Seminar: Numerical Algorithms for Image Analysis

Patrick Koberski

Fast Approximate Energy Minimization via Graph Cuts

Summer Term 2007

Saarland University, Germany
padekobeuni@gmx.de

Contents

- Motivation
- Energy function
- Distance function
- What are graph cuts (1-D)?
- In this context:
  - What is an α-β swap?
  - What is an α-expansion?
- Presentation of the algorithms
- Some examples

Motivation

- Many observed images are often noisy
- $\exists$ several methods to reduce the noise but various of them are slow (simulated annealing)
- If image has many local minima:
  - Graph cuts can be used to find the global one
- Today:
  Presentation of two (similar) algorithms that use graph cuts
- Efficient with respect to two large moves:
  - α-β swap
  - α-expansion

Energy Function

- **Given:**
  Noisy image with pixels $p \in P$ initialized with labels $f_p \in L$
  - For image restauration: L represents intensities

- **Goal:**
  Finding a labeling f that assignes each pixel p a label $f_p \in L$
  - f has to be piecewise smooth and consistent with the observed data.

- **Term of energy minimization:**

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

- $E_{data}(f) = \sum_{p \in P} D_p(f_p)$ with $D_p(f_p) = (f_p - i_p)^2$ where i = intensity

- $E_{smooth}(f) = \sum_{\{p,q\} \in N} V_{\{p,q\}}(f_p, f_q)$ with selected distance function V (N = set of pairs of adjacent pixels)

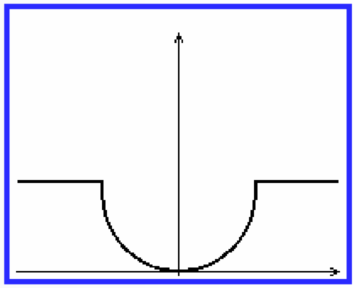
Energy Function (2)

- Choice of E_{smooth} and V respectively important
- Differentiation of V in **2 classes**:
  - V metric
  - V semimetric

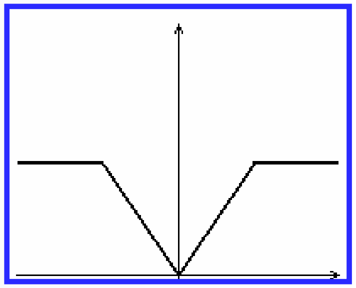
- **Definitions:**
  - V **semimetric** $\Leftrightarrow \begin{cases} V(\alpha,\beta) = 0 \Leftrightarrow \alpha = \beta \\ V(\alpha,\beta) = V(\beta,\alpha) \geq 0 \end{cases}$
  - V **metric** $\Leftrightarrow$ V semimetric $\wedge\ V(\alpha,\beta) \leq V(\alpha,\gamma) + V(\gamma,\beta)$

Distance Functions
Examples

- **Semimetric:**
  - $V(\alpha,\beta) = \min(K, \|\alpha-\beta\|^2)$, a **truncated quadratic** distance

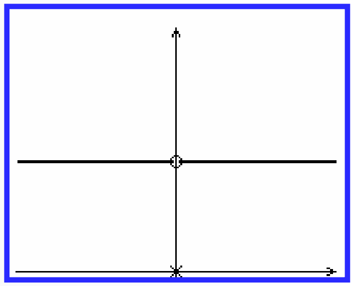


- **Metric:**
  - $V(\alpha,\beta) = \min(K, \|\alpha-\beta\|)$, a **truncated absolute** distance



  - $V(\alpha,\beta) = K \cdot \delta(\alpha \neq \beta)$, with $\delta(x) = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{else} \end{cases}$

