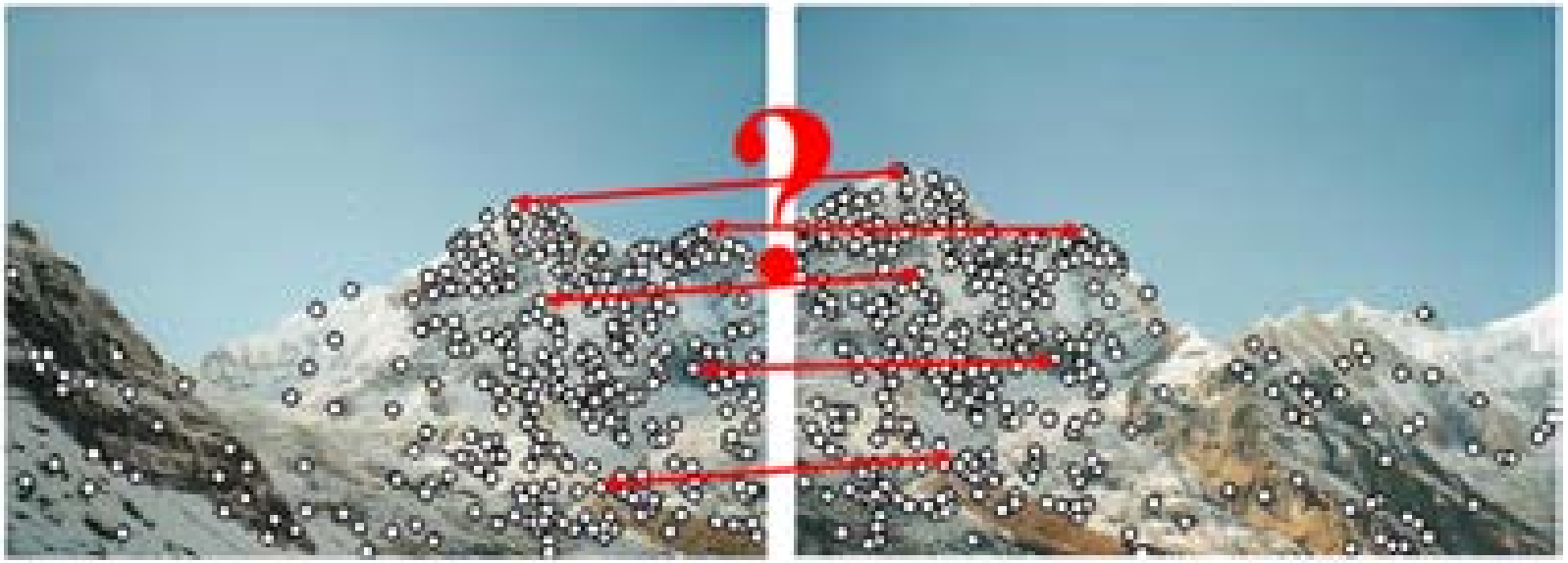
Graph Matching em Redes de Segmentos de Imagens

Gustavo T. Pfeiffer

Graph Matching em Redes de ~~Segmentos~~ Pontos de Imagens

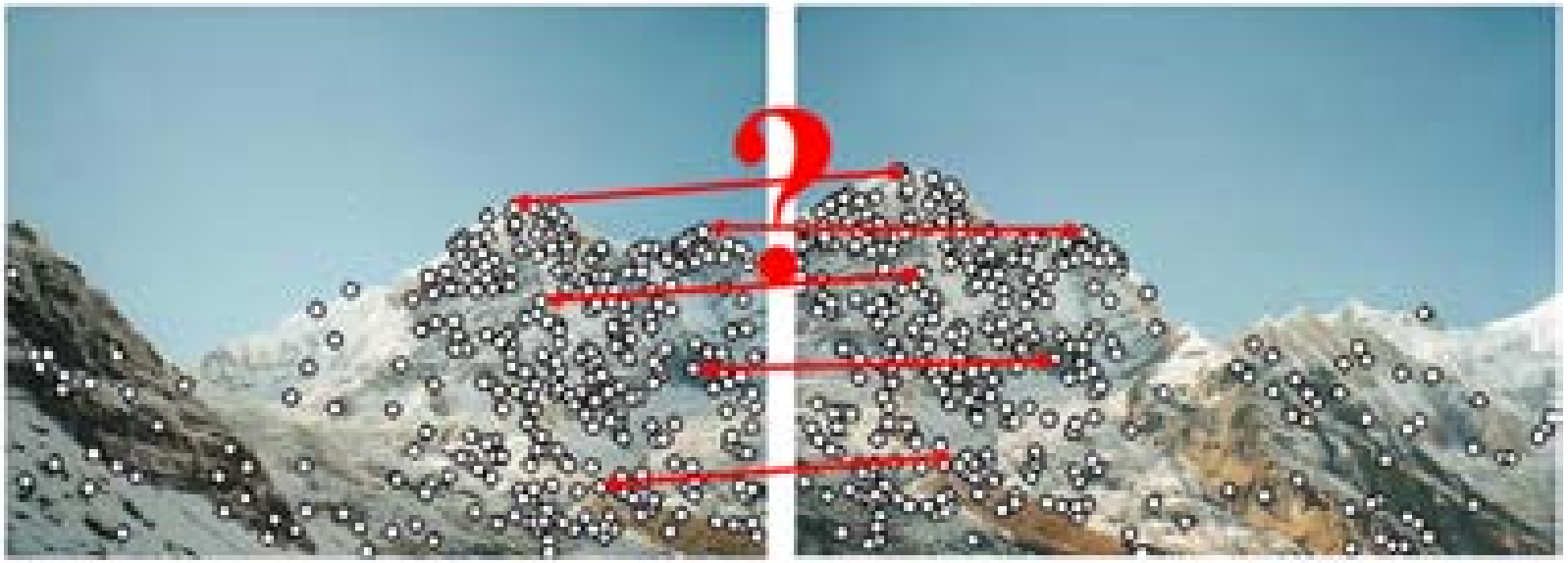
Gustavo T. Pfeiffer

Feature Matching



(Imagem de R. Szeliski, Computer Vision: Algorithms and Applications, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.)

