Stereo Matching Using Graph Cuts

Flavien LÉGER, Thomas PUMIR DE LOUVIGNY, Pauline TAN

ENS Cachan, Department of Mathematics (France)

June, 24, 2010

Internship Supervisors
Jean-Michel MOREL, Neus SABATER
CMLA, Cachan (France)

Outline

Introduction

Given two images of the same scene, the goal in stereo is to compute the depth map of the reference image:

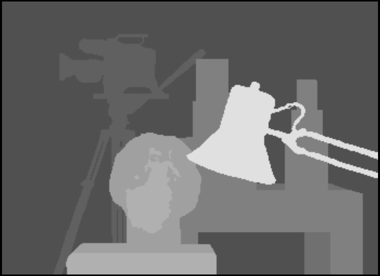
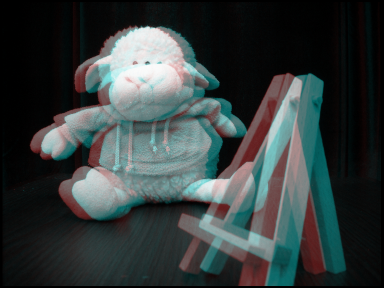


Fig.:(right) Ground truth of the image Tsukuba (the brighter the closer).

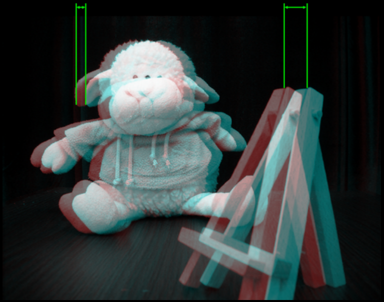
Disparity (with epipolar constraint)

We will assume that the axis of the two cameras are parallel.



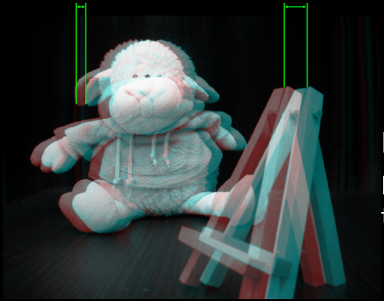
Disparity (with epipolar constraint)

We will assume that the axis of the two cameras are parallel.



Disparity (with epipolar constraint)

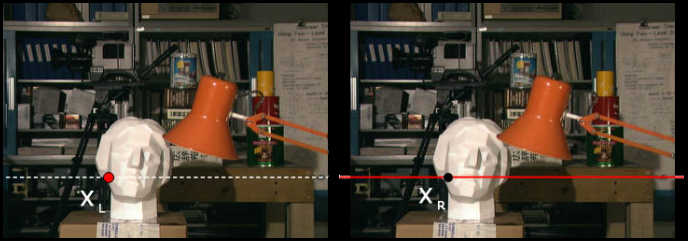
We will assume that the axis of the two cameras are parallel.



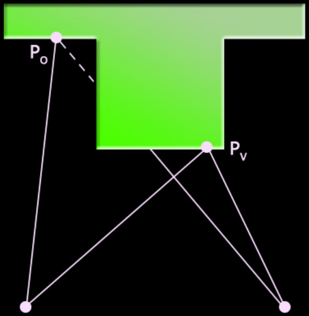
If pixel p in the reference image and pixel q in the other correspond, then the vector $q - p$ is called disparity.

The disparity of a point p is inversely proportional to its depth.

Aim: trying to match each pixel of the reference image with a pixel of the other image in order to compute its disparity.



Occlusion



Points which are not visible from one of the two cameras are called occluded.

Reference

[KZ01] V. Kolmogorov and R. Zabih. Computing Visual Correspondence with Occlusions using Graph Cuts. *International Journal of Computer Vision*, 2001.

Energy function

Assignment

- assignment: pair of pixels $\langle p, q \rangle$ which may potentially correspond, i.e. such that:

$$p_y = q_y \qquad \text{and} \qquad 0 \leq q_x - p_x \leq d$$

- $\mathcal{A}$: the set of assignments
- if $a = \langle p, q \rangle \in \mathcal{A}$, $d(a) := q - p$ (in terms of vectors)

Configuration

- A configuration f is a way to match pixels of the two images.
- active assignment: if two pixels p and q correspond, the assignement $\langle p, q \rangle \in \mathcal{A}$ is called active.
- Let $A(f)$ be the set of active assignments according to the configuration f.
- If $\langle p, q \rangle$ is active, $d_f(p) = d(\langle p, q \rangle)$ is the disparity of the point p.