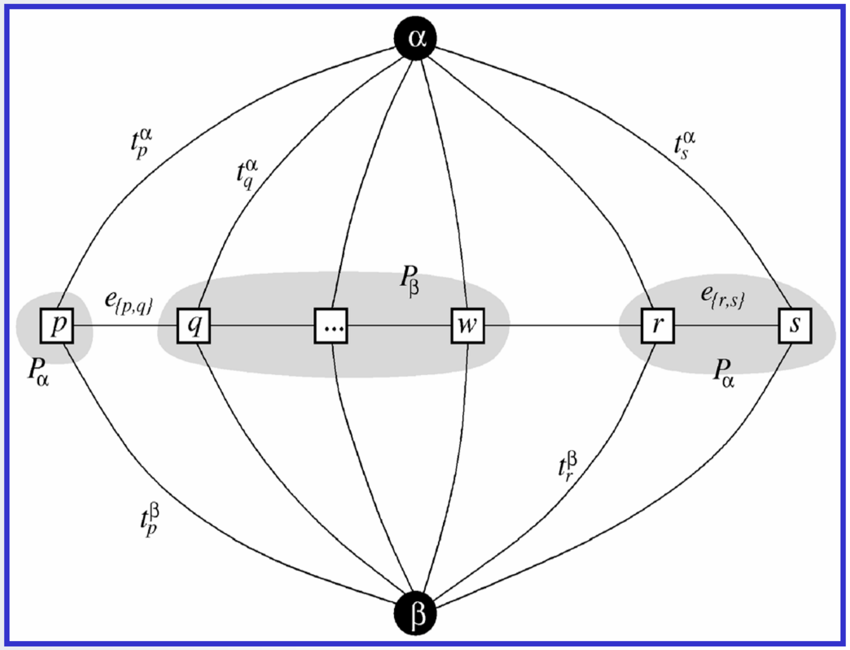
    also called **Potts model**



α-β swap
Graph

- If V semimetric → α-β swap (= Reassignment of labels α and β)
- **Given:**

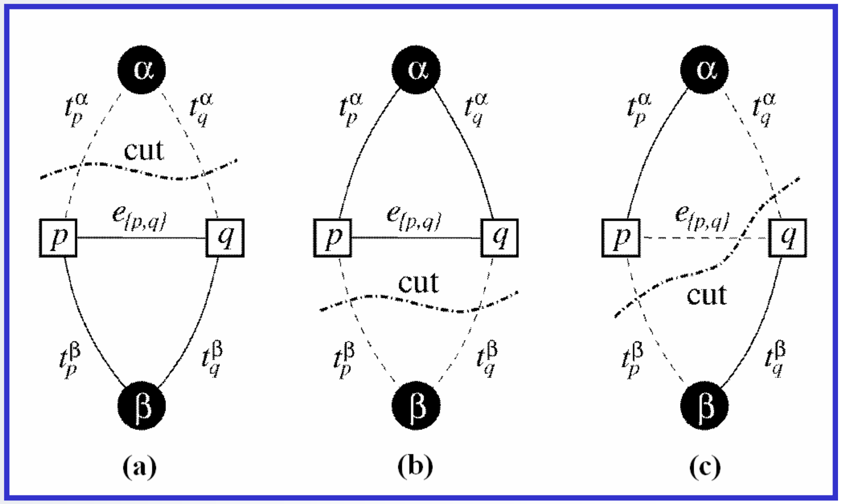
  1-D image with pixels $p \in P$
- α-β swap: subgraph $G_{\alpha\beta}$ consists of nodes: $p, q, \ldots, w \in P_{\alpha\beta}$
- Each pixel is connected with labels (here: α and β) (t-links)
- Adjacent pixels are connected with an edge (n-links)
- Each pixel p corresponds only to one label. I.e. $p \in P_\alpha$ or $p \in P_\beta$
- All edges have weights



α-β swap
Graph Cuts

- **Only allowed:** Separation of one pixel from one label!

- Only 4 possibilities:
  - If $t_p^{\alpha}, t_q^{\alpha} \in C$ then $e_{\{p,q\}} \notin C$ (a)
  - If $t_p^{\beta}, t_q^{\beta} \in C$ then $e_{\{p,q\}} \notin C$ (b)
  - If $t_p^{\beta}, t_q^{\alpha} \in C$ then $e_{\{p,q\}} \in C$ (c)
  - If $t_p^{\alpha}, t_q^{\beta} \in C$ then $e_{\{p,q\}} \in C$,(d)‘



- **Impact:**

$$e_{\{p,q\}} \in C \iff E_{smooth} \text{ has an effect on } E(f), \text{ otherwise } E_{smooth} = 0$$

α-β swap
Properties

- Leads to a labeling f^{C} corresponding to a cut C on $G_{\alpha\beta}$:

$$\boxed{f_p^{C} = \begin{cases} \alpha & \text{if } t_p^{\alpha} \in C \text{ for p} \in P_{\alpha\beta} \\ \beta & \text{if } t_p^{\beta} \in C \text{ for p} \in P_{\alpha\beta} \\ f_p & \text{for p} \in P, \text{p} \notin P_{\alpha\beta} \end{cases}}$$

- **Lemma 1:**

  A labeling f^{C} corresponding to a cut C on $G_{\alpha\beta}$ is one α-β swap away from the initial labeling f.

