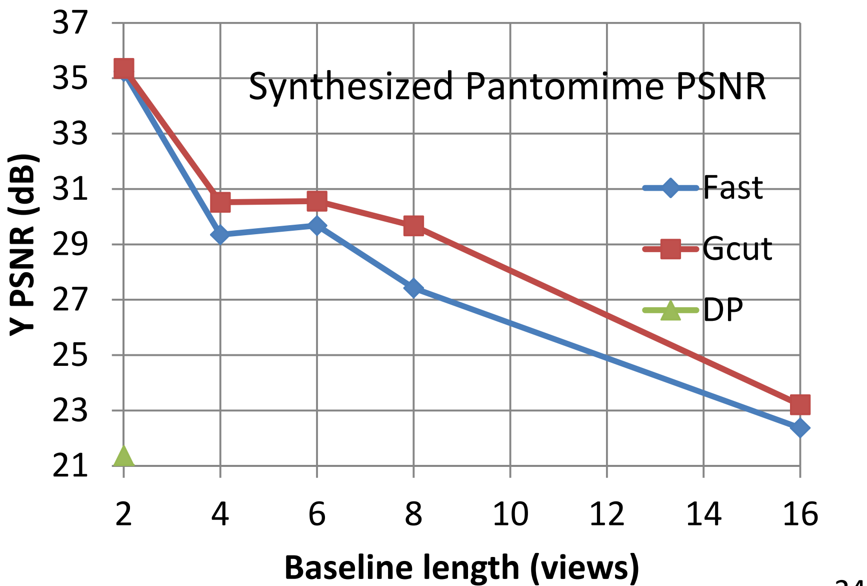
	Proposed	Graph-cuts
Continuity weight	λ=2.0	λ=1.0
Edge weight	ρ=0.1	-
Hysteresis weight	δ=0.04 (δ=1.0/luma<20)	-
Edge threshold	th=20	-



Synthesized Pantomime

- Average PSNR loss is about -1.0 dB compared to Graph-cuts result.

Baseline length between left and right views：
2 views
BL = 50mm × 2 views = 100 mm



Conclusion

- A fast and high-quality depth estimation algorithm was proposed.
- Proposed algorithm accelerated depth estimation speed by 68-116 times faster than a Graph-Cuts algorithm.
- Estimated depth map quality is +0.3 to -1.0dB higher or lower than Graph-Cuts algorithm for super multi-view image synthesis.
- Depth estimation time is determined by the image size and number of depth layers, which is useful for implementation.