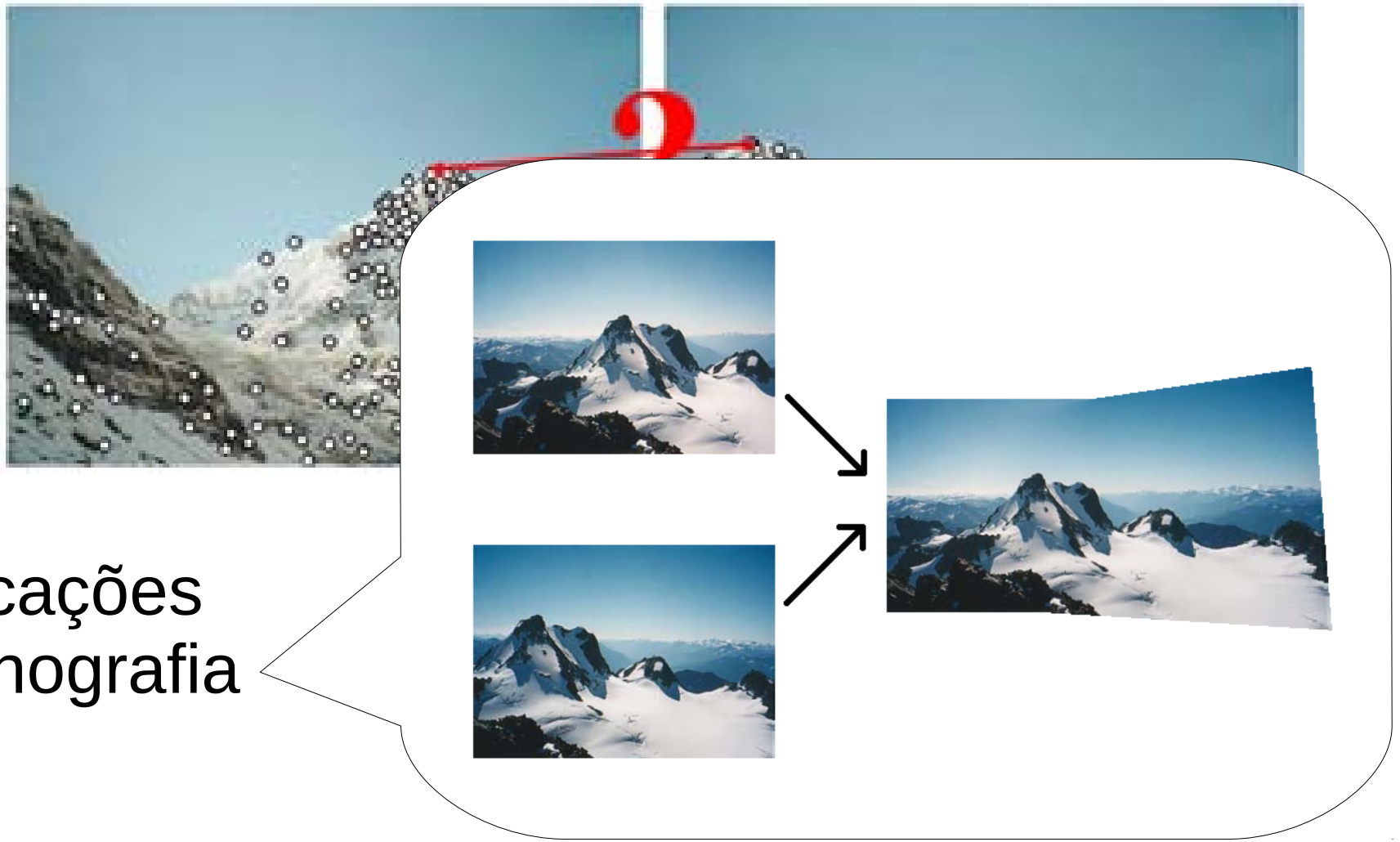
Feature Matching



Aplicações

- Panografia

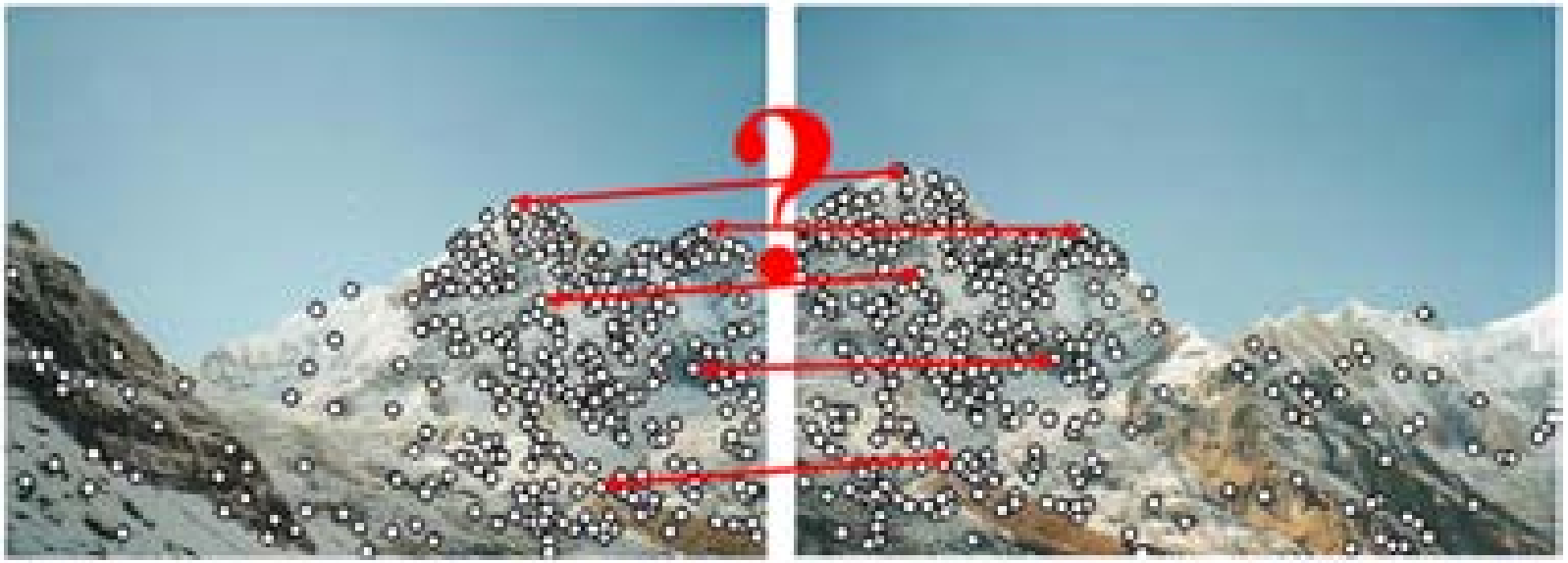
Feature Matching



Aplicações
- Panografia

(Imagem de R. Szeliski, Computer Vision: Algorithms and Applications, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.)

