A Neural Network Model for Solving the Feature Correspondence Problem

Ala Aboudib(✉), Vincent Gripon, and Gilles Coppin

Lab-STICC UMR CNRS 6285, Télécom Bretagne, Technopôle Brest-Iroise CS 83818, 29238 Brest Cedex 3, France
{ala.aboudib,vincent.gripon,gilles.coppin}@telecom-bretagne.eu

Abstract. Finding correspondences between image features is a fundamental question in computer vision. Many models in literature have proposed to view this as a graph matching problem whose solution can be approximated using optimization principles. In this paper, we propose a different treatment of this problem from a neural network perspective. We present a new model for matching features inspired by the architecture of a recently introduced neural network. We show that by using popular neural network principles like max-pooling, k-winners-take-all and iterative processing, we obtain a better accuracy at matching features in cluttered environments. The proposed solution is accompanied by an experimental evaluation and is compared to state-of-the-art models.

Keywords: Artificial neural networks · Feature matching · Graph matching · Iterative processing · Max-pooling

1 Introduction

Establishing correspondences between two sets of visual features is a fundamental problem in computer vision. Solving this problem is essential to many visual processing tasks. This includes feature tracking [10], object discovery [11], structure from motion [17], stereo matching [20], image classification [8] and many other applications. An early class of algorithms consisted in matching features based on the similarity of their descriptor vectors. Such similarity can be obtained using simple metrics such as euclidean or hamming distances for example [19]. While such methods are still widely popular, their ability to find correct matches becomes obsolete in more complex situations such as in the presence of multiple instances of the object whose features are to be matched, or in the case of matching two different objects that belong to the same class, or in the presence of clutter.

Early attempts to address this problem consisted in taking the geometric consistency between features into account. This includes methods such as

This work was supported by the European Research Council under the European Union's Seventh Framework Program (FP7/2007-2013) / ERC grant agreement n° 290901.

A.E.P. Villa et al. (Eds.): ICANN 2016, Part II, LNCS 9887, pp. 439–446, 2016.
DOI: 10.1007/978-3-319-44781-0_52

RANSAC [6] and ICP [2]. These methods assume that the deformations undergone by an object are rigid, i.e., they are governed by some form of a parametric transformation (e.g. planar affine or epipolar). However, these methods are not adapted to non-rigid transformations which are very common in natural images.

To address non-rigid transformations, a class of models emerged in the last two decades that applied graph matching techniques (GM) to the correspondence problem [4,12,22]. These methods formulate the matching problem as an optimization procedure of a well-defined objective function. This function takes individual feature similarity into account, as well as other geometric constraints such as pairwise feature affinity measures [12], or even higher order measures [21]. Little effort, however, was devoted to seeking a potential neural network model for solving the graph matching problem. We think that this is an interesting question from an algorithmic point of view, as well as for researchers interested in Marr's third level of analysis that seeks possible neural mechanisms for implementing vision algorithms [14]. While the present paper addresses this level of analysis, we do not pretend providing a real bio-mimetic solution. We hope that our approach be a step forward for vision research seeking biological inspiration.

The main contribution of our work is to introduce an artificial neural network (ANN) model for addressing the feature correspondence problem. This model is adapted from the sparse clustered neural network designed by Gripon and Berrou in [9], which is a generalization of the Palm-Wilshaw neural network [18]. The main *advantage* of the proposed matching algorithm is its better robustness against clutter compared to state-of-the-art. However, when no clutter is present, which is argued to be a less interesting case, the proposed algorithm only gives a comparable or a less matching accuracy. Another advantage is that our approach implements a cooperative algorithm, meaning that each neuron needs only to know about the activity of a few neighboring neurons, which allows for the algorithm to be run in parallel.

The rest of this paper is organized in four sections. In Sect. 2, a brief overview of state-of-the-art algorithms proposed for solving the correspondence problem is presented. The architecture of the neural network along with the algorithm we propose are presented in Sect. 3. The performance of the proposed model is evaluated in Sect. 4 and compared to some other algorithms. Section 5 is a conclusion.