- **Furthermore:**

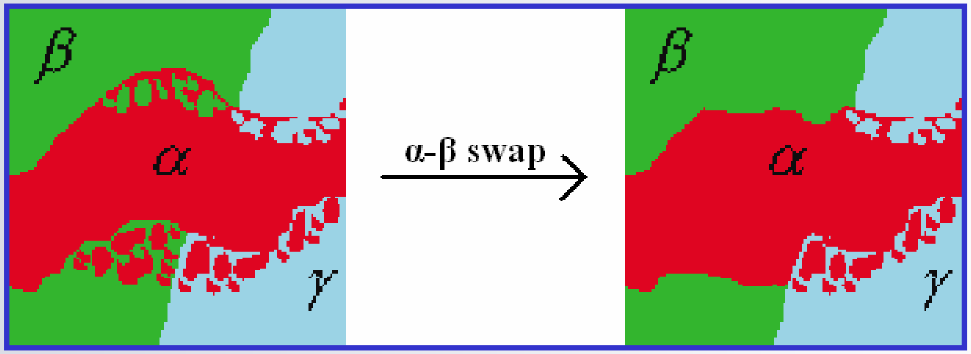
  The cost of a cut C on $G_{\alpha\beta}$ is $|C| = E(f^{C}) + h$ (h constant).

α-β swap
Result

- **Corollary 1:**

The optimal α-β swap from f is f^C where C is the minimum cut on $G_{\alpha\beta}$.

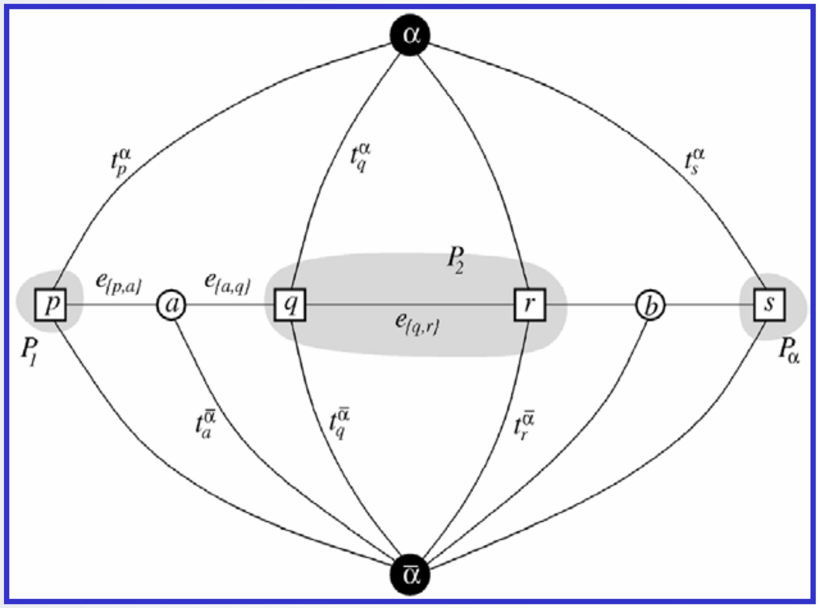
- **Note:** Minimal cut C = cut with smallest costs whereas $|C| = \sum_{e \in C} weight(e)$
- **Example** of an α-β swap (schematical illustration):



α-expansion
Graph

- If V metric → α-expansion
- Analogue to graph cuts for an α-β swap
- Notice following **differences:**

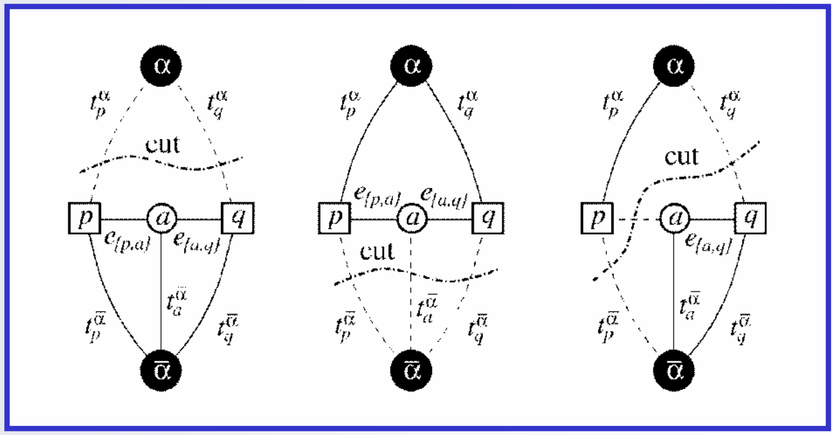
- The label to be extended is α, all the other labels are combined in one label called $\bar{\alpha}$
- Graph G_α consists of additional nodes $a,b,\ldots$ with corresponding edges $t_a^{\bar{\alpha}}, t_b^{\bar{\alpha}}, \ldots$ to $\bar{\alpha}$ introduced at the boundaries between partition sets P_l for $l \in L$