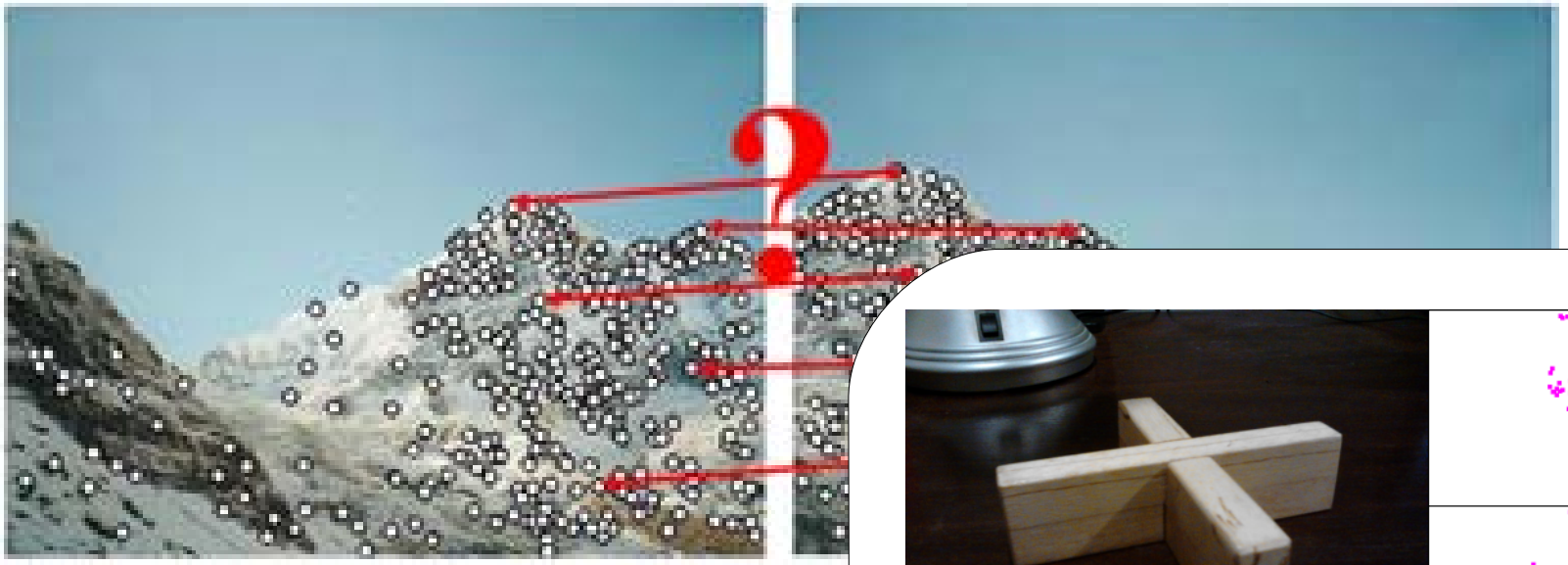
Feature Matching



Aplicações

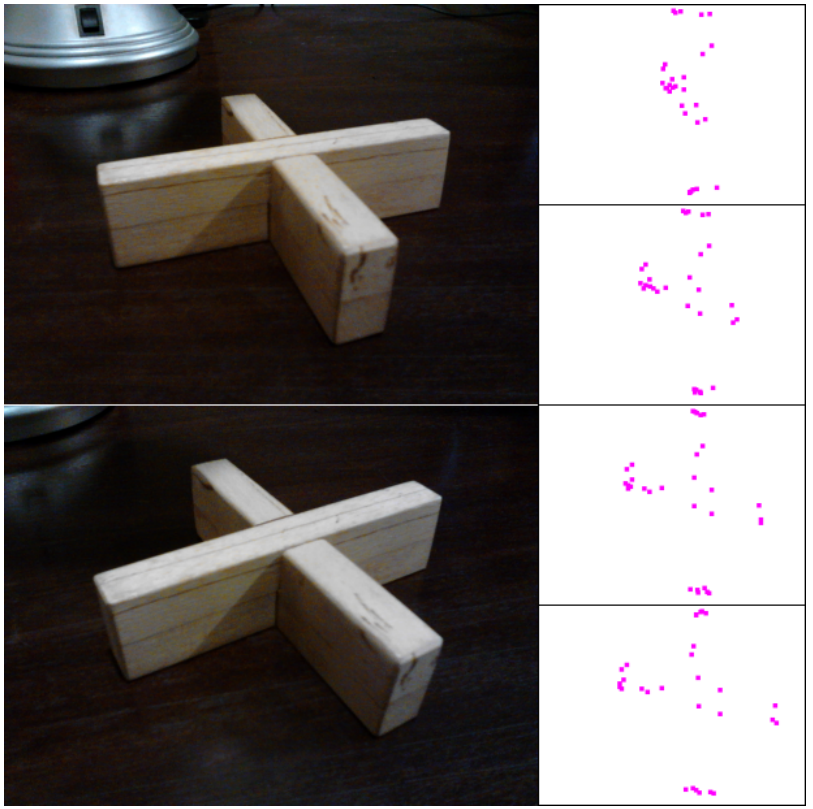
- Panografia
- Reconstrução estéreo

Feature Matching

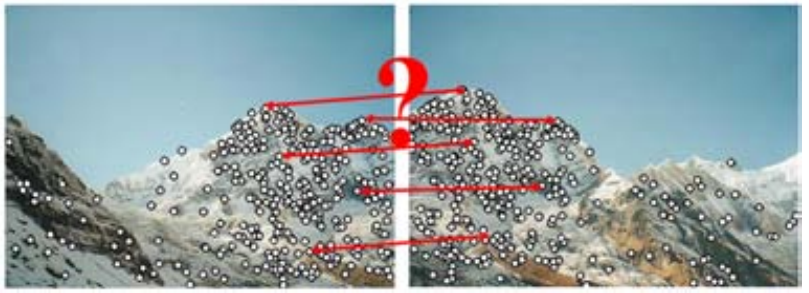


Aplicações

- Panografia
- Reconstrução estéreo

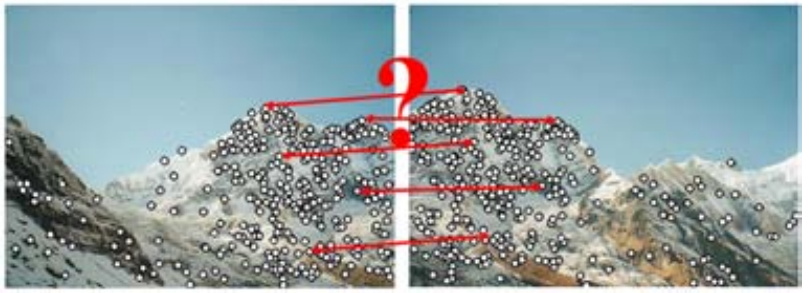


Como resolver?

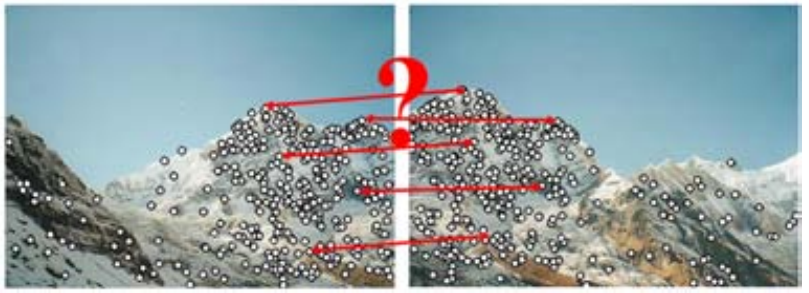


Como resolver?

- #1

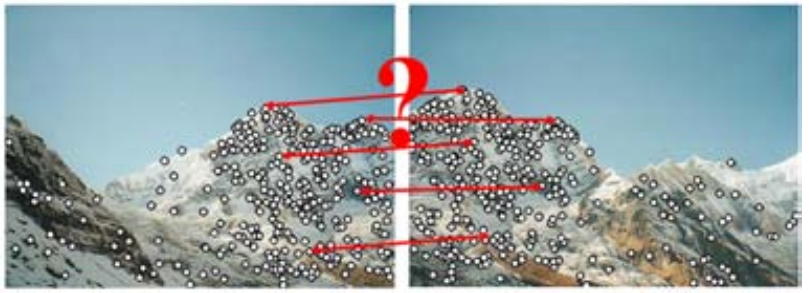


Como resolver?

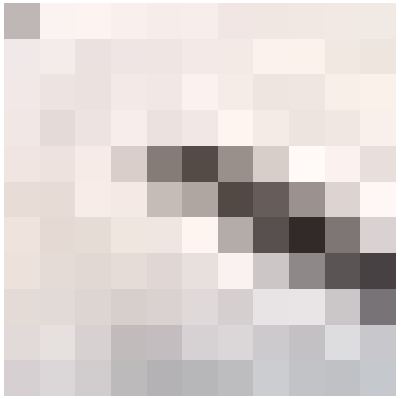


- #1
  - Comparar vizinhanças das *features* (NCC)

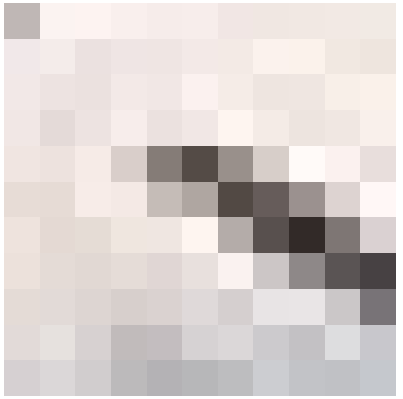
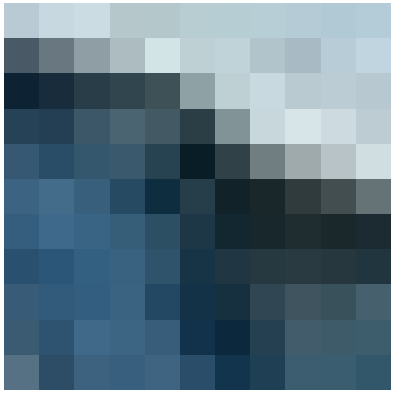
Como resolver?



- #1
  - Comparar vizinhanças das *features* (NCC)

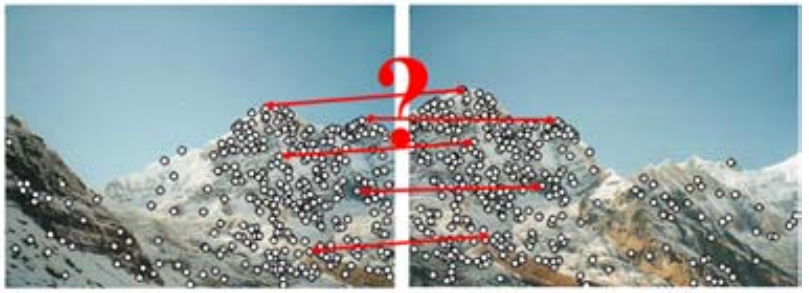



potencialmente a mesma *feature*

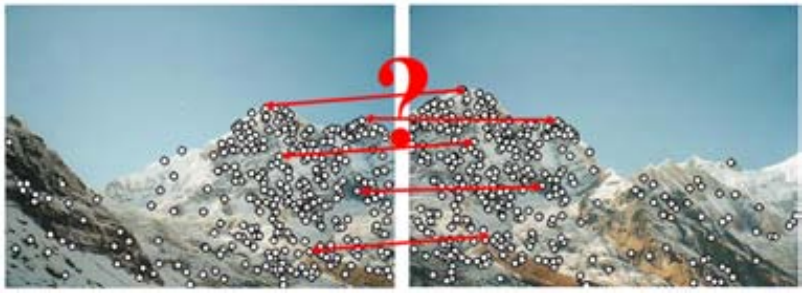
potencialmente *features* diferentes

Como resolver?

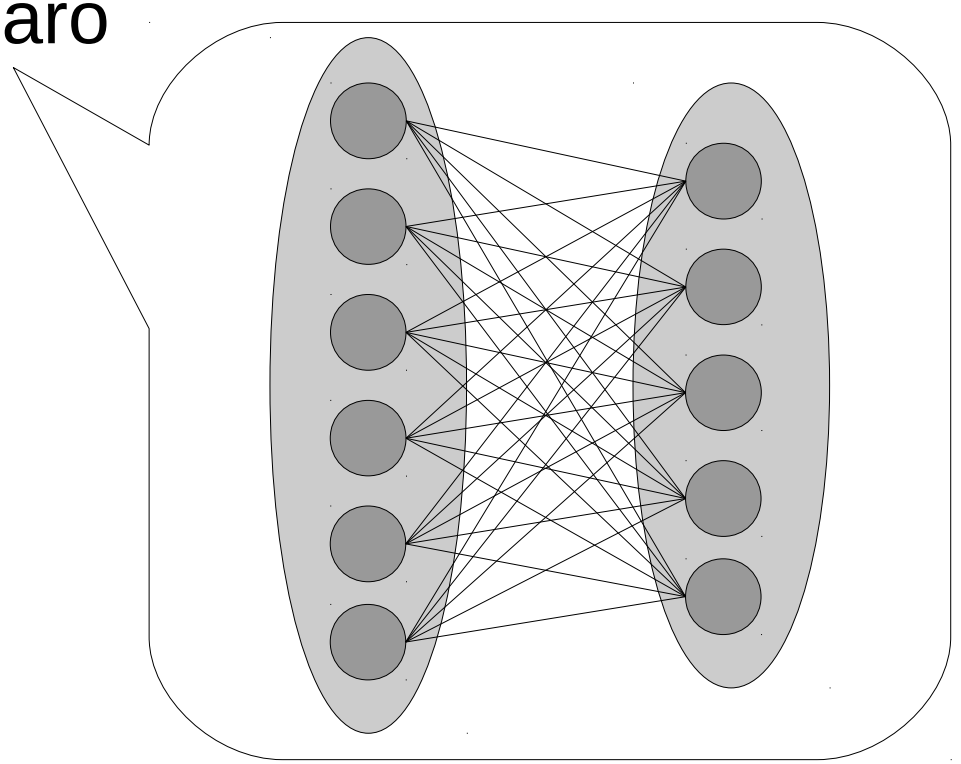


- #1
  - Comparar vizinhanças das *features* (NCC)
  - Aplicar o algoritmo húngaro

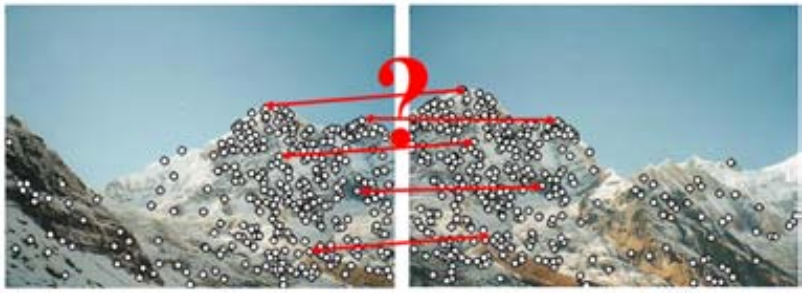
Como resolver?



