

Progressive Graph Matching: Making a Move of Graphs via Probabilistic Voting

Minsu Cho
Kyoung Mu Lee

Proc. Computer Vision and Pattern Recognition (CVPR) 2012

Graph Matching

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:

$$\begin{aligned} x^* &= \arg \max S(x) \\ \text{s.t. } &\begin{cases} x \in \{0, 1\}^{n^P n^Q} \\ \sum_{i=1}^{n^P} x_{ia} \leq 1 \\ \sum_{a=1}^{n^Q} x_{ia} \leq 1 \end{cases} \end{aligned}$$

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

Integer Quadratic Programming

Consider two types of similarity: node and edge similarity. Those can be combined in an affinity matrix W , where non-diagonal elements $W_{ia,jb}$ represent edge similarity of the edges $e_{ij} \in \bar{E}^P$ and $e_{ab} \in \bar{E}^Q$ and diagonal elements $W_{ia,ia}$ represent similarity between nodes $v_i \in \bar{V}^P$ and $v_a \in \bar{V}^Q$.

That leads to the integer quadratic optimization problem:

$$x^* = \arg \max x^T W x$$

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

Issues

- Quadratic assignment problem is NP -complete
- Computation of W for a large graph is almost infeasible
- Reduction of the complexity often leads to worse matching results

Progressive Graph Matching

Refer to the initial graphs \bar{G}^P and \bar{G}^Q as **maximal graphs**. For each maximal graph a sub-graph induced by a reduced set of nodes is called **active graph** (G^P and G^Q respectively).

Idea: match maximal graphs by iteratively matching their active graphs ($t = 0, 1, \dots$)

- **Graph Matching**

match current active graphs $G_t^P = (V_t^P, E_t^P, A_t^P)$ and $G_t^Q = (V_t^Q, E_t^Q, A_t^Q)$, $G_t^P \subset \bar{G}^P, G_t^Q \subset \bar{G}^Q$.

- **Graph Progression**

update active graphs to improve matching score on the next matching step

Graph Matching

To reduce the complexity of graph matching consider a given **set of candidate matches** $C_t \subset V^P \times V^Q$.

The corresponding active graphs are defined by the nodes appeared in C_t .

This method reduces the size of active graphs and makes a affinity matrix more sparse.

Graph Progression I

Let $M_t = \{m_1, \dots, m_{|M_t|}\} \subset C_t$ with $m_i = (v_{p_i}^P, v_{q_i}^Q)$ be a **result of graph matching** of two active graphs and s_t it's **score**.

Consider conditional joint probability $p(V^P, V^Q | M_t)$:

$$\begin{aligned} p(V^P, V^Q | M_t) &= \sum_{m_i \in M_t} p(V^P, V^Q, M = m_i | M_t) \\ &= \sum_{m_i \in M_t} p(V^Q | V^P, M = m_i, M_t) \\ &\quad p(V^P | M = m_i, M_t) p(M = m_i | M_t) \end{aligned}$$

Graph Progression II

$p(v_i^P, v_a^Q | M_t)$ conditional probability of match (v_i^P, v_a^Q) between two maximal graphs.

Base on the probability distribution $p(V^P, V^Q | M_t)$ new candidate set C_{t+1} consists of N_c best matches.

Given new candidate set C_{t+1} we obtain new active graphs G_{t+1}^P and G_{t+1}^Q .

The condition $M_t \subset C_{t+1}$ ensures that $s_{t+1} \geq s_t$, if M_t is an optimal matching.

Algorithm

Algorithmus 1 progressiveGraphMatching($\bar{G}^P, \bar{G}^Q, N_c$) [1]

```

1:  $C_0 = \text{Find\_Initial\_Candidates}(\bar{G}^P, \bar{G}^Q, N_c)$ 
2:  $t = 0, s_0 = 0$ 
3: while score increases do
4:    $(M_t, s_t) = \text{Graph\_Matching}(C_t)$ 
5:    $p(v_p^P, v_q^Q | M_t) = 0 \quad \forall v_p^P \in \bar{V}^P, v_q^Q \in \bar{V}^Q$ 
6:   for each  $m_i = (v_p^P, v_q^Q) \in M_t$  do
7:      $N_A = \{v^P \in \bar{V}^P | p(v^P | m_i, M_t) > \epsilon\}$ 
8:     for each  $v_j^P \in N_A$  do
9:        $N_B = \{v^Q \in \bar{V}^Q | p(v^Q | v_j^P, m_i, M_t) > \epsilon\}$ 
10:      for each  $v_b^Q \in N_B$  do
11:        
$$p(v_p^P, v_q^Q | M_t) = p(v_p^P, v_q^Q | M_t) +$$


$$p(v_b^Q | v_j^P, m_i, M_t)p(v_j^P | m_i, M_t)p(m_i | M_t)$$

12:      end for
13:    end for
14:  end for
15:   $C_{t+1} = N_c$  best matches based on  $p(V^P, V^Q | M_t)$ , which contains  $M_t$ 
16:   $t = t + 1$ 
17: end while
18: return  $M_t$ 