α-expansion

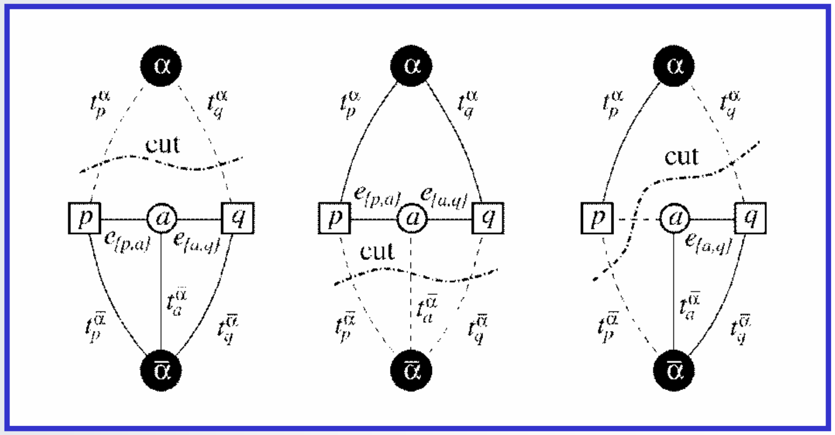
Graph Cuts and Results

- Graph cuts has to be performed in this way (again 4 possibilities):



α-expansion
Graph Cuts and Results

- Graph cuts has to be performed in this way (again 4 possibilities):

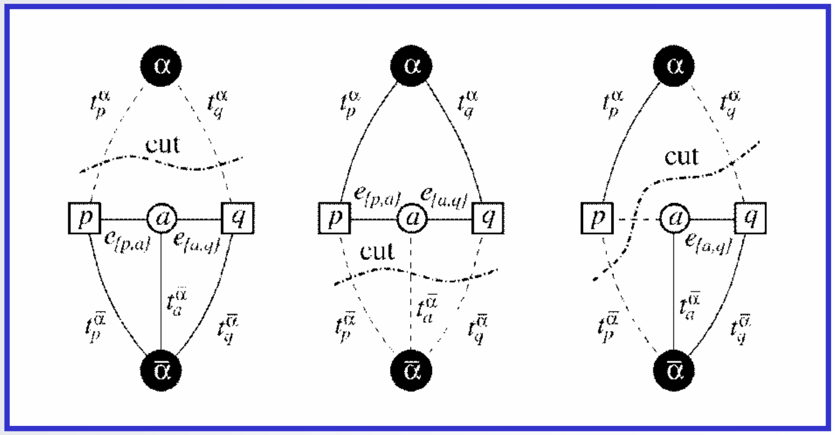


- Similar result like for an α-β swap:

The optimal α-expansion from f is f^{C} where C is the minimum cut on G_{α}.

α-expansion
Graph Cuts and Results

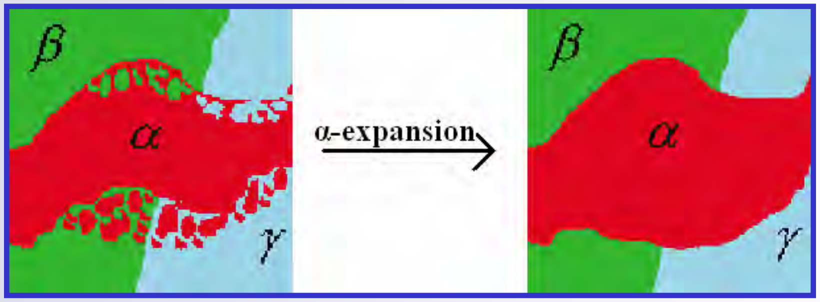
- Graph cuts has to be performed in this way (again 4 possibilities):



- Similar result like for an α-β swap:

> The optimal α-expansion from f is f^{C} where C is the minimum cut on G_{α}.