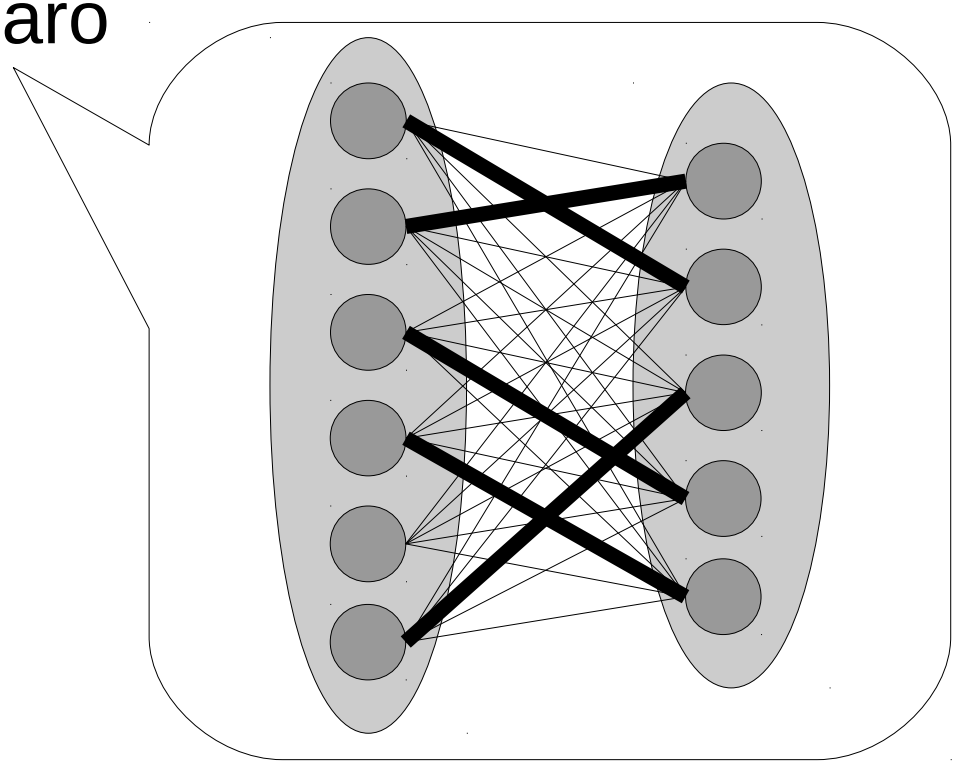
- #1
  - Comparar vizinhanças das *features* (NCC)
  - Aplicar o algoritmo húngaro



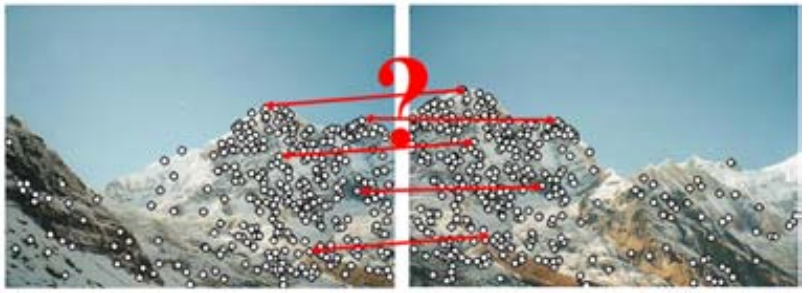
Como resolver?



- #1
  - Comparar vizinhanças das *features* (NCC)
  - Aplicar o algoritmo húngaro

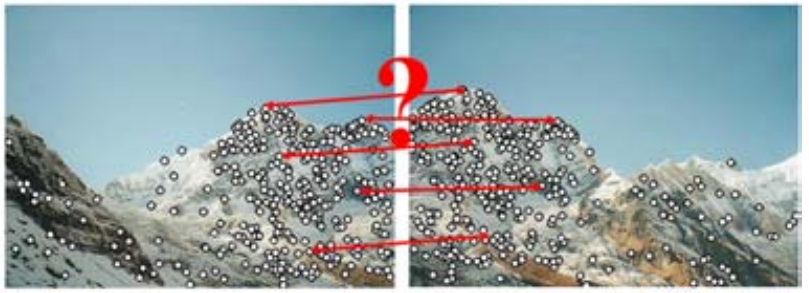


Como resolver?



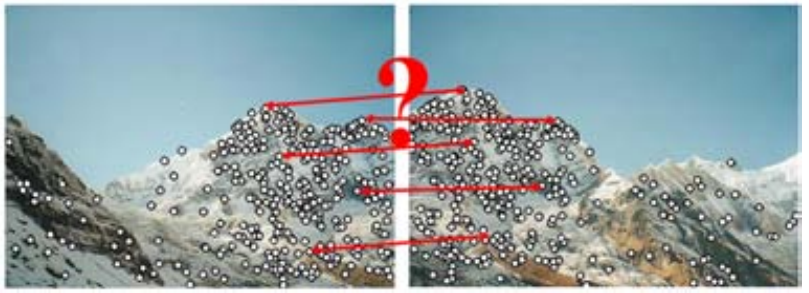
- #1
  - Comparar vizinhanças das *features* (NCC)
  - Aplicar o algoritmo húngaro
  - técnica “ingênua” (não considera distâncias)

Como resolver?



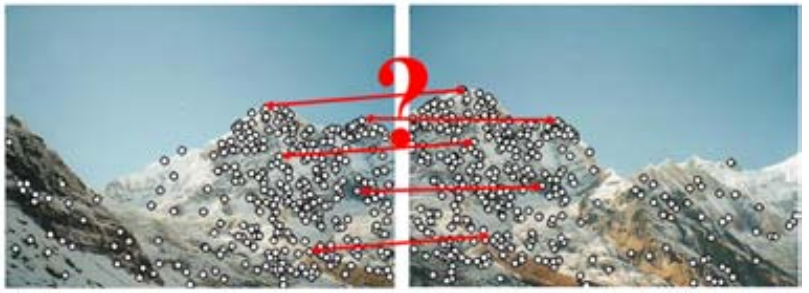
- #1
  - Comparar vizinhanças das *features* (NCC)
  - Aplicar o algoritmo húngaro
  - técnica “ingênua” (não considera distâncias)
- #2

Como resolver?

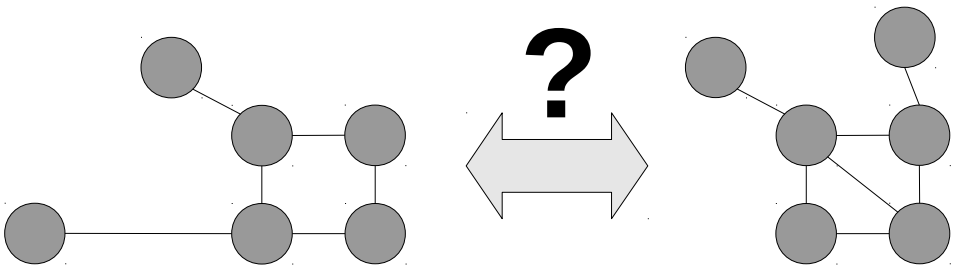


- #1
  - Comparar vizinhanças das *features* (NCC)
  - Aplicar o algoritmo húngaro
  - técnica “ingênua” (não considera distâncias)
- #2
  - Considerar também distâncias

Como resolver?



- #1
  - Comparar vizinhanças das *features* (NCC)
  - Aplicar o algoritmo húngaro
  - técnica “ingênua” (não considera distâncias)
- #2
  - Considerar também distâncias
  - *graph matching*!



Algoritmo de Pedarsani et al.

(P. Pedarsani, D. R. Figueiredo e M. Grossglauser, “A bayesian method for matching two similar graphs without seeds,” IEEE 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2013.)

Algoritmo de Pedarsani et al.

