Application of graph matching in Computer Vision

Master Seminar

Ekaterina Tikhoncheva

University of Heidelberg
Faculty of Mathematics and Computer Science
Computer Vision group
at
Heidelberg Collaboratory for Image Processing

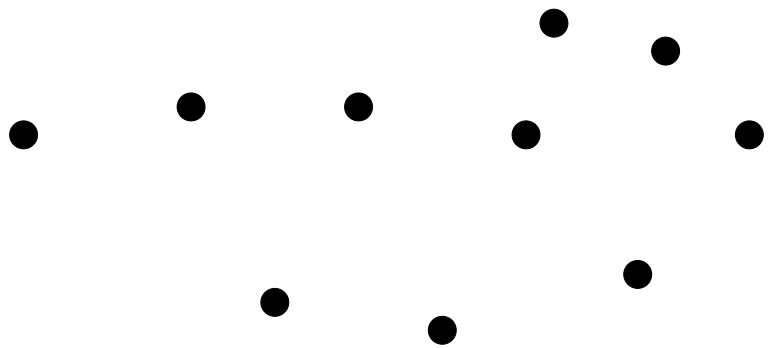
November 2015

Agenda

Attributed undirected graph I

Attributed undirected graph $G = (V,$

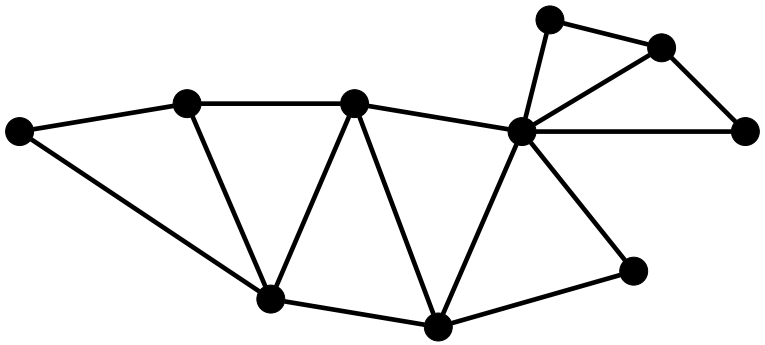
- set of nodes $V = \{v_i\}_{i=1}^{n}$



Attributed undirected graph II

Attributed undirected graph $G = (V, E,$

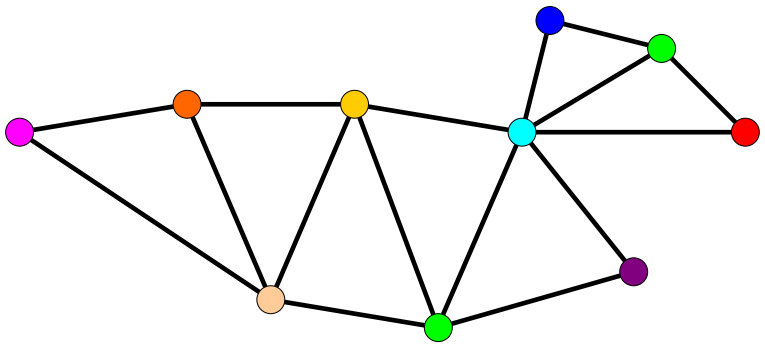
- set of nodes $V = \{v_i\}_{i=1}^{n}$
- set of edges $E \subseteq \{\{u, v\} | u, v \in V\}$



Attributed undirected graph

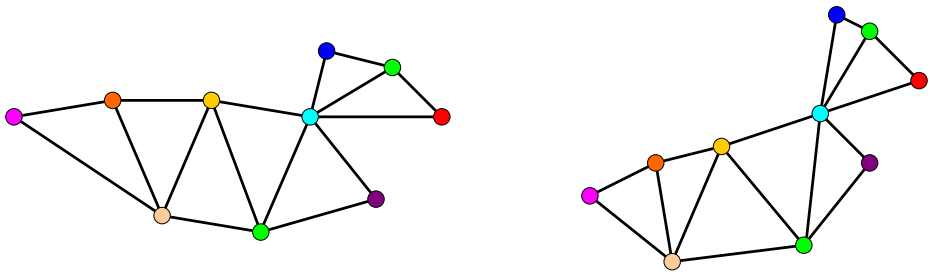
Attributed undirected graph $G = (V, E, D)$

