

# Reconstruction, Occlusion, Immersion

**Sébastien Roy**

Département d'Informatique et de recherche opérationnelle  
Université de Montréal



October 28 2005  
University of Maryland

Two recent algorithms for reconstruction will be presented.

- 1 Passive reconstruction
  - Multi-view stereo algorithm
  - Occlusion modeling
- 2 Active reconstruction
  - Camera-projector system
  - Structured light reconstruction
  - Large multi-projector immersive projection

# Thanks

The work presented has been possible thanks to these members of the Laboratoire Vision 3D:

Marc-Antoine Drouin

Jean-Philippe Tardif

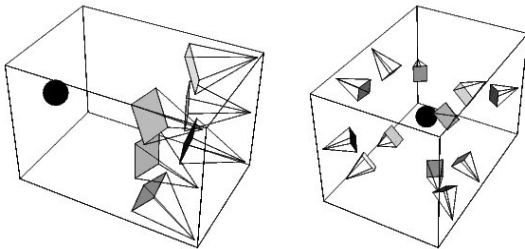
Gaspard Petit

Jamil Draréni

Martin Trudeau

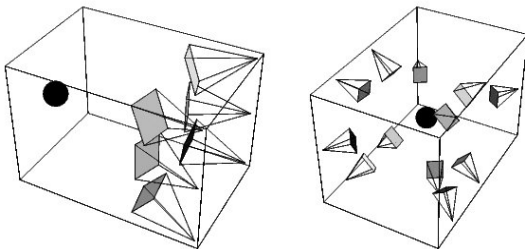
# 3D Reconstruction

Goal: Establish correspondance between two or more views of a scene to reconstruct by triangulation.



# 3D Reconstruction

Goal: Establish correspondance between two or more views of a scene to reconstruct by triangulation.



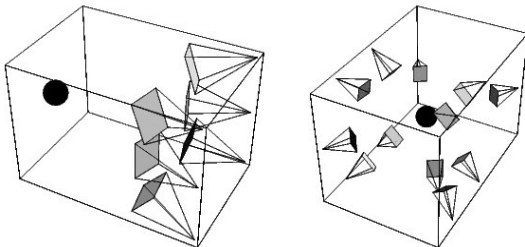
## Stereo reconstruction

All cameras are on one "side" of the scene

- Only allows a partial reconstruction, from the point of view of a camera
- Pixel representation is typical (depth maps)

# 3D Reconstruction

Goal: Establish correspondance between two or more views of a scene to reconstruct by triangulation.



## Volumetric reconstruction

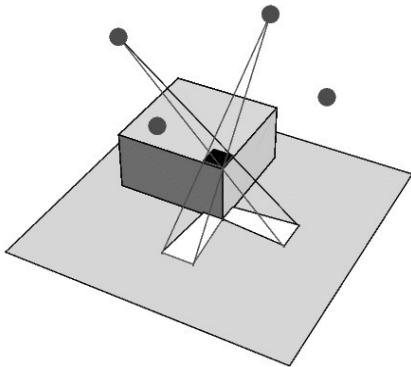
Cameras can "go around" the scene

- Allows a full reconstruction
- Voxel representation is typical

# Occlusion

A point in the scene which is not directly visible from a camera is *occluded*.

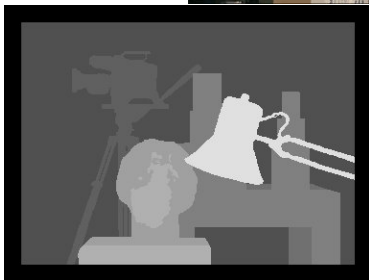
Occluded points affect correspondence as they are harder (or impossible) to match.



Camera separation increase precision, but also occlusion.

# Stereo energy formulation

Occlusion depends on the camera configuration and on the depth variation in the scene.





# Stereo energy formulation

## Depth configuration

For a set of reference pixels  $\mathcal{P}$  and a set of depth labels  $\mathcal{Z}$ , a **depth configuration**  $f$  is

$$f : \mathcal{P} \mapsto \mathcal{Z}$$

The energy function to minimize is

$$E(f) = \sum_{\mathbf{p} \in \mathcal{P}} e(\mathbf{p}, f(\mathbf{p})) + \sum_{\mathbf{p} \in \mathcal{P}} \sum_{\mathbf{r} \in \mathcal{N}_{\mathbf{p}}} s(\mathbf{p}, \mathbf{r}, f(\mathbf{p}), f(\mathbf{r}))$$

$\mathcal{N}_{\mathbf{p}}$  is the neighborhood of pixel  $\mathbf{p}$ .

Can be solved efficiently...

... since  $e(\mathbf{p}, f(\mathbf{p}))$  is independent over  $\mathbf{p}$ .

# Stereo energy formulation

## Depth configuration

For a set of reference pixels  $\mathcal{P}$  and a set of depth labels  $\mathcal{Z}$ , a **depth configuration**  $f$  is

$$f : \mathcal{P} \mapsto \mathcal{Z}$$

The energy function to minimize is

$$E(f) = \sum_{\mathbf{p} \in \mathcal{P}} e(\mathbf{p}, f(\mathbf{p})) + \sum_{\mathbf{p} \in \mathcal{P}} \sum_{\mathbf{r} \in \mathcal{N}_{\mathbf{p}}} s(\mathbf{p}, \mathbf{r}, f(\mathbf{p}), f(\mathbf{r}))$$

$\mathcal{N}_{\mathbf{p}}$  is the neighborhood of pixel  $\mathbf{p}$ .

Can be solved efficiently...

... since the smoothing term has a simple 2-clique form.

# Visibility Masks

## Volumetric visibility

Given a depth configuration  $f$ , the visibility of a 3D reference point  $\mathbf{q}$  for camera  $i$  is

$$V_i(\mathbf{q}, f)$$

(1 if visible, 0 if not)

## Visibility mask

The visibility of point  $\mathbf{q}$  for  $N$  supporting cameras is a vector

$$\mathbf{V}(\mathbf{q}, f) = \bigoplus_{i=1\dots N} V_i(\mathbf{q}, f)$$

( $\mathbf{q}$  is always visible in the reference camera)

Visibility is a long-range interaction. Immediate neighbors are insufficient to compute it.

## Visibility configuration

For a set of possible visibility masks  $\mathcal{M}$ , a **visibility configuration**  $g$  is

$$g : \mathcal{P} \mapsto \mathcal{M}$$

New energy *visibility aware* function is

$$E(f, g) = \sum_{\mathbf{p} \in \mathcal{P}} e(\mathbf{p}, f(\mathbf{p}), g(\mathbf{p})) + \text{smoothing}$$

A typical matching term is

$$e(\mathbf{p}, z, \mathbf{m}) = \frac{\mathbf{m} \cdot C(\mathbf{p}|z)}{|\mathbf{m}|} \quad \text{instead of} \quad e(\mathbf{p}, z) = C(\mathbf{p}|z)$$

- 3D points  $\mathbf{p}|z$  is  $\mathbf{p}$  augmented by depth  $z$
- $C(\mathbf{q})$  is a vector of matching costs for each camera
- the  $l_1$  norm  $|\mathbf{m}|$  is the number of visible cameras
- the case  $|\mathbf{m}| = 0$  is interesting...

# Depth and Visibility

Let's try to disentangle visibility and depth...

## Geo-consistency

A depth configuration  $f$  is geo-consistent with a visibility configuration  $g$  if

$$g(\mathbf{p}) \leq \mathbf{V}(\mathbf{p}|f(\mathbf{p}), f)$$

for each components and all  $\mathbf{p} \in \mathcal{P}$ .

## Intuitively

A geo-consistent depth configuration is still geo-consistent when removing too many cameras from the visibility.