```

Matching of images I

Given two images.

- Detected features of the images represent nodes of the maximal graphs (and their geometric relations edges)
- Node similarity = similarity of feature descriptors of two graphs
- Edge similarity = Symmetric Transfer Error (STE) (see [3])

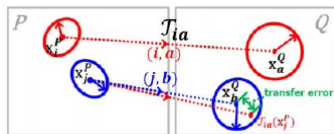
Derive affine homography transformation $\tau_{ia}(\cdot)$ between two features

The transfer error of (j, b) with respect to (i, a)

$$d_{jb|ia} = \|x_b^Q - \tau_{ia}(x_j^P)\|$$

Edge similarity

$$W_{ia,jb} = \max(0, \alpha - \frac{d_{jb|ia} + d_{bj|ai} + d_{ia|jb} + d_{ai|jb}}{4})$$

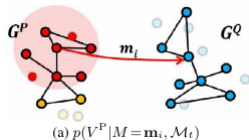


Matching of images II

$$p(\mathbf{V}^P, \mathbf{V}^Q | \mathbf{M}_t) = \sum_{\mathbf{m}_i \in \mathbf{M}_t} p(\mathbf{V}^Q | \mathbf{V}^P, \mathbf{M} = \mathbf{m}_i, \mathbf{M}_t) p(\mathbf{V}^P | \mathbf{M} = \mathbf{m}_i, \mathbf{M}_t) p(\mathbf{M} = \mathbf{m}_i | \mathbf{M}_t)$$

- $p(\mathbf{M} = \mathbf{m}_i | \mathbf{M}_t) = \text{score}(m_i) / \sum_i \text{score}(m_i)$
- Assuming $m_i = (v_{p_i}^P, v_{q_i}^Q)$ and use function $kNN(\cdot, k)$ to find k -nearest neighbours of a node

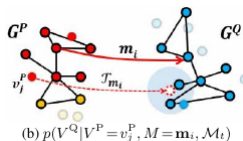
$$p(\mathbf{V}^P = v_j^P | \mathbf{M} = \mathbf{m}_i, \mathbf{M}_t) = \begin{cases} 1/k_1 & , \text{ if } v_j^P \in kNN(v_{p_i}^P, k_1) \\ 0 & , \text{ otherwise} \end{cases}$$



- Let $NN(\cdot)$ define a nearest neighbour of a node

$$p(\mathbf{V}^Q = v_b^Q | \mathbf{V}^P = v_j^P, \mathbf{M} = \mathbf{m}_i, \mathbf{M}_t) =$$

$$\begin{cases} 1 & , \text{ if } v_b^Q = NN(\tau_{m_i}(p_j^P)) \text{ and } (v_j^P, v_b^Q) \in M_t \\ \exp(-d_{jb|m_i}) / Z & , \text{ if } v_b^Q \in kNN(\tau_{m_i}(p_j^P), k_2) \text{ and } (v_j^P, NN(\tau_{m_i}(p_j^P))) \notin M_t \\ 0 & , \text{ otherwise} \end{cases}$$