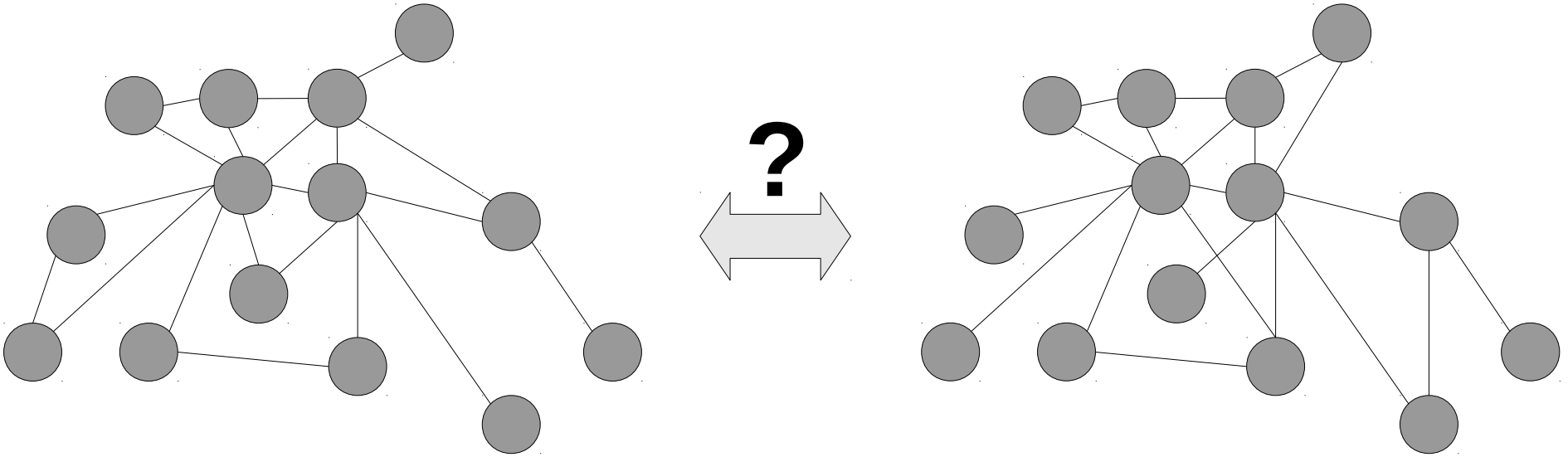
- *Framework* probabilístico (bayesiano) para *graph matching*

Algoritmo de Pedarsani et al.

- *Framework* probabilístico (bayesiano) para *graph matching*
- Iterativo

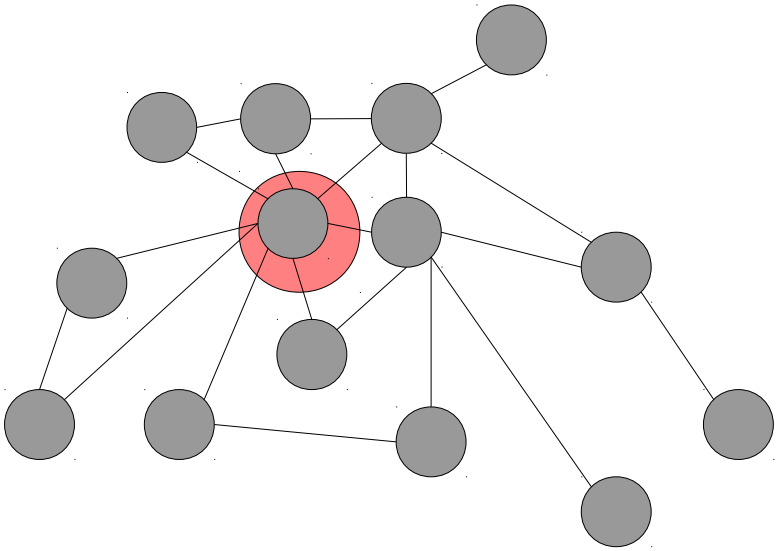
Algoritmo de Pedarsani et al.

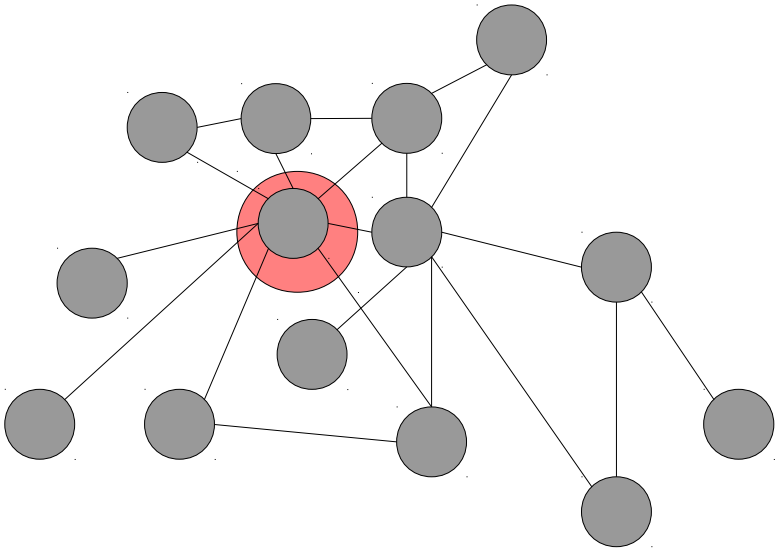
- *Framework* probabilístico (bayesiano) para *graph matching*
- Iterativo



Algoritmo de Pedarsani et al.

- *Framework* probabilístico (bayesiano) para *graph matching*
- Iterativo

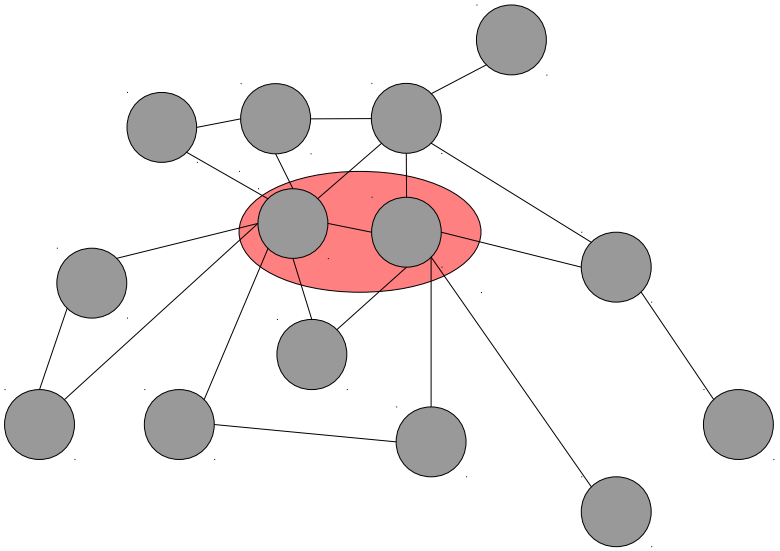


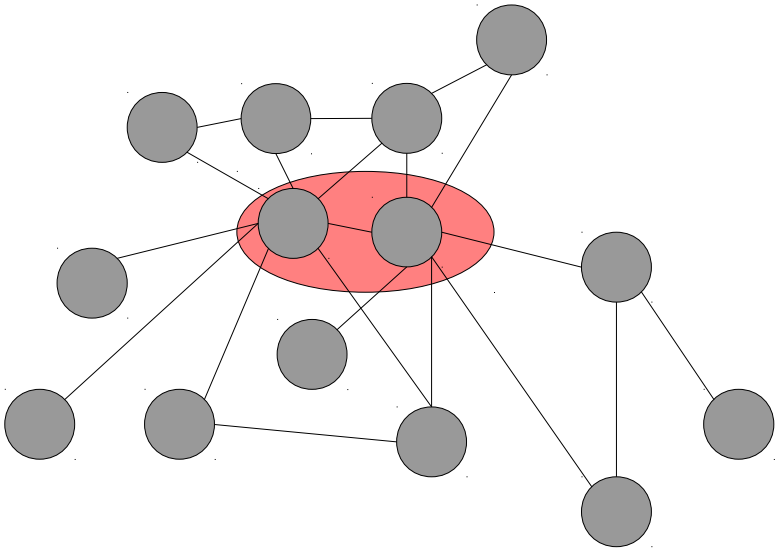


Começa pelos vértices de maior grau

Algoritmo de Pedarsani et al.