## A new problem

Solve  $E$  in depth  $f$  and visibility  $g$ , with the constraint that  $f$  is geo-consistent with  $g$ .

# Depth and Visibility

Let's try to disentangle visibility and depth...

## Geo-consistency

A depth configuration  $f$  is geo-consistent with a visibility configuration  $g$  if

$$g(\mathbf{p}) \leq \mathbf{V}(\mathbf{p}|f(\mathbf{p}), f)$$

for each components and all  $\mathbf{p} \in \mathcal{P}$ .

## Intuitively

A geo-consistent depth configuration is still geo-consistent when removing too many cameras from the visibility.

## A new problem

Solve  $E$  in depth  $f$  and visibility  $g$ , with the constraint that  $f$  is geo-consistent with  $g$ .

# Depth and Visibility

Let's try to disentangle visibility and depth...

## Geo-consistency

A depth configuration  $f$  is geo-consistent with a visibility configuration  $g$  if

$$g(\mathbf{p}) \leq \mathbf{V}(\mathbf{p}|f(\mathbf{p}), f)$$

for each components and all  $\mathbf{p} \in \mathcal{P}$ .

## Intuitively

A geo-consistent depth configuration is still geo-consistent when removing too many cameras from the visibility.

## A new problem

Solve  $E$  in depth  $f$  and visibility  $g$ , with the constraint that  $f$  is geo-consistent with  $g$ .

# Solving Depth and Visibility

Some approaches...

## All cameras visible

Set  $g^0(\mathbf{p}) = (1, 1, \dots, 1)$  for all  $\mathbf{p}$ . Solve  $E(f, g^0)$ .

This is equivalent to solving only  $E(f)$ .

## Simultaneous depth and visibility

Test all combinations in  $\mathcal{Z} \times \mathcal{M}$ .

This is an intractable search space.

## Reduce the set $\mathcal{M}$

For a given camera configuration, not all masks are possible.

Still, the set is too big, or wrong.

## Photo-consistency

Remove cameras that provide photo-inconsistent matches.

Problematic. Some cameras should be removed and are not. Space carving uses this.



# Solving Depth and Visibility

Some approaches...

All cameras visible

Set  $g^0(\mathbf{p}) = (1, 1, \dots, 1)$  for all  $\mathbf{p}$ . Solve  $E(f, g^0)$ .

This is equivalent to solving only  $E(f)$ .

Simultaneous depth and visibility

Test all combinations in  $\mathcal{Z} \times \mathcal{M}$ .

This is an intractable search space.

Reduce the set  $\mathcal{M}$

For a given camera configuration, not all masks are possible.

Still, the set is too big, or wrong.

Photo-consistency

Remove cameras that provide photo-inconsistent matches.

Problematic. Some cameras should be removed and are not. Space carving uses this.

# Solving Depth and Visibility

Some approaches...

All cameras visible

Set  $g^0(\mathbf{p}) = (1, 1, \dots, 1)$  for all  $\mathbf{p}$ . Solve  $E(f, g^0)$ .

This is equivalent to solving only  $E(f)$ .

Simultaneous depth and visibility

Test all combinations in  $\mathcal{Z} \times \mathcal{M}$ .

This is an intractable search space.

Reduce the set  $\mathcal{M}$

For a given camera configuration, not all masks are possible.

Still, the set is too big, or wrong.

Photo-consistency

Remove cameras that provide photo-inconsistent matches.

Problematic. Some cameras should be removed and are not. Space carving uses this.

# Solving Depth and Visibility

Some approaches...

## All cameras visible

Set  $g^0(\mathbf{p}) = (1, 1, \dots, 1)$  for all  $\mathbf{p}$ . Solve  $E(f, g^0)$ .

This is equivalent to solving only  $E(f)$ .

## Simultaneous depth and visibility

Test all combinations in  $\mathcal{Z} \times \mathcal{M}$ .

This is an intractable search space.

## Reduce the set $\mathcal{M}$

For a given camera configuration, not all masks are possible.

Still, the set is too big, or wrong.

## Photo-consistency

Remove cameras that provide photo-inconsistent matches.

Problematic. Some cameras should be removed and are not. Space carving uses this.

# A new implicit occlusion model

Make the dependency temporal

Let  $f^0$  be the depth configuration minimizing  $E(f, g^0)$ .

For  $t > 0$ , we minimize

$$\sum_{\mathbf{p} \in \mathcal{P}} e(\mathbf{p}, f^t(\mathbf{p}), \mathbf{V}(\mathbf{p} | f^t(\mathbf{p}), f^{t-1})) + \text{smoothing}$$

where

$$g^t(\mathbf{p}) = \mathbf{V}(\mathbf{p} | f^t(\mathbf{p}), f^{t-1})$$

Intuitively

- 1 Start solving depth with all cameras visible.
- 2 Use the current depth configuration to compute new visibility.
- 3 Solve for depth using the new visibility, and go back to step 2.

# A new implicit occlusion model

Make the dependency temporal

Let  $f^0$  be the depth configuration minimizing  $E(f, g^0)$ .

For  $t > 0$ , we minimize

$$\sum_{\mathbf{p} \in \mathcal{P}} e(\mathbf{p}, f^t(\mathbf{p}), \mathbf{V}(\mathbf{p} | f^t(\mathbf{p}), f^{t-1})) + \text{smoothing}$$

where

$$g^t(\mathbf{p}) = \mathbf{V}(\mathbf{p} | f^t(\mathbf{p}), f^{t-1})$$

Intuitively

- 1 Start solving depth with all cameras visible.
- 2 Use the current depth configuration to compute new visibility.
- 3 Solve for depth using the new visibility, and go back to step 2.

# The question that kills...

It's all nice, but...

Does it converge?

## History of visibility

The new visibility mask is multiplied by the previous visibility.  
This ensures that once a camera is removed, it is never used again.  
This also ensures convergence.

The intermediate depth and visibility configurations are not geo-consistent, except at convergence.

# The question that kills...

It's all nice, but...

Does it converge?

## History of visibility

The new visibility mask is multiplied by the previous visibility.  
This ensures that once a camera is removed, it is never used again.  
This also ensures convergence.

The intermediate depth and visibility configurations are not geo-consistent, except at convergence.

# The question that kills...

It's all nice, but...

Does it converge?

## History of visibility

The new visibility mask is multiplied by the previous visibility. This ensures that once a camera is removed, it is never used again. This also ensures convergence.

At convergence, the final depth  $f^{t+1} = f^t$  is geo-consistent with the visibility  $g^{t+1}$ .

The intermediate depth and visibility configurations are not geo-consistent, except at convergence.



# The question that kills...

It's all nice, but...

Does it converge?

## History of visibility

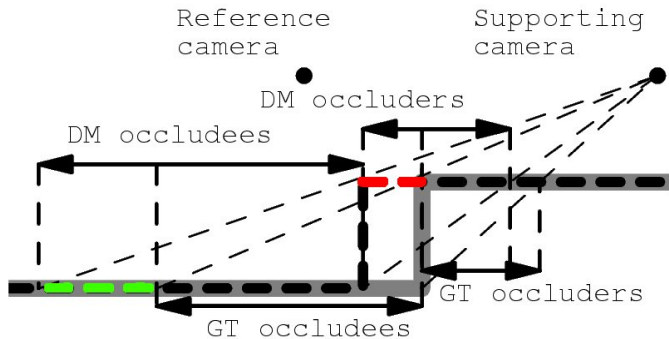
The new visibility mask is multiplied by the previous visibility. This ensures that once a camera is removed, it is never used again. This also ensures convergence.

At convergence, the final depth  $f^{t+1} = f^t$  is geo-consistent with the visibility  $g^{t+1}$ .

The intermediate depth and visibility configurations are not geo-consistent, except at convergence.

# Pseudo-visibility

- The depth of occluded pixels tends to be underestimated.
- These misplaced pixels can wrongly occlude other pixel.
- This makes closer objects appear larger.