Three terms energy function

Let f be a configuration. We define its energy:

$$E(f) = E_{\text{data}}(f) + E_{\text{occ}}(f) + E_{\text{smooth}}(f)$$

Three terms energy function

Let f be a configuration. We define its energy:

$$E(f) = E_{\mathrm{data}}(f) + E_{\mathrm{occ}}(f) + E_{\mathrm{smooth}}(f)$$

Three terms energy function

Let f be a configuration. We define its energy:

$$E(f) = E_{\text{data}}(f) + E_{\text{occ}}(f) + E_{\text{smooth}}(f)$$

data term:

$$E_{\text{data}}(f) = \sum_{a \in A(f)} D(a)$$

Three terms energy function

Let f be a configuration. We define its energy:

$$E(f) = E_{\text{data}}(f) + E_{\text{occ}}(f) + E_{\text{smooth}}(f)$$

data term:

$$E_{\text{data}}(f) = \sum_{a \in A(f)} D(a)$$

where for an assignment $a = \langle p, q \rangle$,

$$D(a) = (I(p) - I(q))^2$$

with I the intensity of the pixel.

Three terms energy function

Let f be a configuration. We define its energy:

$$E(f) = E_{\mathrm{data}}(f) + E_{\mathrm{occ}}(f) + E_{\mathrm{smooth}}(f)$$

occlusion term:

$$E_{\mathrm{occ}}(f) = \sum_{p \text{ occluded}} 2\,K$$

Three terms energy function

Let f be a configuration. We define its energy:

$$E(f) = E_{\mathrm{data}}(f) + E_{\mathrm{occ}}(f) + E_{\mathrm{smooth}}(f)$$

occlusion term:

$$E_{\mathrm{occ}}(f) = \sum_{p\ \mathrm{occluded}} 2\,K$$

where $2\,K$ is the occlusion penalty.
In [KZ01], K is chosen to be $5\,\lambda$. λ is a parameter.

Three terms energy function

Let f be a configuration. We define its energy:

$$E(f) = E_{\text{data}}(f) + E_{\text{occ}}(f) + E_{\text{smooth}}(f)$$

smoothness term:

$$E_{\text{smooth}}(f) = \sum_{(a_1,a_2)\in\mathcal{N}\cap A(f)\times A(f)^c} V_{a_1,a_2}$$

Three terms energy function

Let f be a configuration. We define its energy:

$$E(f) = E_{\text{data}}(f) + E_{\text{occ}}(f) + E_{\text{smooth}}(f)$$

smoothness term:

$$E_{\text{smooth}}(f) = \sum_{(a_1,a_2)\in\mathcal{N}\cap A(f)\times A(f)^c} V_{a_1,a_2}$$

- $\mathcal{N} \subset \mathcal{A} \times \mathcal{A}$ is a neighborhood system on assignments
- $(a_1, a_2) \in A(f) \times A(f)^c$ means that a_1 is active and a_2 is not
- In [KZ01], if $a_1 = \langle p, q \rangle$ and $a_2 = \langle r, s \rangle$ then

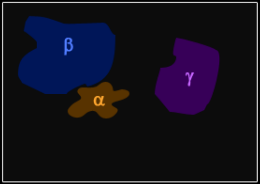
$$V_{a_1,a_2} = \begin{cases} 3\,\lambda & \text{if } \max(|I(p) - I(r)|, |I(q) - I(s)|) < 8 \\ \lambda & \text{otherwise} \end{cases}$$

Expansion moves

α–expansion move

Let f be a configuration and $\alpha \in [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ a disparity value. f' is an α–expansion move if the disparity $d_{f'}(p)$ of a point p which is not occluded is either $d_f(p)$ or α.

With an α–expansion move we potentially extend the set of points with disparity α.

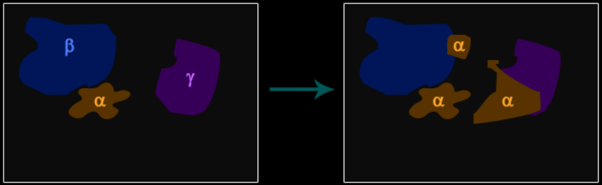


Expansion moves

α–expansion move

Let f be a configuration and $\alpha \in [x_{\text{min}}, x_{\text{max}}] \times [y_{\text{min}}, y_{\text{max}}]$ a disparity value. f' is an α–expansion move if the disparity $d_{f'}(p)$ of a point p which is not occluded is either $d_f(p)$ or α.

With an α–expansion move we potentially extend the set of points with disparity α.



Overview of the algorithm

Aim: Trying to reduce the energy by expansion moves.

Initialisation: Let f° be a configuration.

for each $\alpha \in [x_{\text{min}}, x_{\text{max}}] \times [y_{\text{min}}, y_{\text{max}}]$

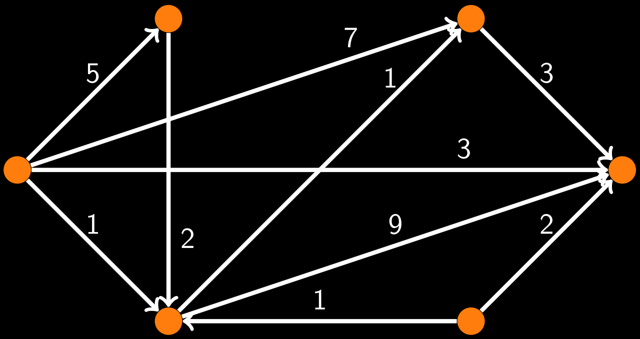
- realize an α–expansion move f'
- compute its energy $E(f')$
- if $E(f') < E(f^\circ)$ then do $f^\circ := f'$