- *Framework* probabilístico (bayesiano) para *graph matching*
- Iterativo

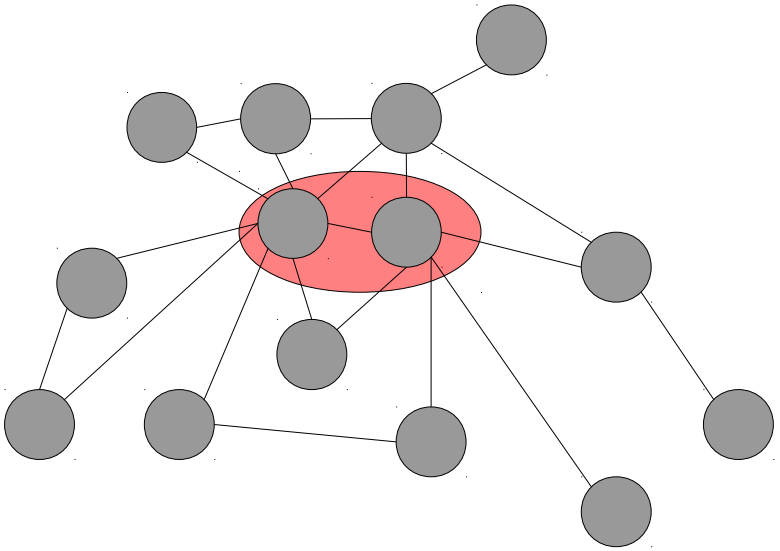


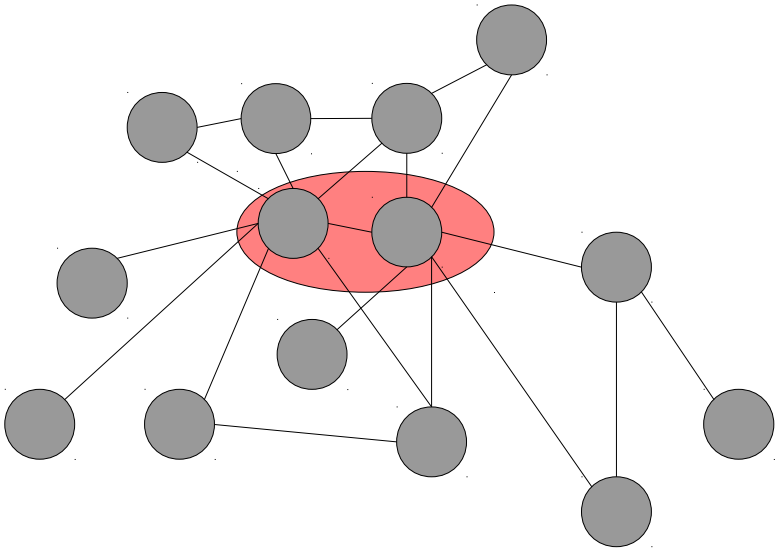


iterativo

Algoritmo de Pedarsani et al.

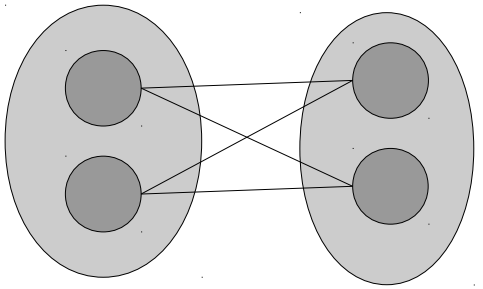
- *Framework* probabilístico (bayesiano) para *graph matching*
- Iterativo





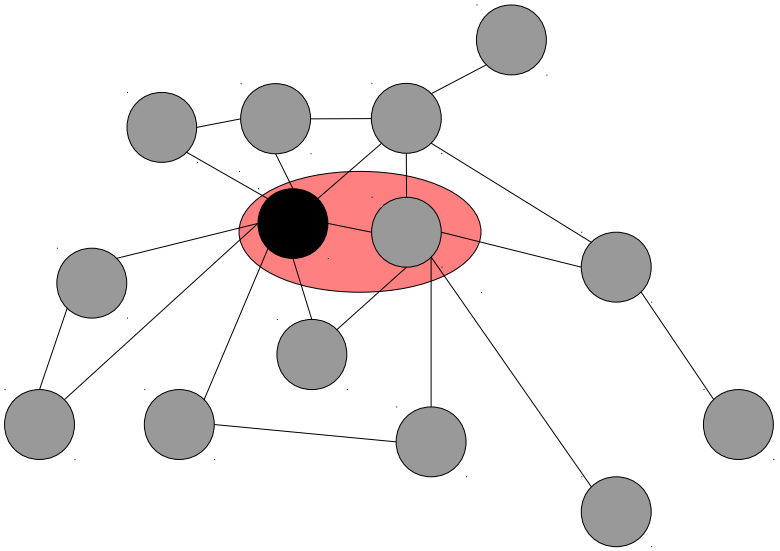
probabilidade de u=v?
(dado o grau)

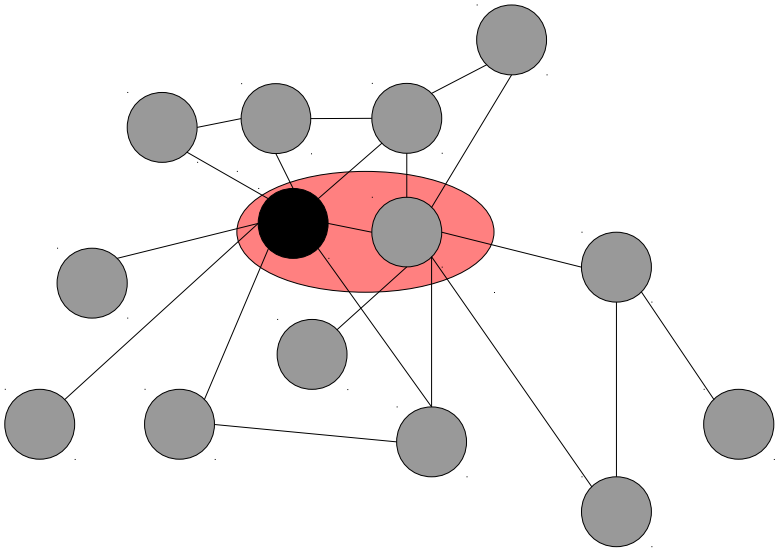
algoritmo húngaro



Algoritmo de Pedarsani et al.

- *Framework* probabilístico (bayesiano) para *graph matching*
- Iterativo

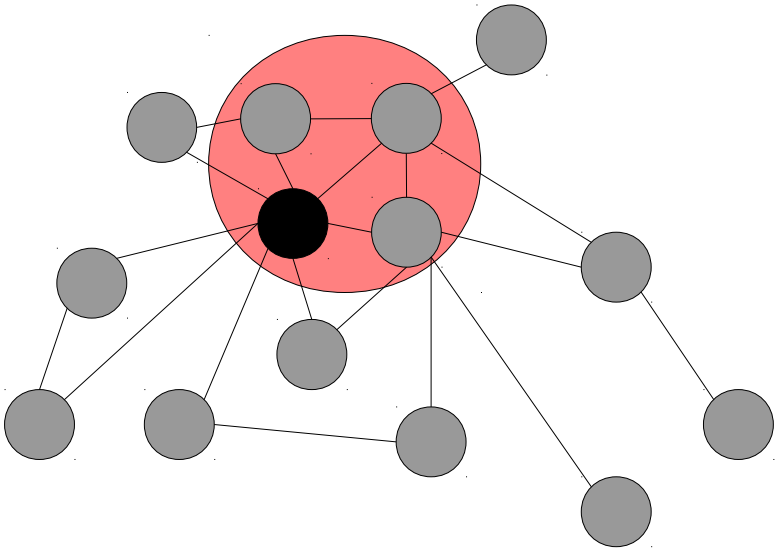


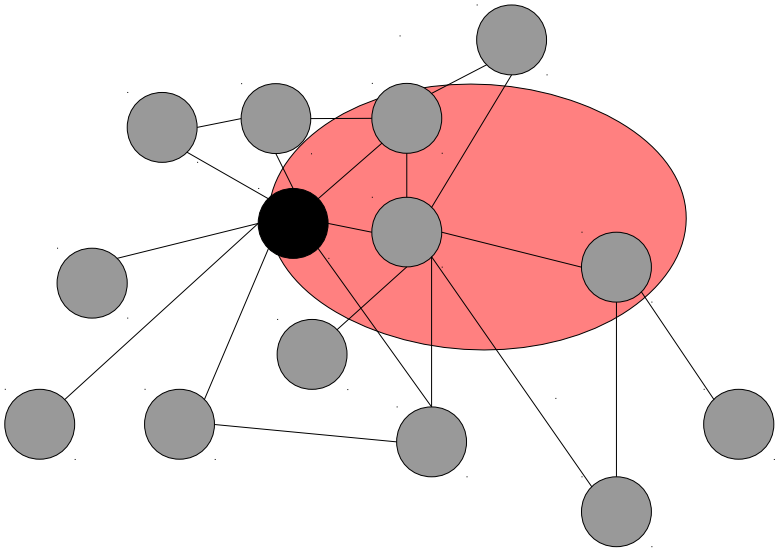


escolha de vértices-âncora
(os melhor mapeados)

Algoritmo de Pedarsani et al.