# Pseudo-visibility

Empirical study of misclassification of pixels.

Algo	Scenes from Middlebury comparative study						
	Tsukuba Head and Lamp			Sawtooth			
	Real status of pixels classified from depth map as occluders						
	occludee	occluder	regular	occludee	occluder	regular	
bp [24]	<b>44.8</b> ●	16.3	38.9	<b>42.6</b> ●	3.8	53.6	
bnv [2]	<b>50.4</b> ●	15.4	34.2	<b>42.6</b> ●	4.3	53.3	
	Real status of pixels classified from depth map as occludees						
	occludee	occluder	regular	occludee	occluder	regular	
	bp [24]	15.5	5.9	<b>76.6</b> ●	5.5	1.1	<b>93.4</b> ●
	bnv [2]	16.4	5.8	<b>77.8</b> ●	7.2	1.1	<b>91.7</b> ●
	Real status of pixels classified from depth map as regulars						
	occludee	occluder	regular	occludee	occluder	regular	
	bp [24]	1.0	<b>2.0</b>	97.0	0.5	<b>1.5</b>	98.0
	bnv [2]	1.0	<b>2.0</b>	96.9	0.5	<b>1.5</b>	98.0

## Pseudo-visibility

An occluder should be considered invisible, just like an occluded pixel.

# Pseudo-visibility

Empirical study of misclassification of pixels.

Algo	Scenes from Middlebury comparative study						
	Tsukuba Head and Lamp			Sawtooth			
	Real status of pixels classified from depth map as occluders						
	occludee	occluder	regular	occludee	occluder	regular	
bp [24]	<b>44.8</b> ●	16.3	38.9	<b>42.6</b> ●	3.8	53.6	
bnv [2]	<b>50.4</b> ●	15.4	34.2	<b>42.6</b> ●	4.3	53.3	
	Real status of pixels classified from depth map as occludees						
	occludee	occluder	regular	occludee	occluder	regular	
	bp [24]	15.5	5.9	<b>76.6</b> ●	5.5	1.1	<b>93.4</b> ●
	bnv [2]	16.4	5.8	<b>77.8</b> ●	7.2	1.1	<b>91.7</b> ●
	Real status of pixels classified from depth map as regulars						
	occludee	occluder	regular	occludee	occluder	regular	
	bp [24]	1.0	<b>2.0</b>	97.0	0.5	<b>1.5</b>	98.0
	bnv [2]	1.0	<b>2.0</b>	96.9	0.5	<b>1.5</b>	98.0

## Pseudo-visibility

An occluder should be considered invisible, just like an occluded pixel.

The cost function is the same for all algorithms.

The smoothing term is

$$s(\mathbf{p}, \mathbf{r}, f(\mathbf{p}), f(\mathbf{r})) = \lambda g(\mathbf{p}, \mathbf{r}) l(f(\mathbf{p}), f(\mathbf{r}))$$

where

$$g(\mathbf{p}, \mathbf{r}) = \begin{cases} 3 & \text{if } |I_{ref}(\mathbf{p}) - I_{ref}(\mathbf{q})| < 5 \\ 1 & \text{otherwise} \end{cases}$$

There is a lot to do to make the matching and smoothing cost more general.

# Results

The algorithms are labelled as **C-M** where **C** is the cost function and **M** is the minimization algorithm used.

## Cost functions

FULL	No occlusion modeling
GEO	Our method
KAN	Improved Nakamura (all left/all right)

## Minimization algorithms

MF	Maximum-flow
BNV	Graph-cuts

# Results : Middlebury dataset

Algorithm	Scenes from Middlebury					
	Barn1	Barn2	Bull	Poster	Venus	Sawtooth
FULL-BNV	3.5 %	3.1 %	0.7 %	3.7 %	3.4 %	3.3%
FULL-MF	4.0 %	5.4 %	0.7 %	3.4 %	4.4 %	3.8 %
GEO-BNV	<b>0.8 %</b>	<b>0.6 %</b>	0.4 %	1.1 %	<b>2.4 %</b>	<b>1.1 %</b>
GEO-MF	1.5 %	0.9 %	<b>0.3 %</b>	1.4 %	3.4 %	1.5 %
KAN-BNV	1.4 %	1.5 %	0.9 %	1.1 %	4.0 %	1.5%
KAN-MF	1.1 %	1.2 %	<b>0.3 %</b>	<b>0.9 %</b>	5.8 %	2.2 %

- Best performance results (select the best  $\lambda$  per image)
- 8 images used on a single baseline
- 8 iterations are used in GEO, but change is minimal after 4

# Results : Middlebury dataset





# Results : Tsukuba

Algorithm	Baseline	Error (whole image)	Error (mask)
GEO-BNV pt	1x	2.23%	1.53%
KZ1	1x	2.30%	2.01%
GEO-BNV	1x	2.46%	1.64%
GEO-MF	1x	3.42%	2.52%
GEO-BNV	2x	2.69%	2.11%
GEO-MF	2x	2.62%	1.28%

- 5 images used, single baseline
- The right column removes pixels detected as breaking the ordering constraint

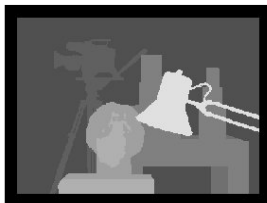
# Results : Middlebury dataset



GEO-BNV



GEO-BNV pt



Ground truth

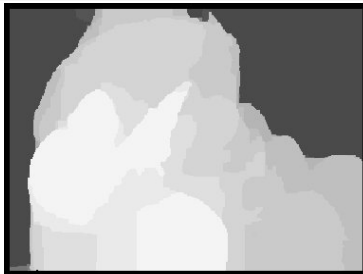


Ordering constraint mask

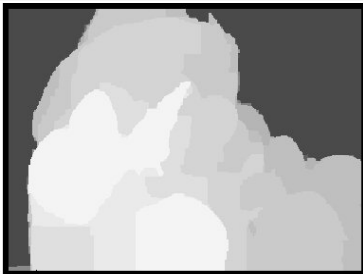
An increase of the baseline always increase the ammount of occlusion.

⇒

We can test the robustness to occlusion by increasing the baseline.



GEO-MF 1x



GEO-MF 3x

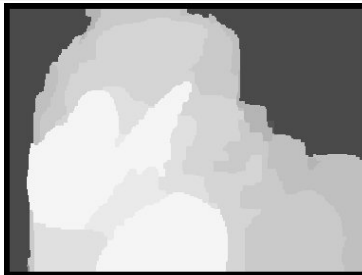
If the results are similar, the occlusion model is good.

# Results : Noël

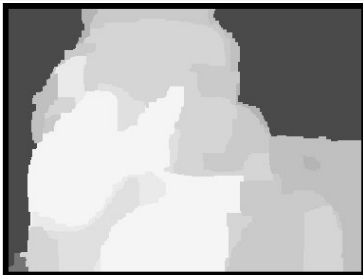
An increase of the baseline always increase the ammount of occlusion.

⇒

We can test the robustness to occlusion by increasing the baseline.



KZI 1x



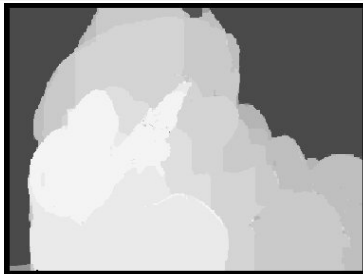
KZI 3x

If the results are similar, the occlusion model is good.

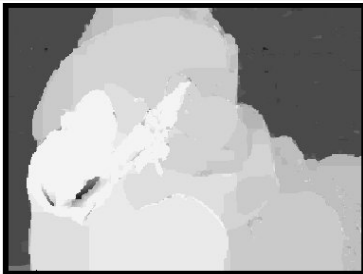
An increase of the baseline always increase the ammount of occlusion.