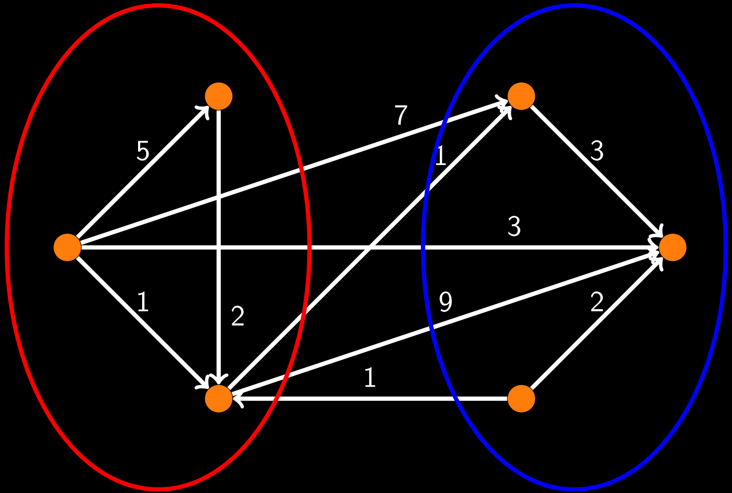
Graph Cut

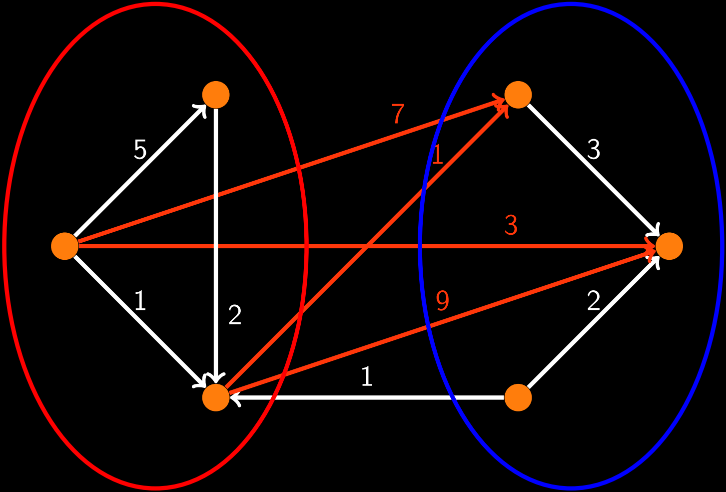
Aim: Constructing a graph in which a cut $\mathcal{C}$ is equivalent to an expansion move f and the cost of $\mathcal{C}$ equals $E(f)$ plus a constant.

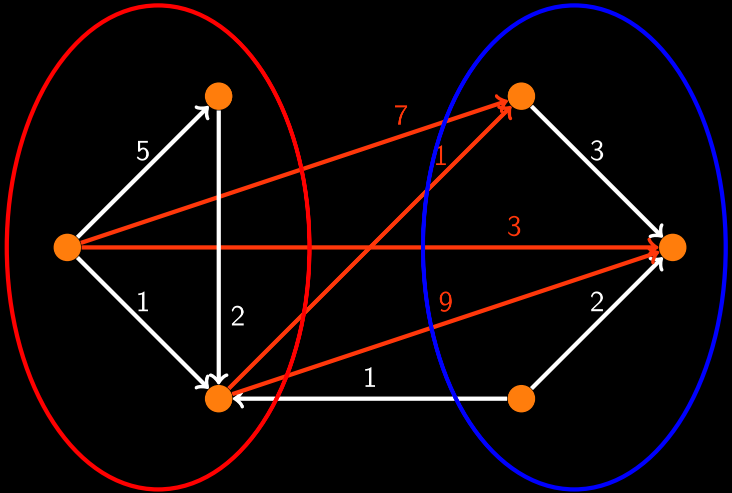
Graph cut

- Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a graph with two vertices s and t.
- A cut is a partition $(\mathcal{V}^s, \mathcal{V}^t)$ of $\mathcal{V}$, such that $s \in \mathcal{V}^s$ and $t \in \mathcal{V}^t$.
- The cost of the cut is the sum of the weight of the edges from a vertex in $\mathcal{V}^s$ to a vertex in $\mathcal{V}^t$.


7
5
1
3
3
1
2
9
2
1


5
7
1
3
3
1
2
9
2
1


5
7
1
3
3
1
2
9
2
1

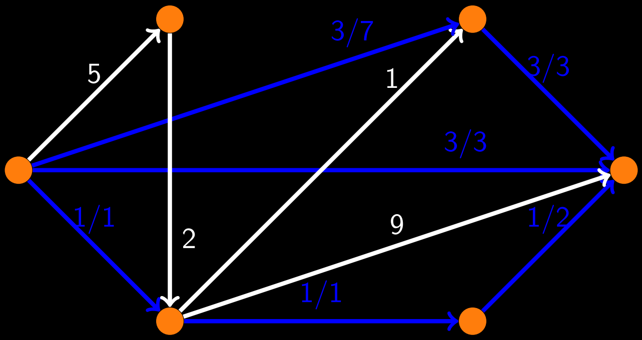


$$7 + 1 + 3 + 9 = 20$$

A minimum cut of a graph is a cut with the minimum cost.