- set of nodes $V = \{v_i\}_{i=1}^{n}$
- set of edges $E \subseteq \{\{u, v\} | u, v \in V\}$
- node attributes $D = \{d_i\}_{i=1}^{n}$

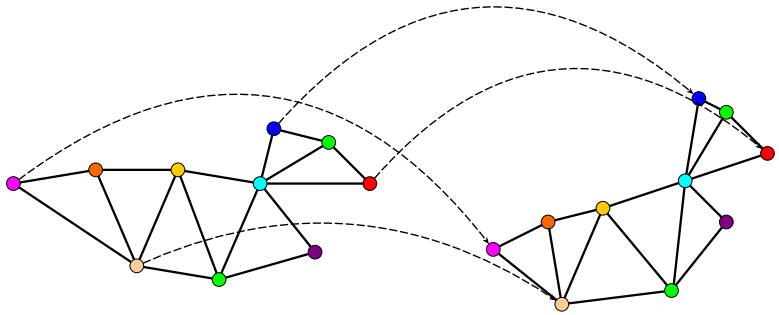


Graph matching

Let us consider two undirected attributed graphs $G^I = (V^I, E^I, D^I)$ and $G^J = (V^J, E^J, D^J)$:



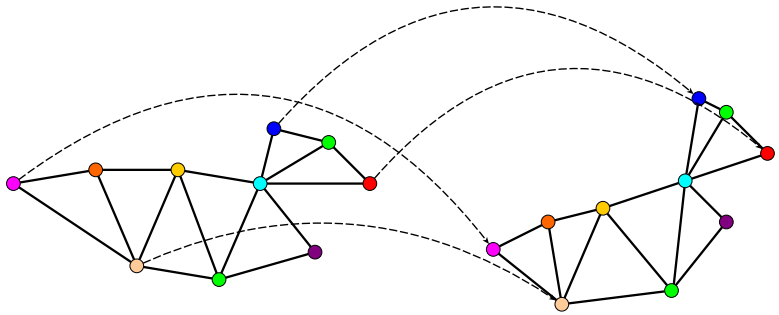
Graph matching



A matching function between G^I and G^J is a mapping

$$m : V^I \rightarrow V^J$$

Graph matching

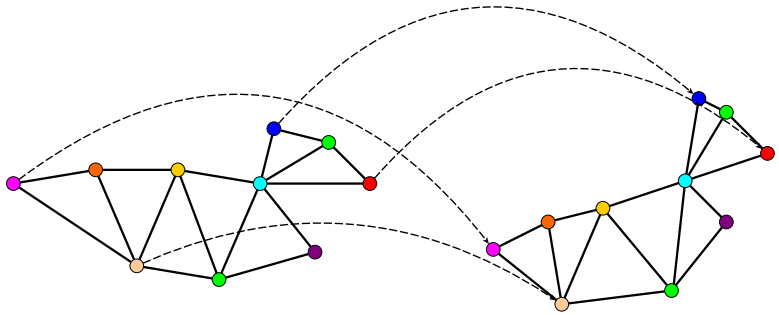


A matching function between G^I and G^J is a mapping

$$m : V^I \rightarrow V^J$$

not unique!

Graph matching



A matching function between G^I and G^J is a mapping

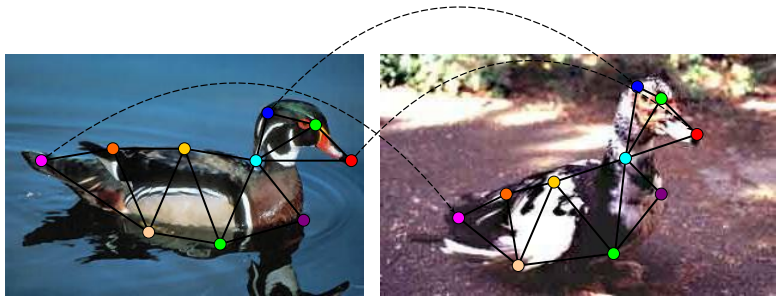
$$m : V^I \rightarrow V^J$$

Define a function $S(G^I, G^J, m)$ to measure the quality of matching m that fulfills some conditions