⇒

We can test the robustness to occlusion by increasing the baseline.



NAKA-MF 1x



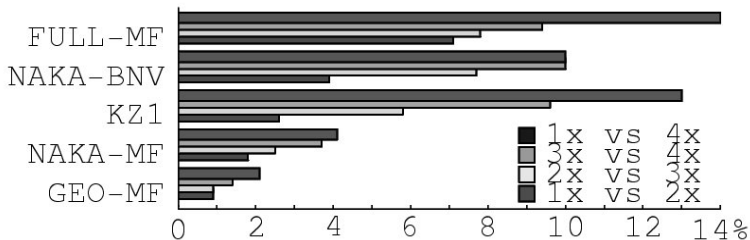
NAKA-MF 3x

If the results are similar, the occlusion model is good.

## Relative evaluation

The incompatibility of pixel  $\mathbf{p}$  between two experimental conditions  $i$  and  $j$  is

$$|f_i(\mathbf{p}) - f_j(\mathbf{p})| > 1$$



Percentage of incompatible pixels with different baselines.

# From Passive to Active

Passive reconstruction has a lot of problems to address

- photometric ambiguity
- visibility problem (geo-consistency)

To simplify the matching process, we can use a projector.

- Control the texture of scene objects
- Project codes to simplify correspondence

99% of industrial scanners use some form of active reconstruction

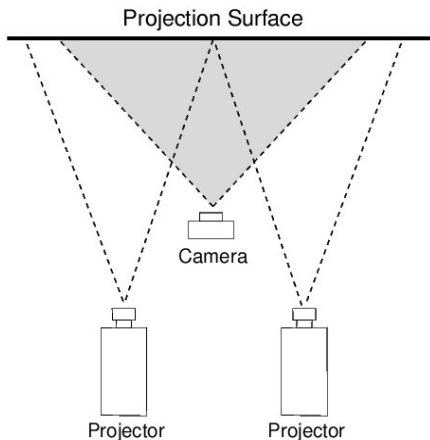
# Projecting on anything

The future of projection is simple: project on any surface you wish, and cover that surface with many projectors.

Ideal case

- perfectly aligned projectors
- no radial distortion
- flat projection surface

Most common distortion:  
perspective solved by keystone  
adjustment



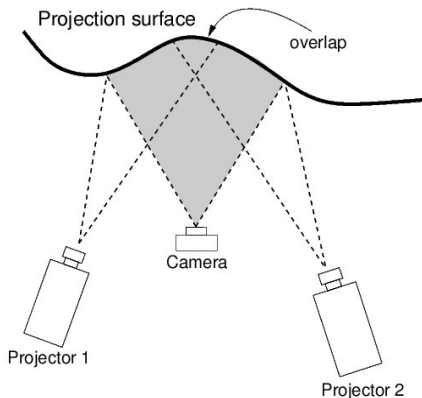


# Projecting on anything

Real life...

Ideal case

- non aligned projectors
- radial distortion
- curved projection surface



# Multi-projector Immersion

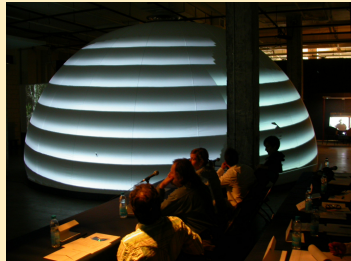
## Cyclorama

- 9 projectors
- 120' circumference cylinder
- 10' high



## Half-Sphere

- inflated
- 24' diameter
- 6 projectors from outside



# Catadioptric camera

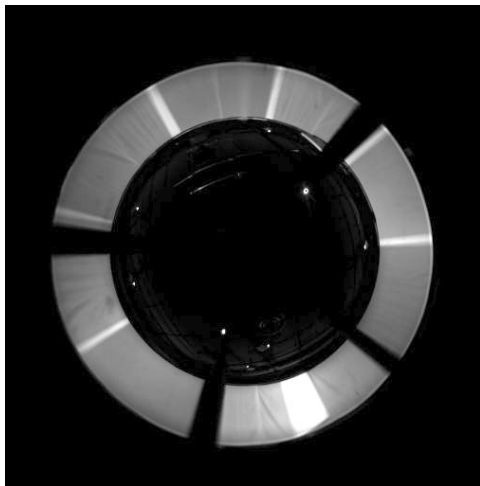
The camera represent the observer.  
For complete immersion, use a catadioptric camera or a fisheye.



# Pixel ratio

Here is the view of the catadioptric camera.

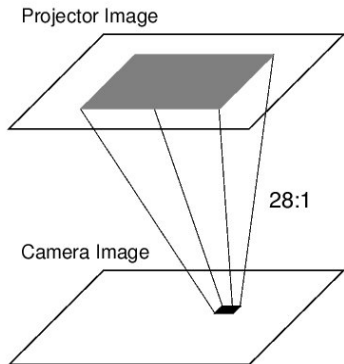
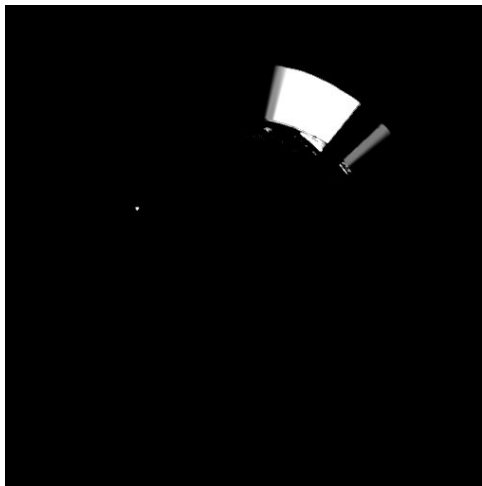
The  $1024 \times 768$  projector image (786432 pixels) is compressed into only 28000 pixels of the camera image ( $1008 \times 1017 = 1025136$ ), which is a compression ratio of 28:1.



# Pixel ratio

Here is the view of the catadioptric camera.

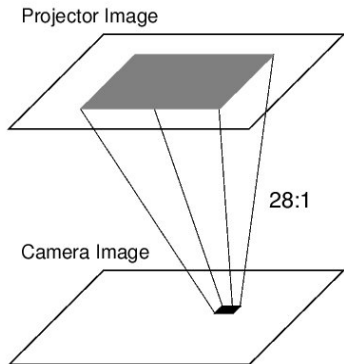
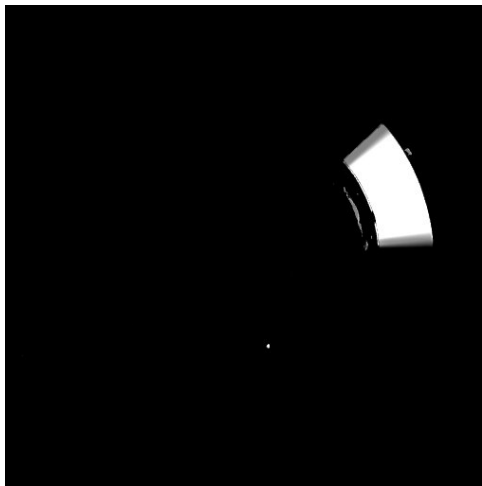
The  $1024 \times 768$  projector image (786432 pixels) is compressed into only 28000 pixels of the camera image ( $1008 \times 1017 = 1025136$ ), which is a compression ratio of 28:1.



# Pixel ratio

Here is the view of the catadioptric camera.