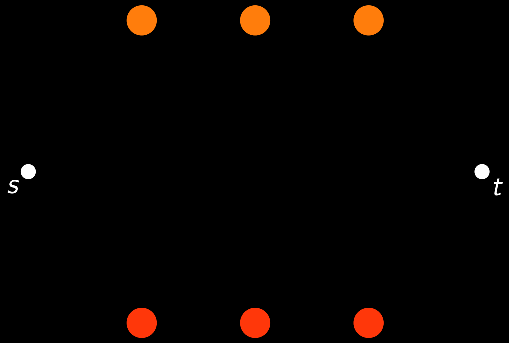
Max–Flow / Min–Cut

Thanks to Ford-Fulkerson algorithm, we are able to compute the minimum cut of a graph, by computing a maximum flow. [Ford-Fulkerson62]



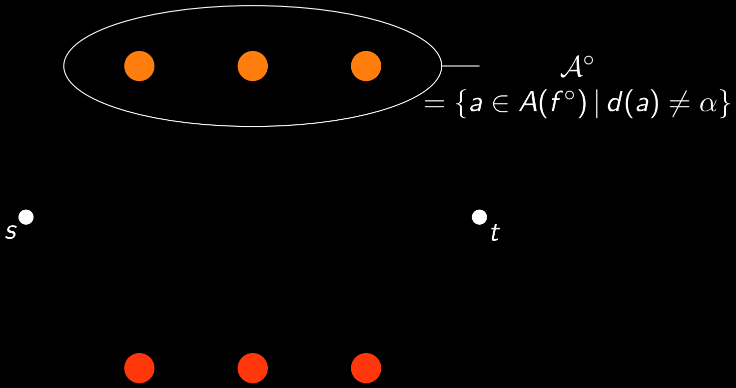
Graph structure

vertices:



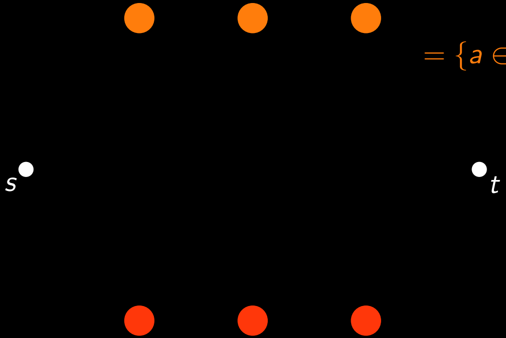
Graph structure

vertices:



Graph structure

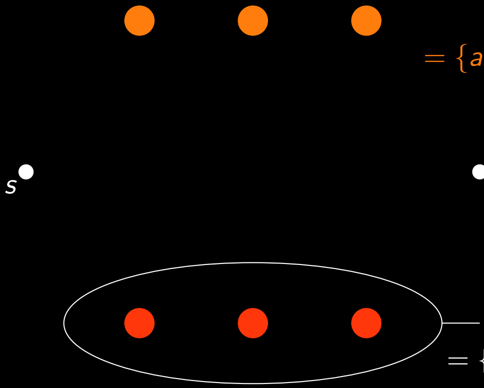
vertices:



$$\mathcal{A}^{\circ} = \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$$

Graph structure

vertices:

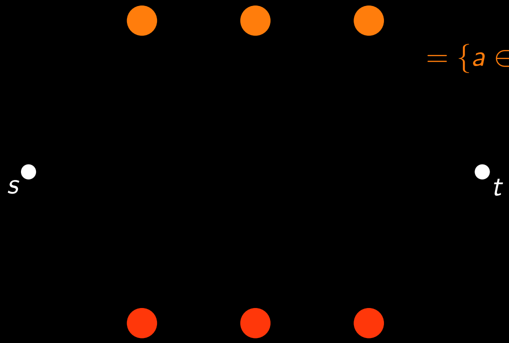


$$\mathcal{A}^{\circ} = \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$$

$$\mathcal{A}^{\alpha} = \{a \in \mathcal{A} \mid d(a) = \alpha\}$$

Graph structure

vertices:

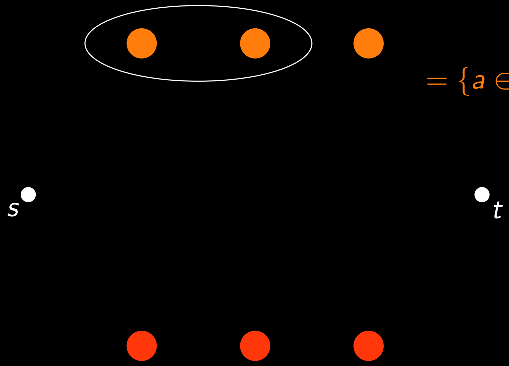


$$\mathcal{A}^{\circ} = \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$$

$$\mathcal{A}^{\alpha} = \{a \in \mathcal{A} \mid d(a) = \alpha\}$$

Graph structure

α–expansion move:

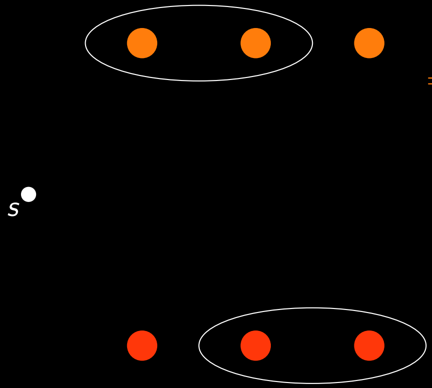


$$\mathcal{A}^{\circ} = \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$$

$$\mathcal{A}^{\alpha} = \{a \in \mathcal{A} \mid d(a) = \alpha\}$$

Graph structure

α–expansion move:

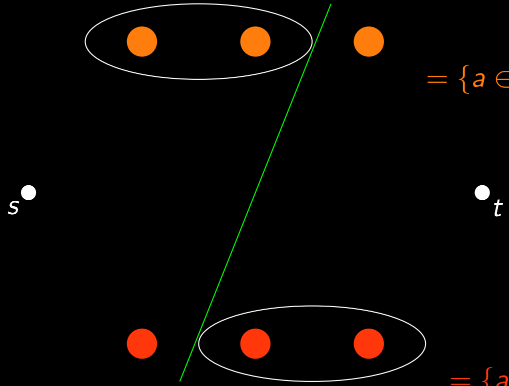


$$\mathcal{A}^{\circ} = \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$$

$$\mathcal{A}^{\alpha} = \{a \in \mathcal{A} \mid d(a) = \alpha\}$$

Graph structure

α–expansion move:

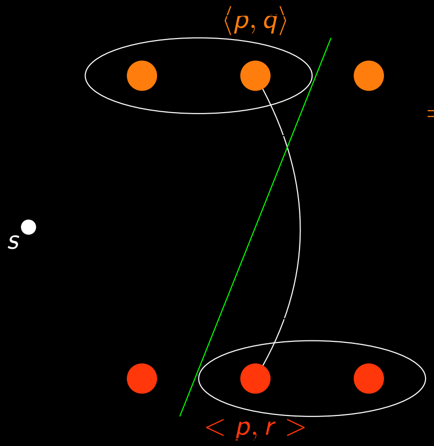


$$\mathcal{A}^{\circ} = \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$$

$$\mathcal{A}^{\alpha} = \{a \in \mathcal{A} \mid d(a) = \alpha\}$$

Graph structure

uniqueness constraint:

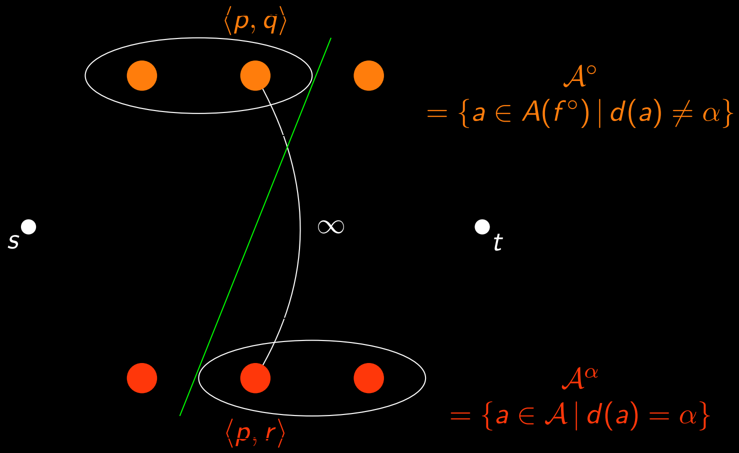


$\mathcal{A}^{\circ}$
$= \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$

$\mathcal{A}^{\alpha}$
$= \{a \in \mathcal{A} \mid d(a) = \alpha\}$

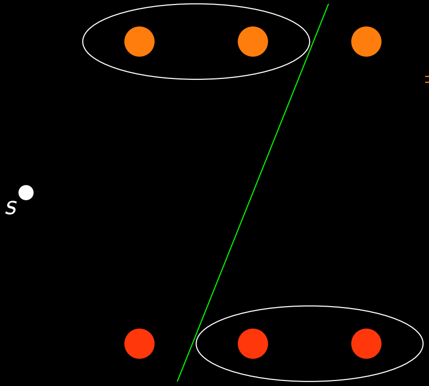
Graph structure

uniqueness constraint:



Graph structure

active assignment: data penalty

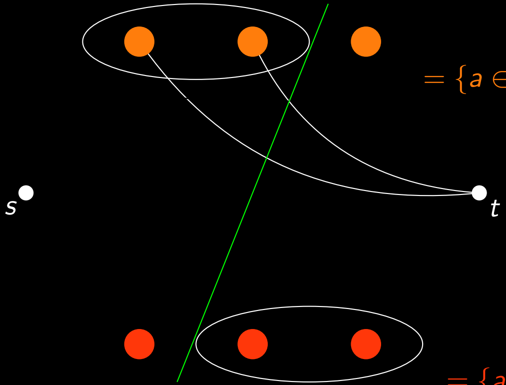


$$\mathcal{A}^{\circ} = \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$$

$$\mathcal{A}^{\alpha} = \{a \in \mathcal{A} \mid d(a) = \alpha\}$$

Graph structure

active assignment: data penalty

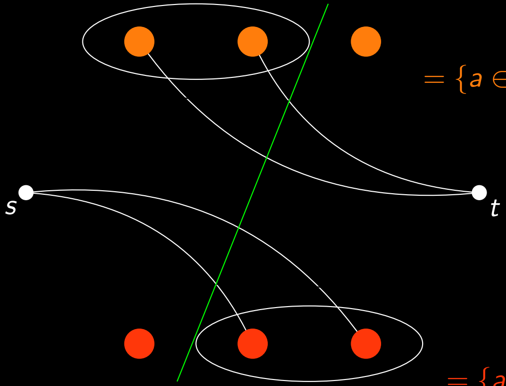


$\mathcal{A}^{\circ}$
$= \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$

$\mathcal{A}^{\alpha}$
$= \{a \in \mathcal{A} \mid d(a) = \alpha\}$

Graph structure

active assignment: data penalty

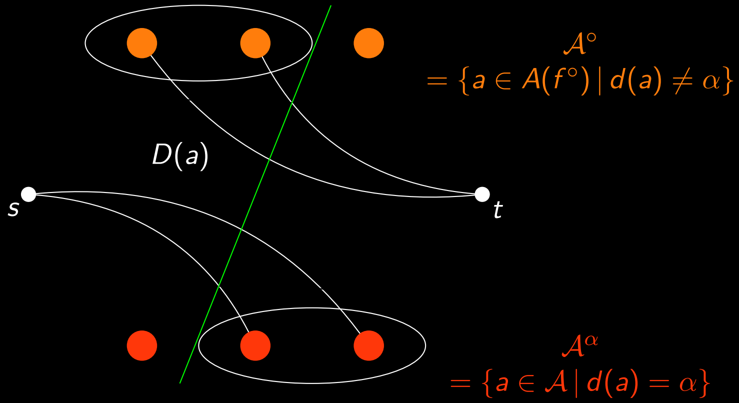


$\mathcal{A}^{\circ} = \{a \in A(f^{\circ}) \,|\, d(a) \neq \alpha\}$

$\mathcal{A}^{\alpha} = \{a \in \mathcal{A} \,|\, d(a) = \alpha\}$

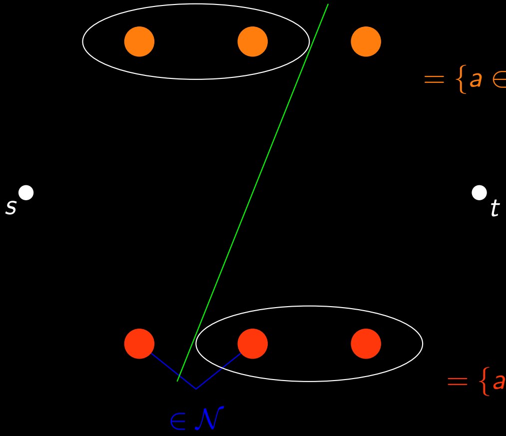
Graph structure

active assignment: data penalty



Graph structure

active assignment: smoothness penalty

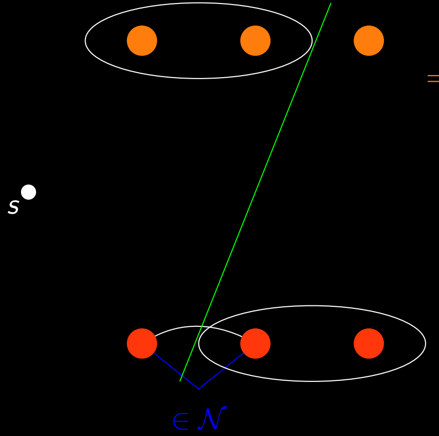


$\mathcal{A}^{\circ}$
$= \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$

$\mathcal{A}^{\alpha}$
$= \{a \in \mathcal{A} \mid d(a) = \alpha\}$

Graph structure

active assignment: smoothness penalty

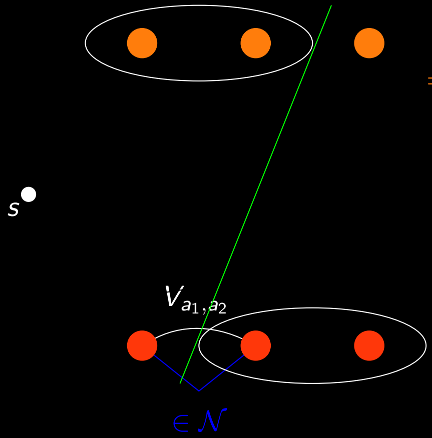


$$\mathcal{A}^{\circ} = \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$$

$$\mathcal{A}^{\alpha} = \{a \in \mathcal{A} \mid d(a) = \alpha\}$$

Graph structure

active assignment: smoothness penalty

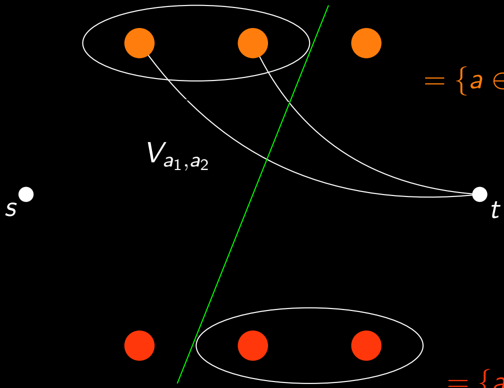


$\mathcal{A}^\circ$
$= \{a \in A(f^\circ) \,|\, d(a) \neq \alpha\}$

$\mathcal{A}^\alpha$
$= \{a \in \mathcal{A} \,|\, d(a) = \alpha\}$

Graph structure

active assignment: smoothness penalty