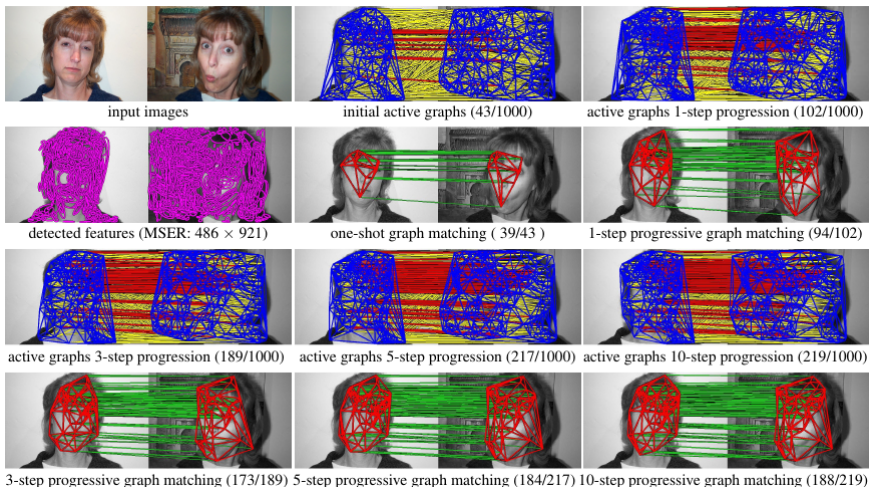


$$Z = \sum_{v_b^Q \in kNN(\tau_{m_i}(p_j^P), k_2)} \exp(-d_{jb|m_i}) \text{ is a normalization constant}$$

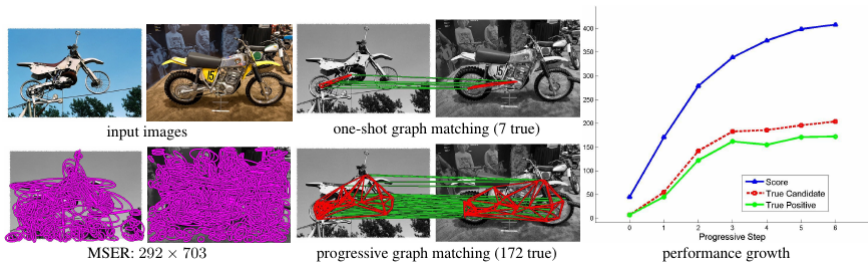
Evaluation

- Parameters of the affinity matrix W : $\alpha = 50$, $k_1 = 25$, $k_2 = 5$
- Used Descriptors: **MSER** [4] and Harris-affine and Hessian-affine features [5]
- used graph matching methods:
 - Graduate Assignment Algorithm (**SM**) [8]
 - Spectral Matching with Affine Constraint (**SMAC**) [9]
 - Probabilistic Graph Matching (**PM**) [7]
 - Reweighted Random Walk Method (**RRWM**) [3]
 - Integer Projected Fixed Point Method (**IPFP**) [6]
- Datasets: **VOC 2010**, **Caltech-101**, **MSRC**, **ETHZ** toys dataset

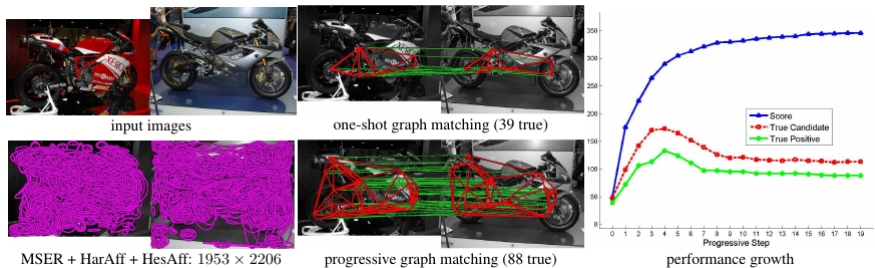
Evaluation: Progressive Graph Matching ($N_c = 1000$)



Evaluation: Progressive Graph Matching ($N_c = 3000$)



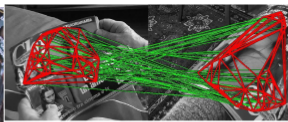
Evaluation: Progressive Graph Matching ($N_c = 3000$)



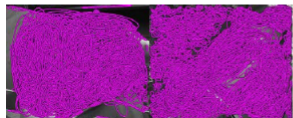
Evaluation: Progressive Graph Matching ($N_c = 3000$)



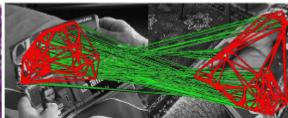
input images



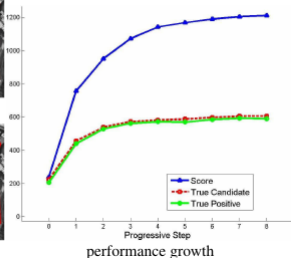
one-shot graph matching (206 true)