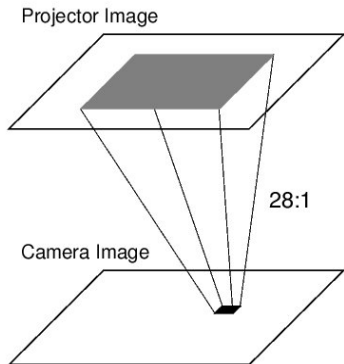
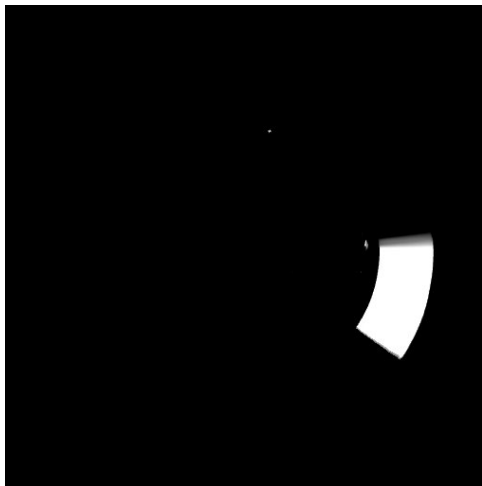
The  $1024 \times 768$  projector image (786432 pixels) is compressed into only 28000 pixels of the camera image ( $1008 \times 1017 = 1025136$ ), which is a compression ratio of 28:1.



# Pixel ratio

Here is the view of the catadioptric camera.

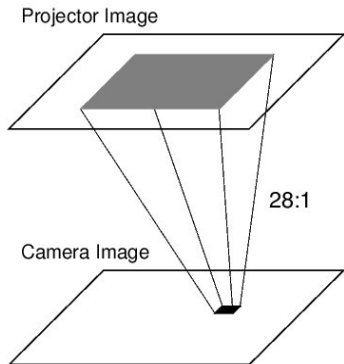
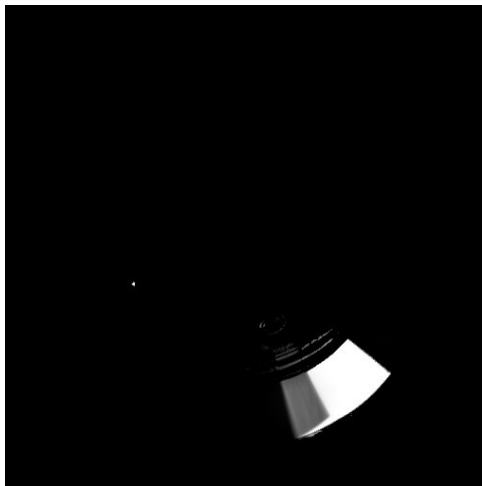
The  $1024 \times 768$  projector image (786432 pixels) is compressed into only 28000 pixels of the camera image ( $1008 \times 1017 = 1025136$ ), which is a compression ratio of 28:1.



# Pixel ratio

Here is the view of the catadioptric camera.

The  $1024 \times 768$  projector image (786432 pixels) is compressed into only 28000 pixels of the camera image ( $1008 \times 1017 = 1025136$ ), which is a compression ratio of 28:1.

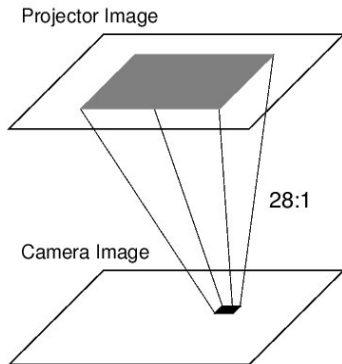
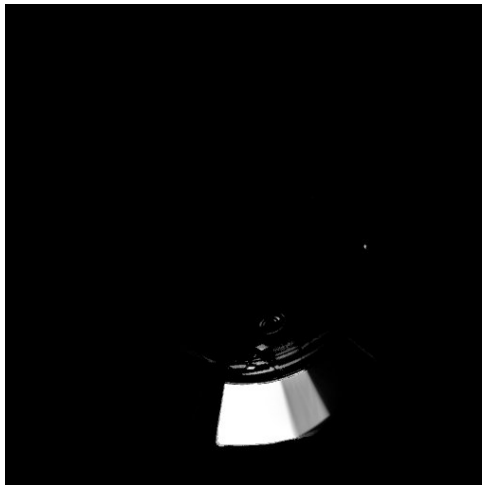




# Pixel ratio

Here is the view of the catadioptric camera.

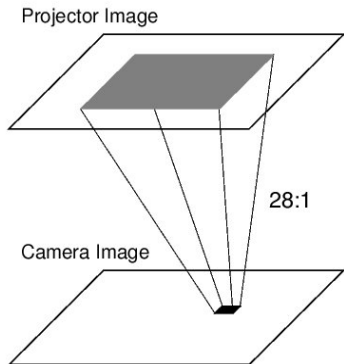
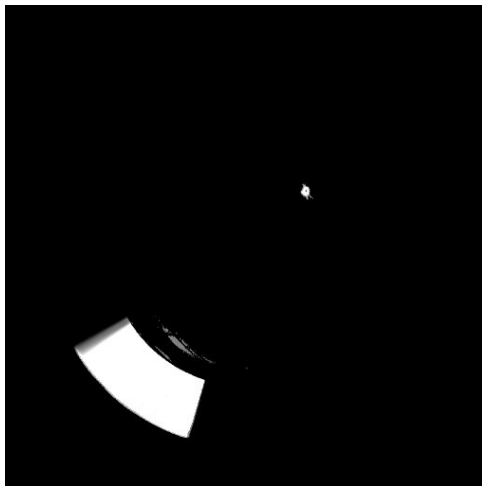
The  $1024 \times 768$  projector image (786432 pixels) is compressed into only 28000 pixels of the camera image ( $1008 \times 1017 = 1025136$ ), which is a compression ratio of 28:1.



# Pixel ratio

Here is the view of the catadioptric camera.

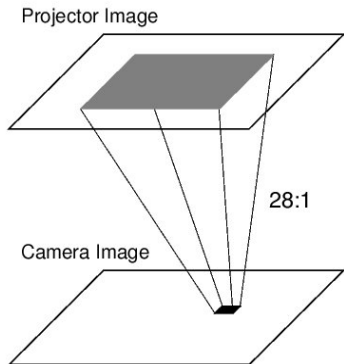
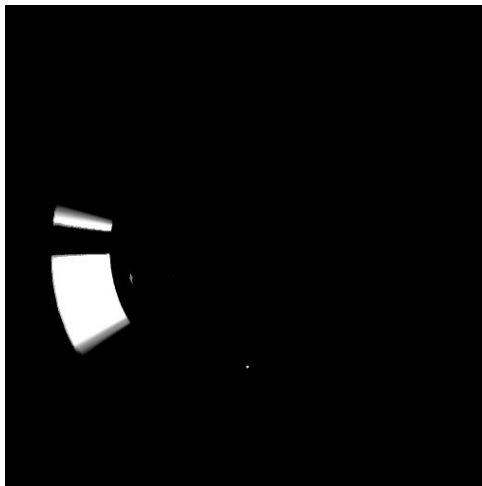
The  $1024 \times 768$  projector image (786432 pixels) is compressed into only 28000 pixels of the camera image ( $1008 \times 1017 = 1025136$ ), which is a compression ratio of 28:1.



# Pixel ratio

Here is the view of the catadioptric camera.