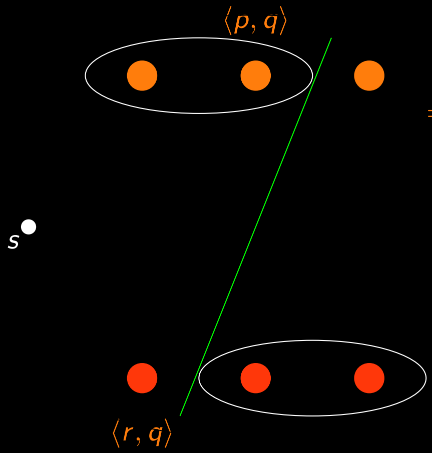


$\mathcal{A}^{\circ}$
$= \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$

$\mathcal{A}^{\alpha}$
$= \{a \in \mathcal{A} \mid d(a) = \alpha\}$

Graph structure

inactive assignment: occlusion penalty

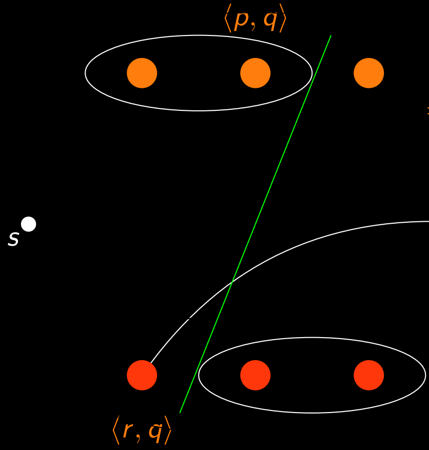


$$\mathcal{A}^{\circ} = \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$$

$$\mathcal{A}^{\alpha} = \{a \in \mathcal{A} \mid d(a) = \alpha\}$$

Graph structure

inactive assignment: occlusion penalty

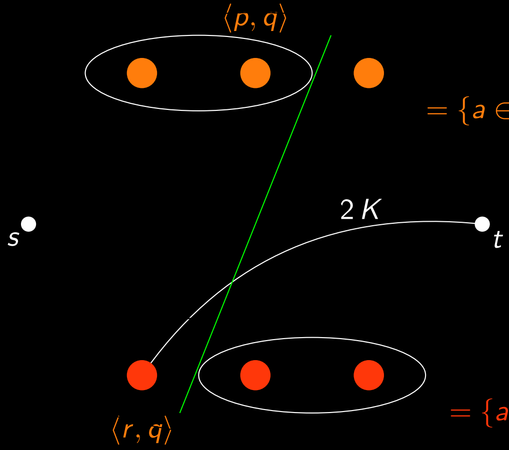


$$\mathcal{A}^{\circ} = \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$$

$$\mathcal{A}^{\alpha} = \{a \in \mathcal{A} \mid d(a) = \alpha\}$$

Graph structure

inactive assignment: occlusion penalty



$\mathcal{A}^{\circ}$
$= \{a \in A(f^{\circ}) \mid d(a) \neq \alpha\}$

$\mathcal{A}^{\alpha}$
$= \{a \in \mathcal{A} \mid d(a) = \alpha\}$

Let us summarize the construction of the edges.

edges	weight	for
(s, a)	$D_{\mathrm{occ}}(a)$	$a \in \mathcal{A}^\circ$
(a, t)	$D_{\mathrm{occ}}(a)$	$a \in \mathcal{A}^\alpha$
(a, t)	$D(a) + D_{\mathrm{smooth}}(a)$	$a \in \mathcal{A}^\circ$
(s, a)	$D(a)$	$a \in \mathcal{A}^\alpha$
(a_1, a_2) (a_2, a_1)	V_{a_1,a_2}	$\{a_1, a_2\} \in \mathcal{N},\ a_1, a_2 \in \tilde{A}$
(a_1, a_2)	∞	$p \in \mathcal{P},\ a_1 \in A^\circ\ a_2 \in \mathcal{A}^\alpha,$ $a_1, a_2 \in N_p(f)$
(a_2, a_1)	$2\,K$	$p \in \mathcal{P},\ a_1 \in \mathcal{A}^\circ,\ a_2 \in \mathcal{A}^\alpha,$ $a_1, a_2 \in N_p(f)$

Theorem

$\mathcal{C}$ is a cut on $\mathcal{G}$ if and only if the corresponding configuration is an α–expansion move of f°.