- **Example:** Schematical illustration of an α-expansion:



Algorithms

α-β swap move algorithm:

1. Start with an arbitrary labeling f
2. Set success $:= 0$
3. For each pair of labels $\{\alpha, \beta\} \subset L$
   - 3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α - β swap of f
   - 3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$
4. If success $= 1$ goto 2
5. Return f

α-expansion move algorithm:

1. Start with an arbitrary labeling f
2. Set success $:= 0$
3. For each label $\alpha \in L$
   - 3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α - expansion of f
   - 3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$
4. If success $= 1$ goto 2
5. Return f

- The algorithms are quiet similar in their structure

Algorithms

α-β swap move algorithm:

1. Start with an arbitrary labeling f
2. Set success $:= 0$
3. For each pair of labels $\{\alpha, \beta\} \subset L$
   3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α - β swap of f
   3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$
4. If success $= 1$ goto 2
5. Return f

α-expansion move algorithm:

1. Start with an arbitrary labeling f
2. Set success $:= 0$
3. For each label $\alpha \in L$
   3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α - expansion of f
   3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$
4. If success $= 1$ goto 2
5. Return f

- The algorithms are quiet similar in their structure
- Difference: Use of α-β swap and α-expansion respectively in 3.1.
- Both allow a large number of pixels to change their labels simultaniously

Overview

Comparison of standard move, α-β swap, α-expansion

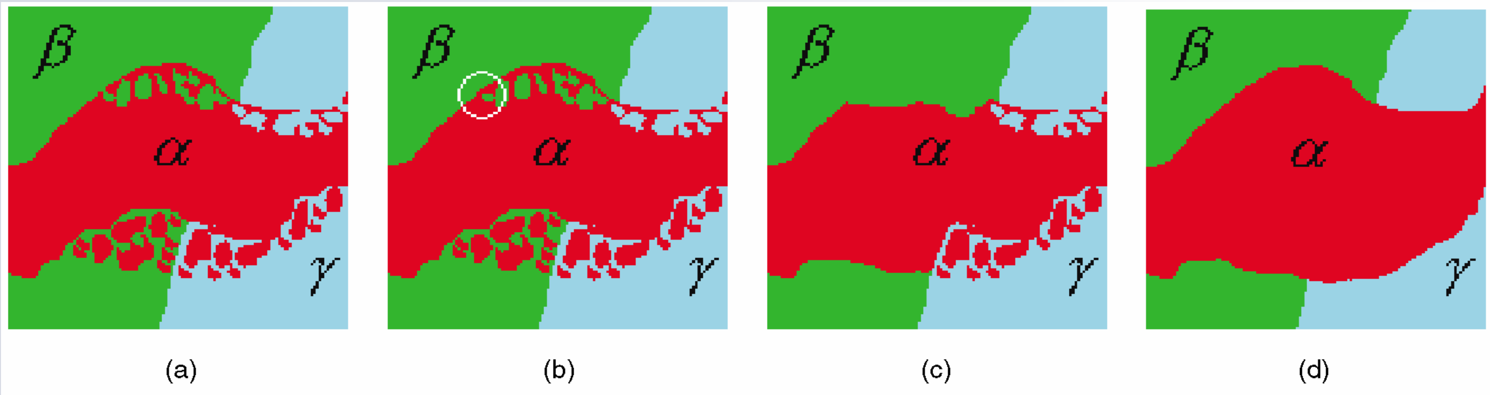


Fig. 1: Comparison of standard and large moves from a given initial labeling (a). The number of labels is $|L| = 3$.