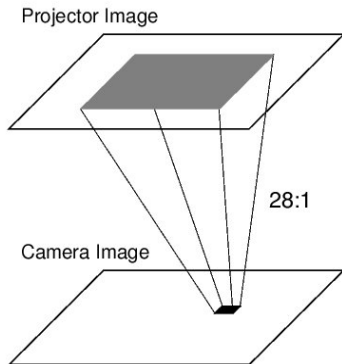
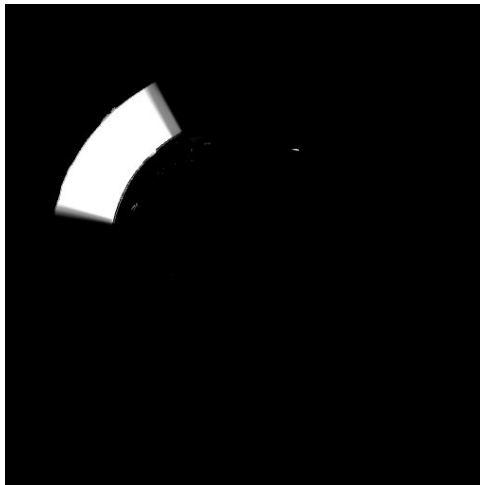
The  $1024 \times 768$  projector image (786432 pixels) is compressed into only 28000 pixels of the camera image ( $1008 \times 1017 = 1025136$ ), which is a compression ratio of 28:1.



# Pixel ratio

Here is the view of the catadioptric camera.

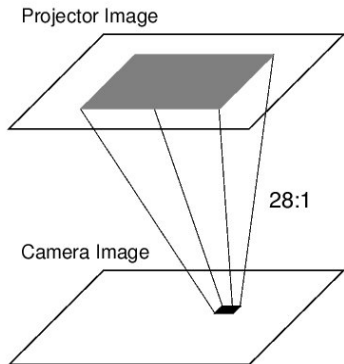
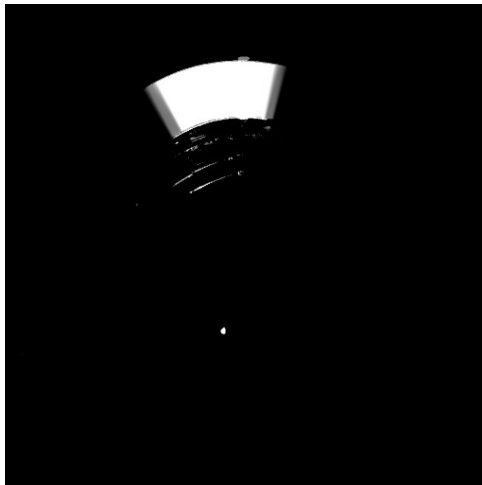
The  $1024 \times 768$  projector image (786432 pixels) is compressed into only 28000 pixels of the camera image ( $1008 \times 1017 = 1025136$ ), which is a compression ratio of 28:1.



# Pixel ratio

Here is the view of the catadioptric camera.

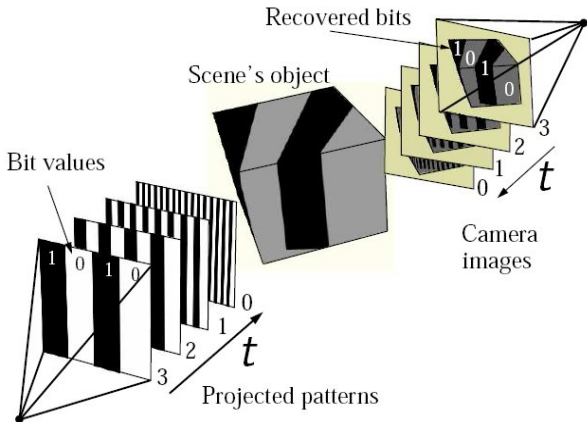
The  $1024 \times 768$  projector image (786432 pixels) is compressed into only 28000 pixels of the camera image ( $1008 \times 1017 = 1025136$ ), which is a compression ratio of 28:1.



# Structured Light patterns

For each projector pixel  $(x, y)$  we define a code  $P(x, y) = x \oplus y$  (10+10 bits). A succession of pattern images  $P_i$  and their inverse  $\bar{P}_i$  is projected.

$$P_i(x, y) = \text{bit}_i(P(x, y)) \quad \text{and} \quad \bar{P}_i(x, y) = 1 - P_i(x, y)$$



# Gray Codes

$$G(x) = x \text{ xor } (x \gg 1)$$

For two neighbor pixels  $x$  and  $x + 1$ , the number of changed bits is the number of black-white transitions in the pattern images.

$x$	0	1	2	3	4	5	6	7
$P_0$	0	1	0	1	0	1	0	1
$P_1$	0	0	1	1	0	0	1	1
$P_2$	0	0	0	0	1	1	1	1

Gray code ensure that one bit changes between consecutive codes.

$G(x)$	0	1	3	2	6	7	5	4
$P_0$	0	1	1	0	0	1	1	0
$P_1$	0	0	1	1	1	1	0	0
$P_2$	0	0	0	0	1	1	1	1

# Decoding Structured Light patterns

The camera record pattern  $P_i$  as image  $C_i$ , and  $\bar{P}_i$  as  $\bar{C}_i$ .  
To recover the code  $C(x, y)$

$$C(x, y) = \bigoplus_i \mathbf{thresh}(C_i(x, y) - \bar{C}_i(x, y))$$

where

$$\mathbf{thresh}(v) = \begin{cases} 1 & \text{if } v > 0 \\ 0 & \text{otherwise} \end{cases}$$

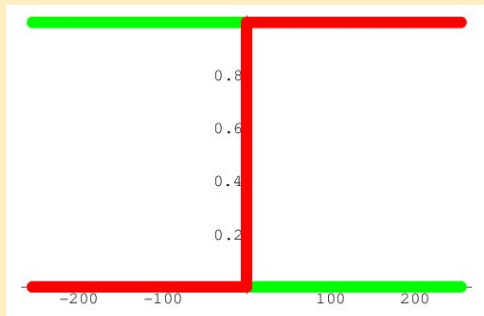
Hummmm....



# Decoding Structured Light patterns

Seen as a probability, we could improve the selection in the ambiguous zone near 0.

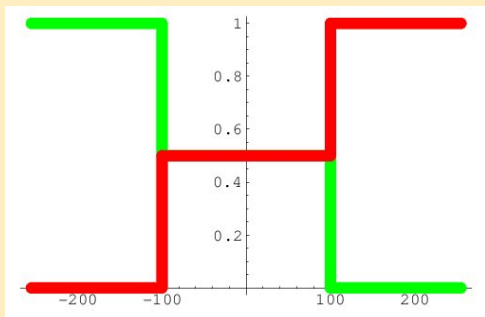
Threshold



# Decoding Structured Light patterns

Seen as a probability, we could improve the selection in the ambiguous zone near 0.

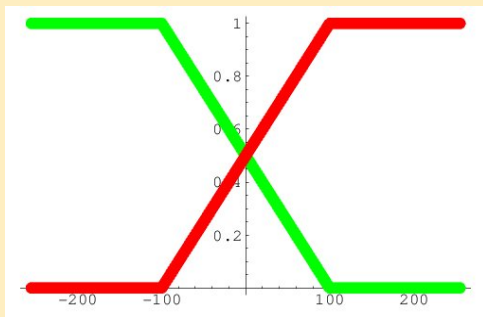
Fixed ambiguity



# Decoding Structured Light patterns

Seen as a probability, we could improve the selection in the ambiguous zone near 0.

## Progressive ambiguity



# Code probability

- First unknown bit is 0 / 1 with probability 0.2 / 0.8
- Second unknown bit is 0 / 1 with probability 0.4 / 0.6

1	?	0	1	0	1	0	?	1.00
1	0	0	1	0	1	0	?	0.20
1	1	0	1	0	1	0	?	0.80
1	?	0	1	0	1	0	0	0.40
1	?	0	1	0	1	0	1	0.60
1	0	0	1	0	1	0	0	0.08
1	0	0	1	0	1	0	1	0.12
1	1	0	1	0	1	0	0	0.32
1	1	0	1	0	1	0	1	0.48

# Markov Random Fields

## Sites

Pixels  $(x, y)$  of the camera image.