Theorem

Let $\mathcal{C}$ be a cut on $\mathcal{G}$. Then its cost equals the energy of the corresponding configuration plus a constant.

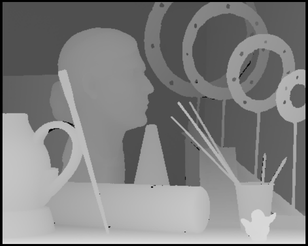
Theorem

Let $\mathcal{C}$ be a minimum cut on $\mathcal{G}$. Then the corresponding configuration is a configuration that minimizes the energy E.

Experimental results

Art (Middlebury Benchmark)

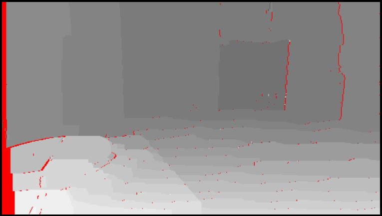
(b) Graph Cut



(c) Groundtruth

Portal

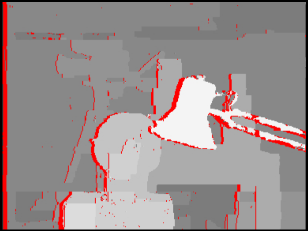
(d) Scene



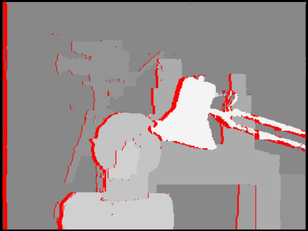
(e) Graph Cut

Dependance on parameters

(f) Scene



(g) Slicing in six



(h) Without slicing

Accomplished work

- Stereovision general concepts
- Kolmogorov & Zabih's algorithm
- Kolmogorov's thesis
- Publication of this algorithm on IPOL

ipol.im » edit » algo »

edition help

IPOL : Computing Visual Correspondence with Occlusions using Graph Cuts

WORKSHOP : This work may eventually be submitted for publication.

algorithm

Content

Contacts

- Flavien Léger flavien.leger[AT]ens-cachan.fr
- Jean-Michel Morel morel[AT]cmla.ens-cachan.fr
- Thomas Pumir pumir.thomas[AT]gmail.com
- Neus Sabater sabater[AT]cmla.ens-cachan.fr
- Pauline Tan pauline.tan[AT]ens-cachan.fr

References

1. V. Kolmogorov and R. Zabih, *Computing Visual Correspondence with Occlusions using Graph Cuts*. IEEE International Conference on Computer Vision (ICCV), July 2001.
2. V. Kolmogorov, *Graph Based Algorithms for Scene Reconstruction from Two or More Views*. PhD thesis, Cornell University, September 2003.
3. Y. Boykov and V. Kolmogorov, *An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision*. IEEE Transactions on Pattern Analysis and Machine Intelligence, September 2004.

Overview

The goal of stereovision is to find the 3-dimensional shape of a scene, given two photographies taken simultaneously from two different points of view.

IPOL : `www.ipol.im`

References

[BM08] J.A. Bondy and U.S.R. Murty. *Graph Theory*. Springer. 2008.

[BT96] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *Technical report*, 1996.

[BT98] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, April 1998.

[BK04] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision . *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, September 2004.

[BVZ01] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, November 2001.

[BBH03] M. Brown, D. Burschka, and G. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, August 2003.

[CP09] V. Caselles and N. Papadakis. Multi-label depth estimation for graph cuts stereo problems. *Technical report*, 2009.

[Ch75] N. Christofides. *Graph Theory – an Algorithmic Approach*. Academic Press. 1975.

[CM02] D. Comaniciu and P. Meer. *Mean shift : A robust approach toward feature space analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence. 24, May 2002.

[FF62] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press. 1962.

[Ko03] V. Kolmogorov. *Graph Based Algorithms for Scene Reconstruction from Two or More Views*. PhD thesis. Cornell University. September 2003.

[KZ01] V. Kolmogorov and R. Zabih. Computing Visual Correspondence with Occlusions using Graph Cuts. *International Journal of Computer Vision*, 2001.

[KZ02] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts?. *European Conference on Computer Vision (ECCV)*, May 2002.

[SS02] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondance Algorithms. *International Journal of Computer Vision*, 47, 2002.

[SSZ03] H.-Y. Shum, J. Sun, and N.-N. Zheng. *Stereo matching using belief propagation*. IEEE Transactions on Pattern Analysis and Machine Intelligence. 25, July 2003.

[SK10] Petter Strandmark and Fredrik Kahl. *Parallel and distributed graph cuts by dual decomposition*. In Conference on Computer Vision and Pattern Recognition. 2010.