$\Rightarrow$ Graph matching problem between G^I and G^J

$$m = \underset{\hat{m}}{\operatorname{argmax}}\, S(G^I, G^J, \hat{m})$$

Graph matching in computer vision

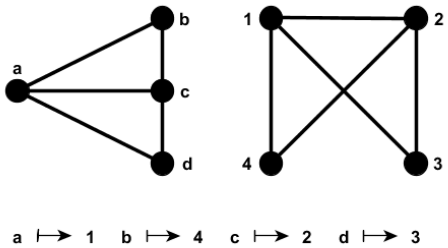


- image matching
- shape matching
- object detection
- object tracking
- ...

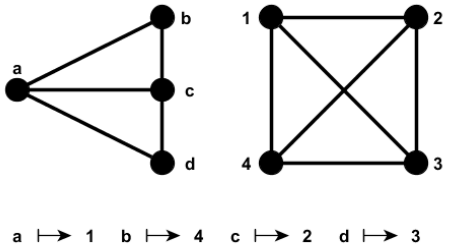
Exact graph matching

Edge preserving mapping m:
$\{v_i, v_{i'}\} \in E^I \Rightarrow \{m(v), m(v_{i'})\} \in E^J$ for all $v_i, v_{i'} \in V^I$.

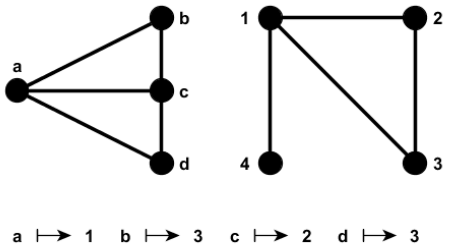
- mapping m is bijective $\rightarrow$ graph isomorphism
- mapping m is injective $\rightarrow$ graph monomorphism
- mapping m is total $\rightarrow$ graph homomorphism



(a) Graph isomorphism



(b) Graph monomorphism

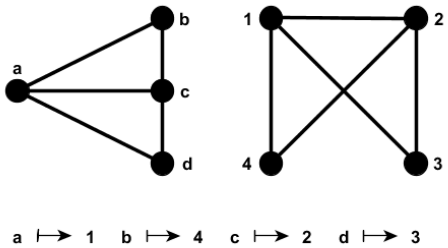


(c) Graph homomorphism

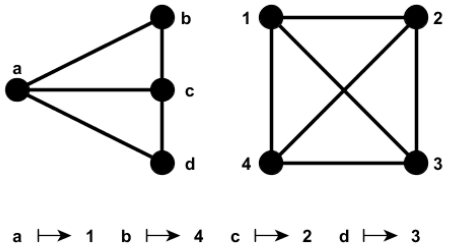
Exact graph matching

Edge preserving mapping m:
$\{v_i, v_{i'}\} \in E^I \Rightarrow \{m(v), m(v_{i'})\} \in E^J$ for all $v_i, v_{i'} \in V^I$.

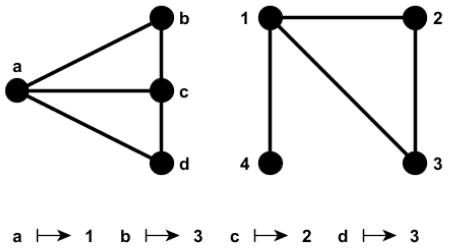
- mapping m is bijective $\rightarrow$ graph isomorphism
- mapping m is injective $\rightarrow$ graph monomorphism
- mapping m is total $\rightarrow$ graph homomorphism



(a) Graph isomorphism



(b) Graph monomorphism



(c) Graph homomorphism

not unique!