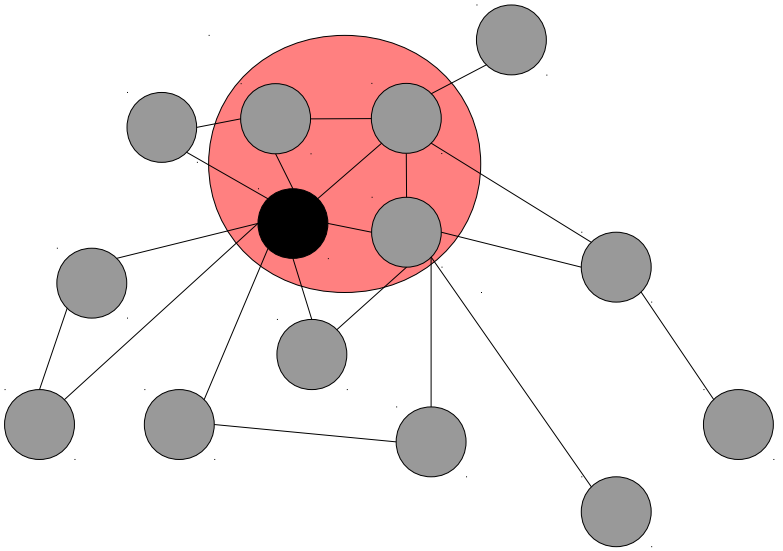
- *Framework* probabilístico (bayesiano) para *graph matching*
- Iterativo

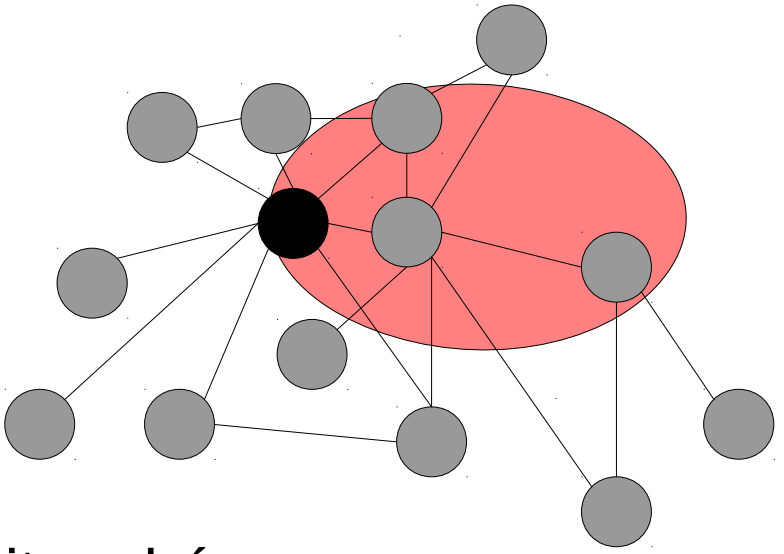




Algoritmo de Pedarsani et al.

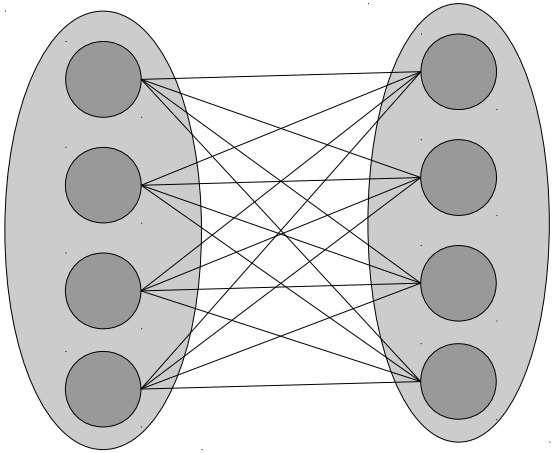
- *Framework* probabilístico (bayesiano) para *graph matching*
- Iterativo





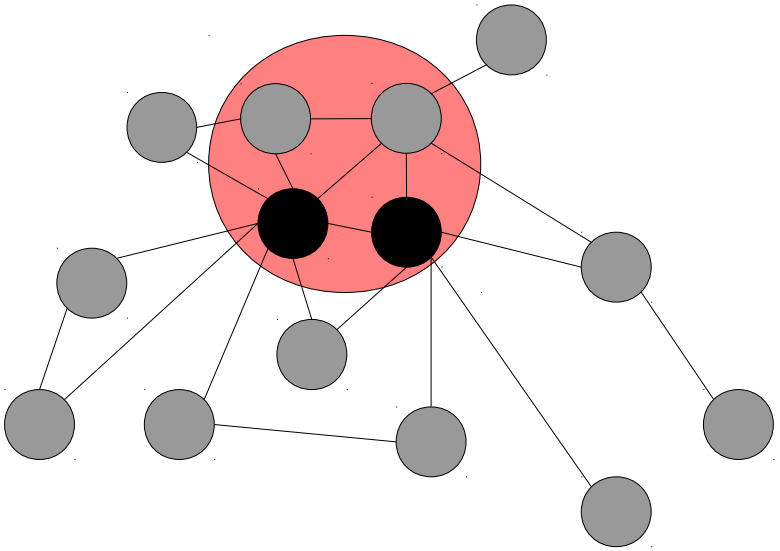
probabilidade de u=v?
(dado o grau e a
distância aos âncoras)

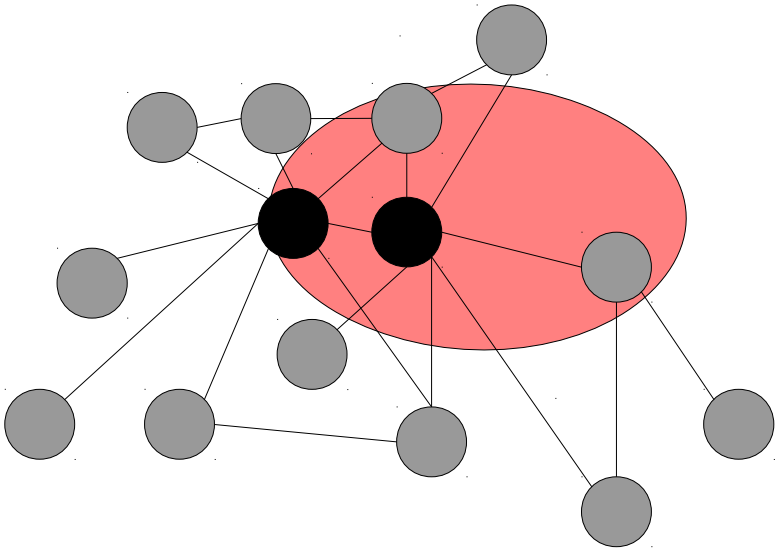
algoritmo húngaro



Algoritmo de Pedarsani et al.

- *Framework* probabilístico (bayesiano) para *graph matching*
- Iterativo

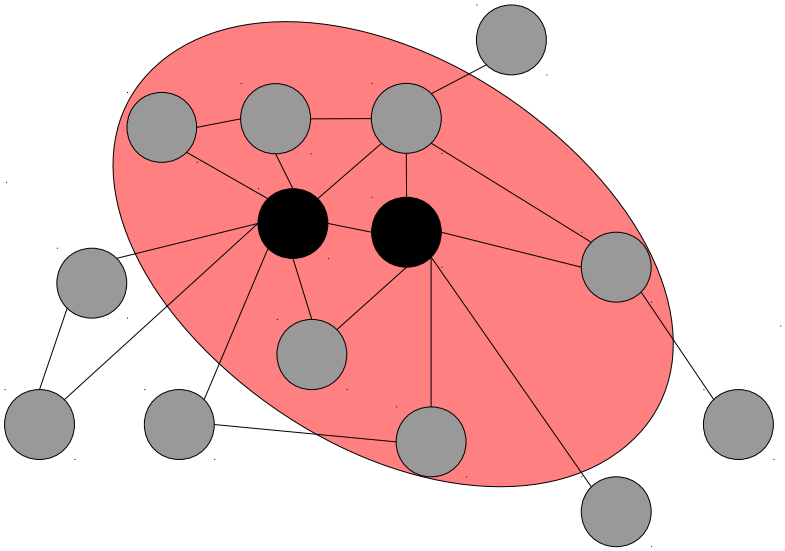


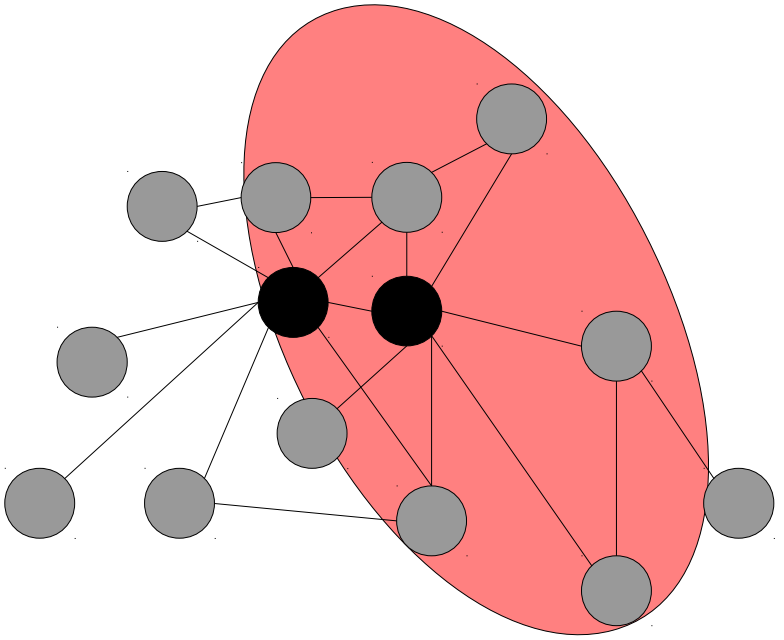


reescolher vértices-âncora

Algoritmo de Pedarsani et al.