(a) Initial labeling

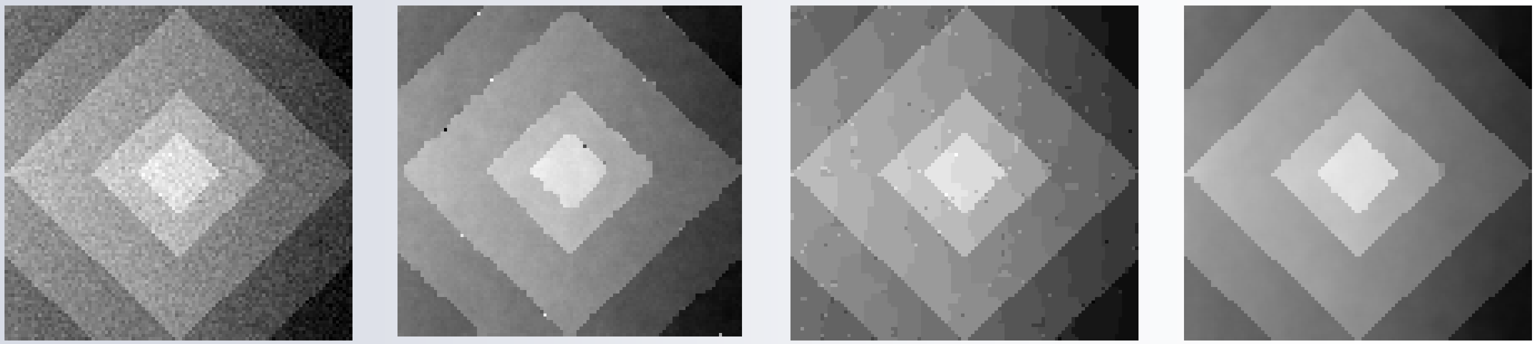
(b) Standard move like ICM (Iterative Conditional Modes) or annealing

→ change of one single pixel

(c) α-β swap

(d) α-expansion

Example (1)



(a) (b) (c) (d)

Fig. 2: Diamond image

(a) Noisy diamond image

Energy minimization with:

(b) Simulated annealing

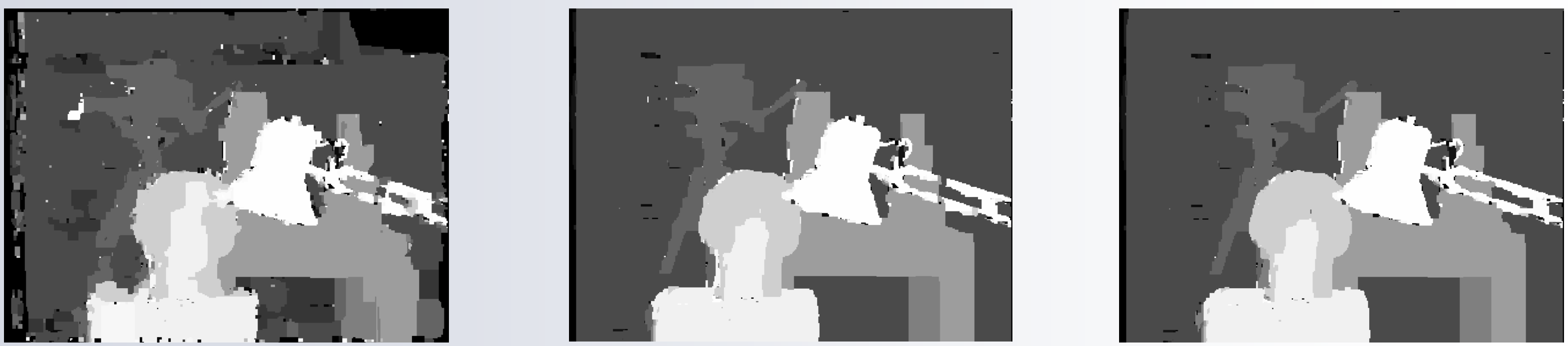
(c) α-expansion move

(d) α-β swap move

Example (2)

Real Stereo Imagery

(a) (b)



(c) (d) (e)

Fig. 3: Original image(-pair) (a) with ground truth (b). (c) Simulated annealing, (d) α-expansion, (e) α-β swap

In this case: The labels $l \in L$ represent disparities

Summary

- The presented algorithms minimise an energy with data and smoothness term
- They use graph cuts to generate a minimum with respect to very large moves (α-expansion and α-β swap)
- Graph cuts can be a good tool for solving miscellaneous computer vision problems
- The presented methods can solve computer vision problems such as image restauration, stereo and motion

References

- Fast Approximate Energy Minimization via Graph Cuts
  Yuri Boykov, Olga Veksler, Ramin Zabih
  In *International Conference on Computer Vision* (ICCV), vol. I, pp. 377--384, 1999.

- Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin
  In *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, pp. 177--180, 1993

- Segmentation by grouping junctions
  H. Ishikawa, D. Geiger
  In *IEEE Conference on Computer Vision and Patern Recognition*, pp. 125--131, 1998

- http://de.wikipedia.org

Questions?

Thank you for listening!