## Labels

All codes  $C(c, y)$  with non zero probability.  
(Codes are real, not Gray)

## Smoothness term

Codes of neighboring pixels should be similar.  
(Absolute difference of codes)

# Markov Random Fields

## Sites

Pixels  $(x, y)$  of the camera image.

## Labels

All codes  $C(c, y)$  with non zero probability.  
(Codes are real, not Gray)

## Smoothness term

Codes of neighboring pixels should be similar.  
(Absolute difference of codes)

# Markov Random Fields

## Sites

Pixels  $(x, y)$  of the camera image.

## Labels

All codes  $C(c, y)$  with non zero probability.  
(Codes are real, not Gray)

## Smoothness term

Codes of neighboring pixels should be similar.  
(Absolute difference of codes)

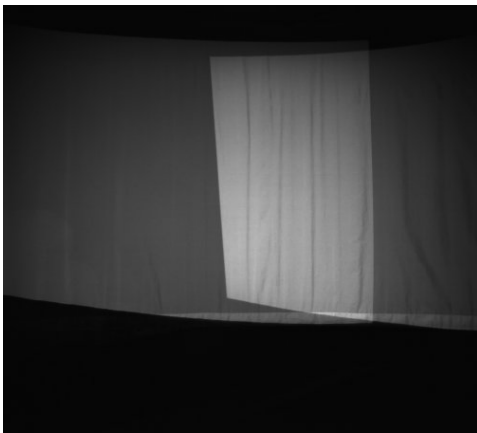
Solved using Maximum-flow, ICM, ...

(pick your favorite, the labels are 1-D ordered, the problem is easy to solve)

# Example

- Two projectors setup
- Flat projection surface.
- Perspective camera

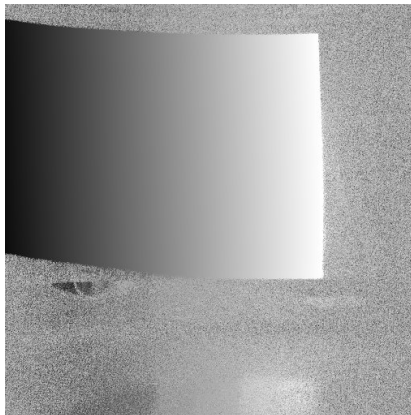
What the camera sees (with two projectors on):



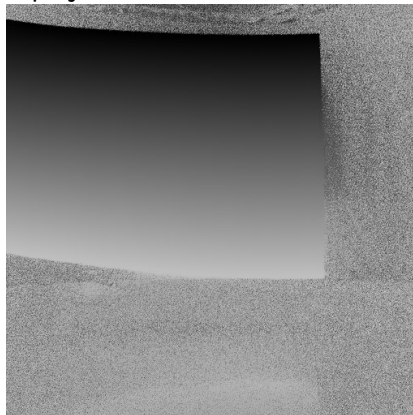


# Example correspondences

Camera to first projector...



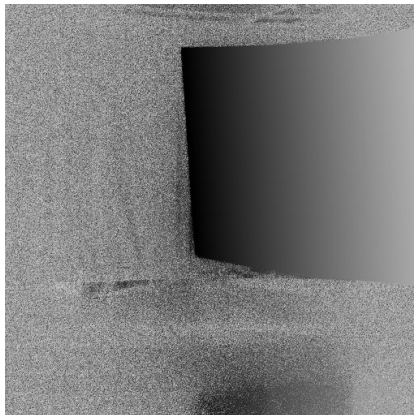
Camera  $\rightarrow$  Projector X



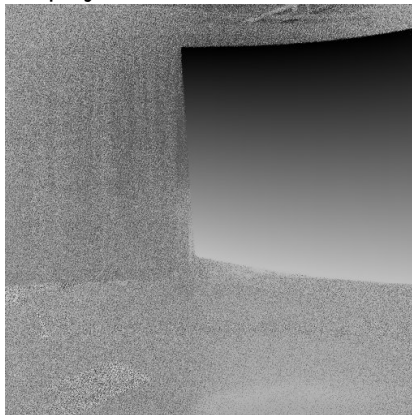
Camera  $\rightarrow$  Projector Y

# Example correspondences

Camera to second projector...



Camera  $\rightarrow$  Projector X

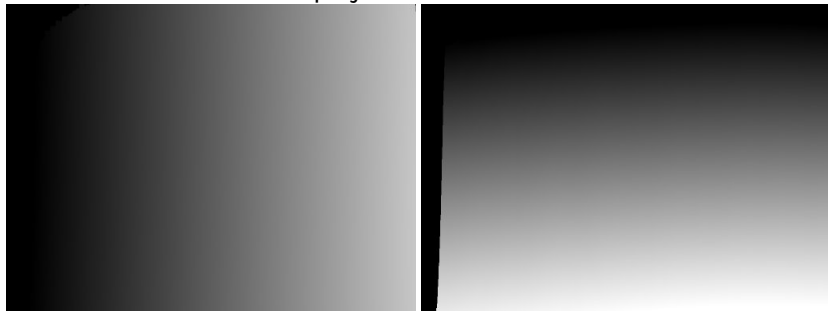


Camera  $\rightarrow$  Projector Y

# Example inverse correspondences

The Camera-Projector correspondences must be inverted (carefully).

First projector to Camera...



Projector  $\rightarrow$  Camera X

Projector  $\rightarrow$  Camera Y

# Example inverse correspondences

The Camera-Projector correspondences must be inverted (carefully).

Second projector to Camera...



Projector  $\rightarrow$  Camera X

Projector  $\rightarrow$  Camera Y

# Conclusion

## Reconstruction

- Occlusion is becoming the dominant problem in stereo
- Visibility maps are as important as depth itself
- $N$ -camera stereo ( $N > 2$ ) is a tremendous help
- Occlusion can be modelled on top of regular stereo algorithms
- Better occlusion benchmarks are needed
- ...

## Immersion

- Immersion = multiple projectors and bad pixel ratios
- Accurate 3D reconstruction is possible.
- ...

# Example 1

A new camera model for distortion, in between: Bonjour, cet été un être.

# Example 2

- Parametric
  - Pin-hole + radial distortion
  - Unified catadioptric model
  - Caustics, Non-Single Viewpoint (NSVP)
- Fully general (generic)
  - One pixel  $\leftrightarrow$  one sampling ray
  - Possible NSVP

# Example 2

- Parametric
  - Pin-hole + radial distortion
  - Unified catadioptric model
  - Caustics, Non-Single Viewpoint (NSVP)
- Fully general (generic)
  - One pixel  $\leftrightarrow$  one sampling ray
  - Possible NSVP



## Example 3: overlays

This frame produces

- This is 1 avec -
- This is 2
- This is 2 avec -
- This is 3

## Example 3: overlays

This frame produces

- This is 1 avec -
- This is 2
- This is 2 avec -
- This is 3

## Example 3: overlays

This frame produces

- This is 1 avec -
- This is 2
- This is 2 avec -
- This is 3

## Example 4 : more overlays

Bonjour, c'est une diapo **rigolote**, n'est-ce pas?  
Et en plus!