Given two attributed graphs $\bar{G}^P = (\bar{V}^P, \bar{E}^P, \bar{A}^P)$ and $\bar{G}^Q = (\bar{V}^Q, \bar{E}^Q, \bar{A}^Q)$ with n^P and n^Q nodes respectively.
A result of graph matching is a subset of possible correspondences between those graphs, which can be represented in form of assignment matrix $X \in \{0,1\}^{n^P \times n^Q}$:

$$X_{ia} = \begin{cases} 1 & \text{node } v_i \in \bar{V}^P \text{matches } v_a \in \bar{V}^Q \\ 0 & \text{otherwise} \end{cases}$$

General formulation:

$$x^* = \arg\max S(x)$$

$$s.t. \begin{cases} x \in \{0,1\}^{n^P n^Q} \\ \sum_{i=1}^{n^P} x_{ia} \leq 1 \\ \sum_{a=1}^{n^Q} x_{ia} \leq 1 \end{cases}$$

The objective function $S(x)$ measures the similarity between the

Drei exemplarische Typen von Suchanfragen

- spezielle:
  „Does Netscape support the JDK 1.1 - code-signing API?“[**?**]

- breit angelegte:
  „Find information about the Java programming language“[**?**]
  - Suche nach ähnlichen Seiten
    „Find pages similar to *java.sun.com*“[**?**]

Drei exemplarische Typen von Suchanfragen

- spezielle: Problem der Knappheit
  „Does Netscape support the JDK 1.1 - code-signing API?“[**?**]

- breit angelegte:
  „Find information about the Java programming language“[**?**]
  - Suche nach ähnlichen Seiten
    „Find pages similar to *java.sun.com*“[**?**]

Drei exemplarische Typen von Suchanfragen

- spezielle: Problem der Knappheit
  „Does Netscape support the JDK 1.1 - code-signing API?“[**?**]

- breit angelegte: Problem der Vielfältigkeit
  „Find information about the Java programming language“[**?**]

  - Suche nach ähnlichen Seiten
    „Find pages similar to *java.sun.com*“[**?**]

Ranking

- Man möchte die angesehensten Seiten (Authorities) aus der Menge aller zu der Anfrage relevanten Seiten finden
- Mögliche Hindernisse:
  - die höchst relevanten Seiten werden nicht unbedingt durch ein textbasiertes Ranking vorgezogen
  - es kann sein, dass die relevanten Seiten die Wörter aus der Suchanfrage gar nicht enthalten

Authorities und Hubs I

Annahme

Die Relevanz zwischen zwei Seiten wurde vom Ersteller des Links zwischen diesen Seiten geprüft

Stimmt im Allgemeinen nicht (z.B. Navigationslinks, Werbung)

Aber unter dieser Annahme reicht es, nur die Linkstruktur des WWW zu betrachten, um die Autorität einer Seite im Bezug auf eine andere zu bestimmen

Authorities und Hubs II

Authorities (Autoritätsseiten)

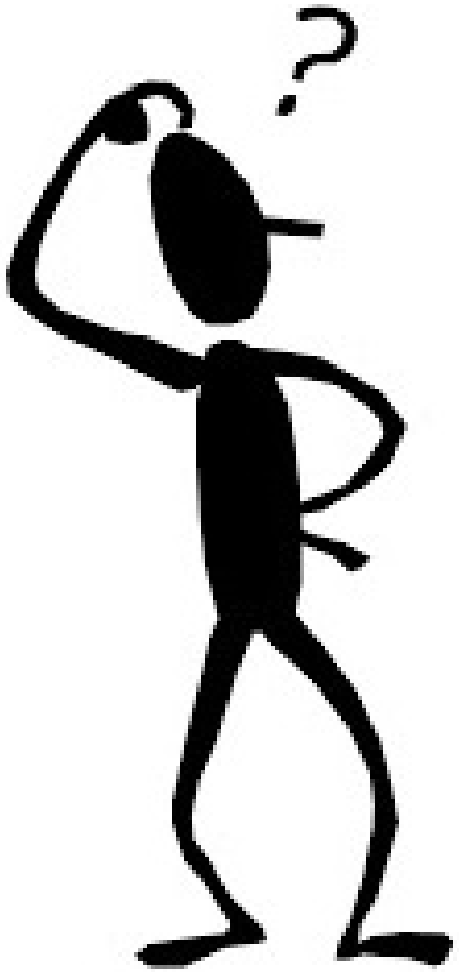
Relevante Seiten, auf die viele weitere relevante Seiten zeigen

Hubs

Seiten, die auf viele Authorities zeigen

The end

Thank you for your attention!



References I

[1] Minsu Cho, Karteek Alahari, and Jean Ponce. Learning Graphs to Match. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.