MSER + HarAff + HesAff: 7315 × 6497

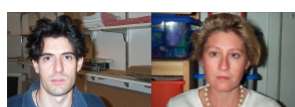


progressive graph matching (591 true)

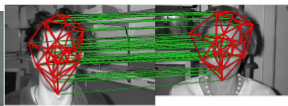


Evaluation: Robustness to initial active graph ($N_c = 3000$)

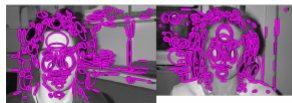
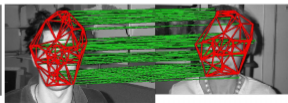
|



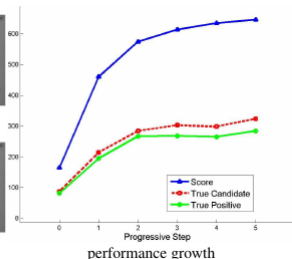
input images



one-shot graph matching (82 true)

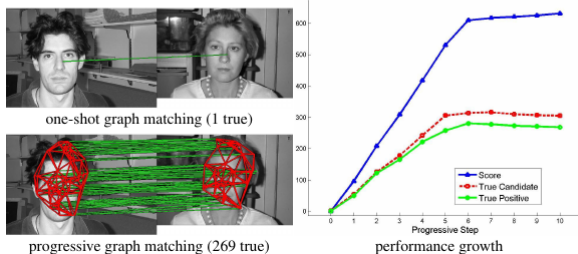
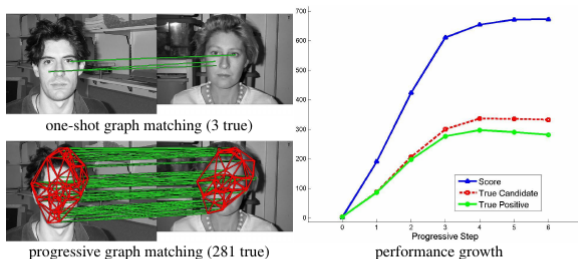
MSER: 262×266 

progressive graph matching (284 true)



Evaluation: Robustness to initial active graph ($N_c = 3000$)

II



Evaluation: Different matching algorithms

	Graph Matching Module				
One-Shot	SM	SMAC	PM	RRWM	IPFP
Accuracy (%)	62.6	57.6	63.7	73.6	71.9
Progressive	SM	SMAC	PM	RRWM	IPFP
Accuracy (%)	68.2	63.6	66.7	81.2	78.2
Prog. vs. One-Shot	SM	SMAC	PM	RRWM	IPFP
Score Growth (%)	+65.0	+38.7	+92.1	+65.7	+63.8
Inlier Growth (%)	+59.6	+17.0	+85.1	+65.6	+69.7

Figure : Dataset of 30 image pairs. Accuracy boost 3% ~ 8%

Conclusions

- graph progression significantly increases one-step graph matching
- algorithm is suitable for large graphs
- (?) convergence of the algorithm
- (?) selection of the candidate matches
- (?) number of the candidate matches
- (?) running time
- (?) geometric graphs

The End

Thank you for your attention!

References I

- [1] Minsu Cho and Kyoung Mu Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [2] Minsu Cho and Kyoung Mu Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. supplementary material. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [3] Minsu Cho, Kyoung Mu Lee, and Jungmin Lee. Reweighted random walks for graph matching. *European Conference on Computer Vision (ECCV)*, 2010.
- [4] Matas. J, Chum. O, Urban. M, and Pajdla. T. Robust wide baseline stereo from maximally stable extremal regions. *BMVC*, 2002.

References II

- [5] Mikolajczyk. K and Schmid. C. Scale and affine invariant interest point detectors. *IJCV*, 2004.
- [6] Leordeanu. M and Herbert. M. An integer projected fixed point method for graph matching and map interface. *NIPS*, 2009.
- [7] Zass. R and Shashua. A. Probabilistic graph and hypergraph matching. *CVRP*, 2008.
- [8] Gold. S and Rangarajan. A. A graduate assignment algorithm for graph matching. *TPAMI*, 1996.
- [9] Cour. T, Srinivasan. P, and Shi. J. Balanced graph matching. *NIPS*, 2007.