## Example 4 : more overlays

Bonjour, c'est une diapo rigolote, n'est-ce pas?  
Et en **plus!**

## Example 5: Alerts

Camera with general radially symmetric distortion:

- Distorsion center = principal point
- Distortion is radially symmetric but general
  - ✦ Each distortion circle  $\rightarrow$  Sampling Right Viewing Cone
- Cones' axis are aligned on the **optical axis**
- SVP: identical cones's vertex
- NSVP: cones's vertex only on the optical axis

## Example 5: Alerts

Camera with general radially symmetric distortion:

- Distorsion center = principal point
- Distortion is radially symmetric but general
  - Each distortion circle  $\rightarrow$  Sampling Right Viewing Cone
- Cones' axis are aligned on the **optical axis**
- SVP: identical cones's vertex
- NSVP: cones's vertex only on the optical axis

## Example 5: Alerts

Camera with general radially symmetric distortion:

- Distorsion center = principal point
- Distortion is radially symmetric but general
  - Each **distortion circle** → Sampling Right Viewing Cone
- Cones' axis are aligned on the **optical axis**
- SVP: identical cones's vertex
- NSVP: cones's vertex only on the optical axis



## Example 5: Alerts

Camera with general radially symmetric distortion:

- Distorsion center = principal point
- Distortion is radially symmetric but general
  - Each **distortion circle** → Sampling Right Viewing Cone
- Cones' axis are aligned on the **optical axis**
- SVP: identical cones's vertex
- NSVP: cones's vertex only on the optical axis

## Example 5: Alerts

Camera with general radially symmetric distortion:

- Distorsion center = principal point
- Distortion is radially symmetric but general
  - Each **distortion circle** → Sampling Right Viewing Cone
- Cones' axis are aligned on the **optical axis**
- **SVP: identical cones's vertex**
- NSVP: cones's vertex only on the optical axis

## Example 5: Alerts

Camera with general radially symmetric distortion:

- Distorsion center = principal point
- Distortion is radially symmetric but general
  - Each **distortion circle** → Sampling Right Viewing Cone
- Cones' axis are aligned on the **optical axis**
- SVP: identical cones's vertex
- **NSVP: cones's vertex only on the optical axis**

## Example 6: multi-eq and table

It is not easy to have a gradual table. pause works, but regular  $j\dot{z}$  does not work well... must be checked...

Class	A	B	C	D
X	1	2	3	4
Y	3	4	5	6
Z	5	6	7	8

For some reason, the gradual equation does not work at the same time as pause.... must be checked...

$$A = B \tag{1}$$

$$= C \tag{2}$$

$$= D \tag{3}$$

Bonjour!

## Example 6: multi-eq and table

It is not easy to have a gradual table. pause works, but regular  $j\dot{z}$  does not work well... must be checked...

Class	A	B	C	D
X	1	2	3	4
Y	3	4	5	6
Z	5	6	7	8

For some reason, the gradual equation does not work at the same time as pause.... must be checked...

$$A = B \tag{1}$$

$$= C \tag{2}$$

$$= D \tag{3}$$

Bonjour!

## Example 6: multi-eq and table

It is not easy to have a gradual table. pause works, but regular  $j\grave{u}$  does not work well... must be checked...

Class	A	B	C	D
X	1	2	3	4
Y	3	4	5	6
Z	5	6	7	8

For some reason, the gradual equation does not work at the same time as pause.... must be checked...

$$A = B \tag{1}$$

$$= C \tag{2}$$

$$= D \tag{3}$$

Bonjour!

## Example 7 : Only, uncover, visible, invisible

On peut utiliser le only(1) pour seulement la slide 1. et oui.

On peut utiliser le uncover(2) pour voir a la slide 2. et oui.

On peut utiliser le visible(3) . et oui.

On peut utiliser le invisible(4) pour seulement la slide 4. et oui.

### Note

Nous sommes à la slide 1.

On va changer de couleur pour la diapo 3!!!

## Example 7 : Only, uncover, visible, invisible

On peut utiliser le only(1) . et oui.

On peut utiliser le uncover(2) pour voir a la slide 2. et oui.

On peut utiliser le visible(3) . et oui.

On peut utiliser le invisible(4) pour seulement la slide 4. et oui.

### Note

Nous sommes à la slide 2.

On va changer de couleur pour la diapo 3!!!



## Example 7 : Only, uncover, visible, invisible

On peut utiliser le only(1) . et oui.

On peut utiliser le uncover(2) pour voir a la slide 2. et oui.

On peut utiliser le visible(3) pour seulement la slide 3. et oui.

On peut utiliser le invisible(4) pour seulement la slide 4. et oui.

### Note

Nous sommes à la slide 3.

On va changer de couleur pour la diapo 3!!!

## Example 7 : Only, uncover, visible, invisible

On peut utiliser le only(1) . et oui.

On peut utiliser le uncover(2) pour voir a la slide 2. et oui.

On peut utiliser le visible(3) . et oui.

On peut utiliser le invisible(4) . et oui.

### Note

Nous sommes à la slide 4.

On va changer de couleur pour la diapo 3!!!

## Example 8 : text box!

### Note 1

Nous sommes ici dans un premier bloc.

$$a = 1$$

$$b = 2 + 2$$

$$c = 3 + 3 + 3$$

$$d = a + b + c$$

### Note 2

Nous sommes ici. Nous sommes ici dans un second bloc.

Et d'ailleurs voici un peu de texte rigolo pour s'amuser.

# Example 9 : Les blocks!

## Viewing cone-Calibration plane intersection

- Plane of known eucliden structure
- Dense image-calibration plane matching
  - Structured light
- Intersection shape: calibration ellipse  
(Hyperbola case discussed in the paper)

## Un autre détail...

Et bien oui, il ne fait pas oublier le code!    `for(i=0;i<10;i++)`  
...

## Un autre détail...

Et aussi une équation...

$$\sum_{i=0}^{10} x^i + i^2$$

## Example 9 : Les blocks!

### Viewing cone-Calibration plane intersection

- Plane of known eucliden structure
- Dense image-calibration plane matching
  - Structured light
- Intersection shape: calibration ellipse  
(Hyperbola case discussed in the paper)

### Un autre détail...

Et bien oui, il ne fait pas oublier le code!    `for(i=0;i<10;i++)`

...

### Un autre détail...

Et aussi une équation...

$$\sum_{i=0}^{10} x^i + i^2$$

# Example 9 : Les blocks!

## Viewing cone-Calibration plane intersection

- Plane of known eucliden structure
- Dense image-calibration plane matching
  - Structured light
- Intersection shape: calibration ellipse  
(Hyperbola case discussed in the paper)

## Un autre détail...

Et bien oui, il ne fait pas oublier le code!    `for(i=0;i<10;i++)`  
...

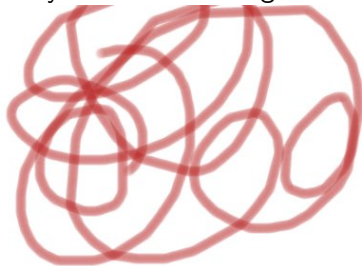
## Un autre détail...

Et aussi une équation...

$$\sum_{i=0}^{10} x^i + i^2$$

## Example 10: Woobling...

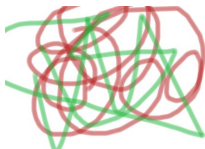
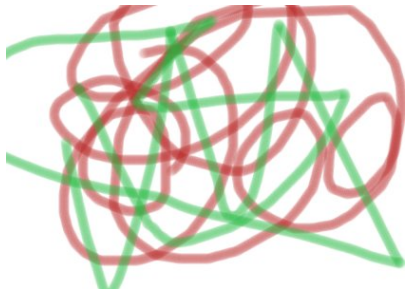
Some text for the first slide.  
Possibly several lines long.



On peut aussi faire...

## Example 10: Woobling...

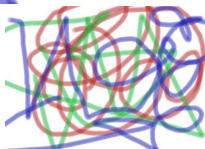
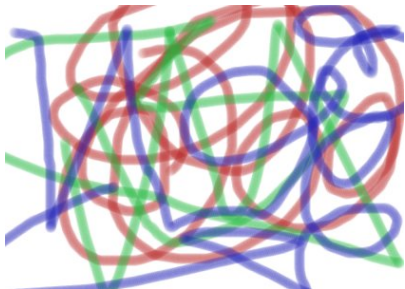
Replacement on the second slide. Supressed for handout.



On peut aussi faire...



## Example 10: Woobling...



On peut aussi faire...