- *Framework* probabilístico (bayesiano) para *graph matching*
- Iterativo





Nossa Adaptação

Nossa Adaptação

- Grafo completo
  - Arestas têm pesos

Nossa Adaptação

- Grafo completo
  - Arestas têm pesos
- Informação de cor (vizinhanças/NCC) no lugar do grau
  - normal multivariável para a cor
  - log-normal para a distância

Nossa Adaptação

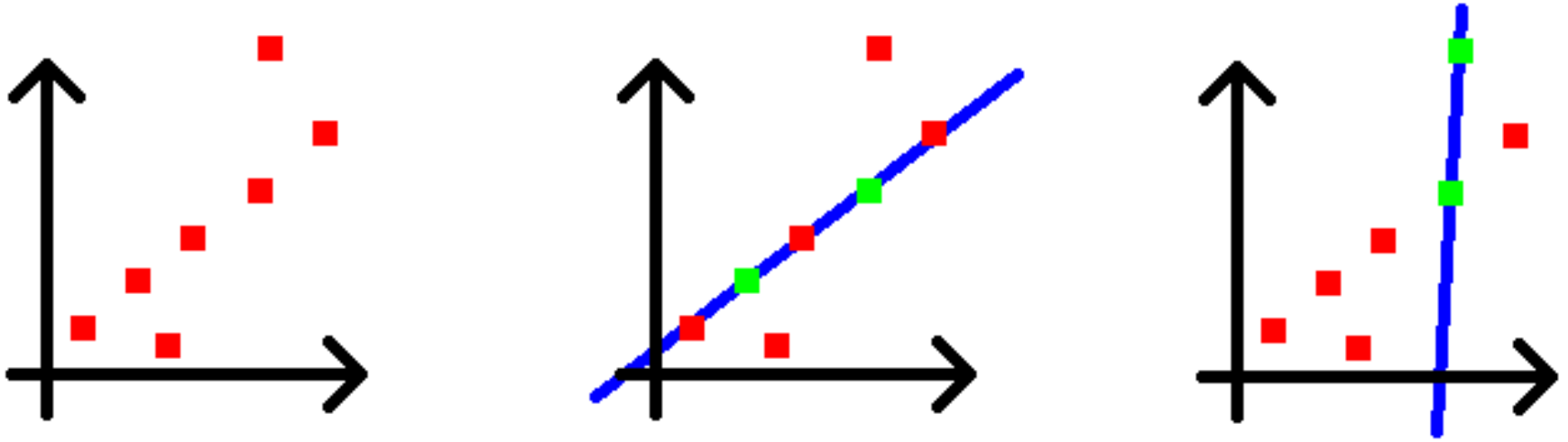
- Grafo completo
  - Arestas têm pesos
- Informação de cor (vizinhanças/NCC) no lugar do grau
  - normal multivariável para a cor
  - log-normal para a distância
- Apenas 5 iterações (grafo euclidiano)
  - Sempre entram todos os vértices

Nossa Adaptação

- Grafo completo
  - Arestas têm pesos
- Informação de cor (vizinhanças/NCC) no lugar do grau
  - normal multivariável para a cor
  - log-normal para a distância
- Apenas 5 iterações (grafo euclidiano)
  - Sempre entram todos os vértices
- Outras alterações na forma de calcular
  - Variáveis são contínuas

Resultados

- Avaliação: RANSAC
  - Algoritmo estocástico
  - Resolve o problema de mínimos quadrados da aplicação (panografia ou reconstrução estéreo) separando *outliers*
  - Métrica de avaliação: Número médio de *inliers* encontrados



Resultados: Panografia

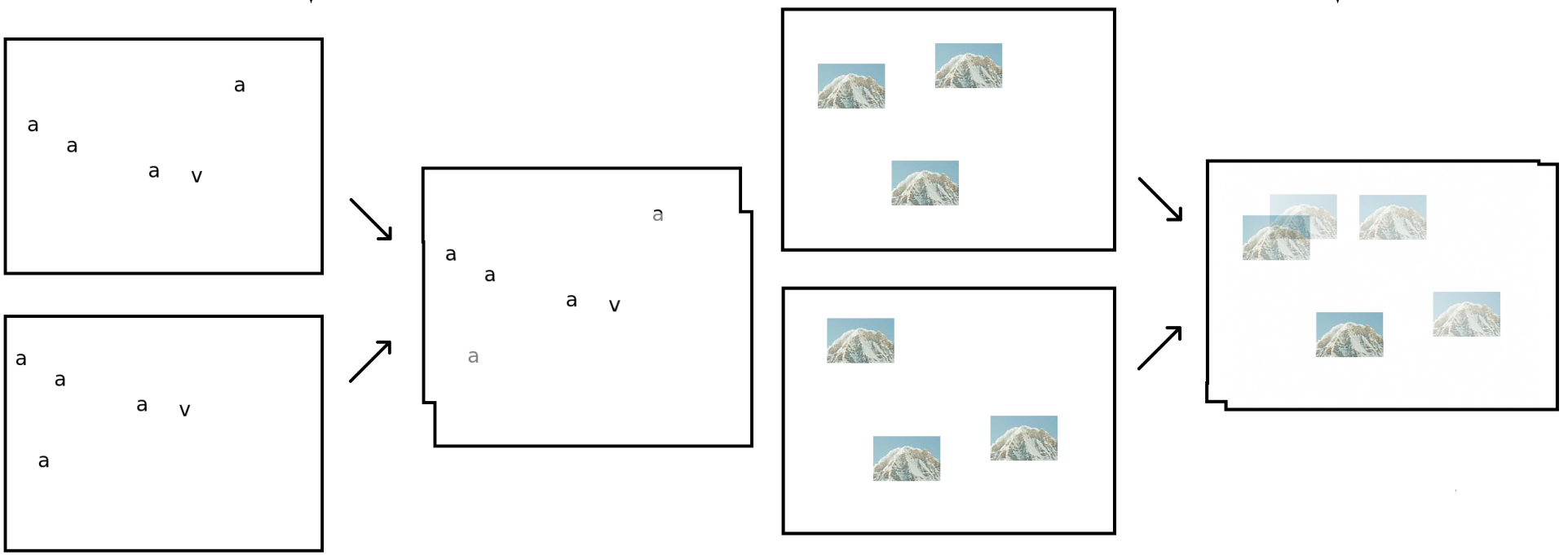
nossa técnica ($\mu \pm \sigma$ *inliers*)	técnica ingênua ($\mu \pm \sigma$ *inliers*)	$\min\{n_1, n_2\}$	melhoria relativa $(\frac{\mu_{\text{nossa}} - \mu_{\text{ingênua}}}{\min\{n_1, n_2\}})$
65.7 ± 3.1	63.8 ± 1.0	112	01.7%
274.5 ± 3.5	266.9 ± 3.6	483	01.2%
140.7 ± 4.0	117.4 ± 3.5	490	04.8%
270.4 ± 3.3	250.8 ± 2.3	425	04.6%
61.1 ± 6.5	55.6 ± 6.2	477	01.6%

(Imagens de:

- R. Szeliski, Computer Vision: Algorithms and Applications, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.
- R. I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

Casos Sintéticos

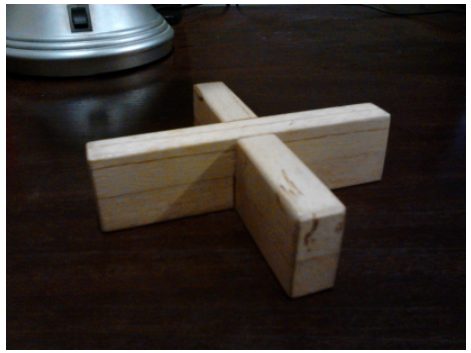
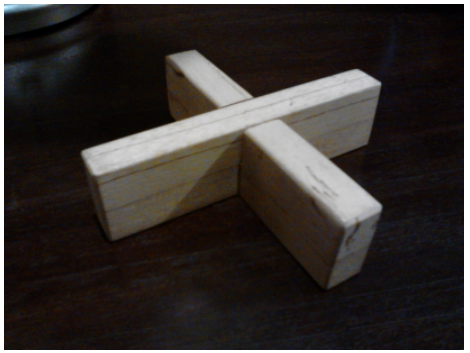
nossa técnica ($\mu \pm \sigma$ *inliers*)	técnica ingênua ($\mu \pm \sigma$ *inliers*)	$\min\{n_1, n_2\}$	melhoria relativa $(\frac{\mu_{\text{nossa}} - \mu_{\text{ingênua}}}{\min\{n_1, n_2\}})$
20.0 ± 0.0	10.0 ± 0.0	25	40.0%
71.0 ± 0.0	71.0 ± 0.0	213	00.0%



(Imagem adaptada de R. Szeliski, Computer Vision: Algorithms and Applications, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.)

Resultados: Reconstrução Estéreo

nossa técnica ($\mu \pm \sigma$ *inliers*)	técnica ingênua ($\mu \pm \sigma$ *inliers*)	$\min\{n_1, n_2\}$	melhoria relativa $(\frac{\mu_{\text{nossa}} - \mu_{\text{ingênua}}}{\min\{n_1, n_2\}})$
36.5 ± 0.7	30.3 ± 3.1	46	13.5%
60.2 ± 3.0	45.0 ± 2.3	76	20.0%

Conclusão

- Panografia
  - Não houve melhorias significativas
  - Ambas as técnicas resolvem o problema bem
- Reconstrução estéreo
  - Distâncias provêem muita informação
  - Ainda não elimina as dificuldades inerentes ao problema

- Relatório e apresentação disponíveis em:
  - http://www.lcg.ufrj.br/Members/gustavopfeiffer/graphncc/

Obrigado pela atenção!

Dúvidas?
Comentários?