2 Related Work

As mentioned earlier, feature correspondence can be viewed as a graph matching (GM) problem, which is traditionally formulated as a quadratic assignment problem (QAP) known to be NP-hard. Its solution is usually approximated by optimizing an objective function with relaxed constraints [12,21,22]. However, there were some attempts to approximate this optimization procedure by applying an iterative process without defining an explicit objective to optimize [3–5,7]. These attempts date back to as early as Marr's cooperative algorithm for solving the stereo matching problem [14]. It provided an insight on how iterative

algorithms can be used to tackle difficult vision problems using only local information.

Max-pooling matching (MPM) introduced by Cho *et al.* in [3] is one recent example of such iterative algorithms. It applies max-pooling to preserve important information while discarding irrelevant details making it more robust in the presence of outliers. Some other methods that use a similar iterative approach include re-weighted random walk matching (RRWM) [4], balanced graph matching [5] and more [7].

Our approach is similar to MPM in that it applies max-pooling to discard irrelevant details. Unlike MPM, pooling is not only applied among features of one image but also in the second one. Another major difference is that the final discretization step is replaced by a non-linear activation function applied at each iteration and a winner-take-all (WTA) applied at the end, which is akin to local inhibition observed among neural assemblies [16].

In the following section, we describe our ANN model and specify the details of the matching algorithm it implements. We use a similar terminology as in [3] in order to highlight the similarities and differences between the two algorithms, and to show where the proposed model is positioned relative to the state-of-the-art.

3 The Proposed Model

Feature correspondence is formulated as the problem of matching a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to a sub-graph of $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, where $\mathcal{E}$, $\mathcal{E}'$ are the sets of graph edges, and $\mathcal{V}$, $\mathcal{V}'$ are sets of nodes. Graph $\mathcal{G}$ represents an object with its features as nodes in $\mathcal{V}$. The same holds for $\mathcal{G}'$ except that it might be representing a scene including other objects than the one we are seeking to match.

We define an assignment matrix $\mathbf{X} \in \{0,1\}^{n \times n'}$, where n and n' denote the number of nodes in $\mathcal{V}$ and $\mathcal{V}'$, respectively. We only set $\mathbf{X}_{ia} = 1$ when a feature $v_i \in \mathcal{V}$ matches another $v_a \in \mathcal{V}'$. We shall use a column-wise vectorized version of $\mathbf{X}$ that we denote $\mathbf{x} \in \{0,1\}^{nn'}$.

We also define a unary similarity function $s_V(v_i, v'_a)$ to describe similarities among descriptor vectors of features in $\mathcal{V}$ and $\mathcal{V}'$, and a pairwise similarity function $s_E(e_{ij}, e'_{ab})$ with $e_{ij} \in \mathcal{E}$ and $e'_{ab} \in \mathcal{E}'$ as in [3,12]. We use these functions to define a unary affinity vector $\mathbf{y}_{ia} = s_V(v_i, v'_a)$ with $\mathbf{y} \in \mathbb{R}^{nn'}$, and a pairwise similarity matrix $\mathbf{A} \in \mathbb{R}^{nn' \times nn'}$ as:

$$\mathbf{A}_{ia;jb} = \begin{cases} s_E(e_{ij}, e'_{ab}) & \text{if } i \neq j \text{ and } a \neq b. \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Notice from (1) that $\mathbf{A}$ is a symmetric matrix, and that elements of its main diagonal are always set to zero. The main diagonal does not hold the unary similarity values as in most traditional algorithms [3,12]. These values are stored in the vector $\mathbf{y}$.

The neural network we propose for solving the correspondence problem is constructed on the graph captured by the affinity matrix $\mathbf{A}$, as in the example

of Fig. 1. The architecture of this network is adapted from the sparse clustered network (SCN) [9] which was proposed as a generalization of Palm-Wilshaw networks [18].

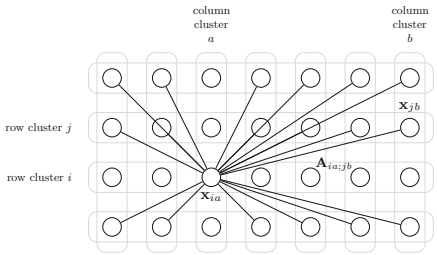

Fig. 1. The architecture of the proposed neural network.

The network grid structure depicted in Fig. 1 corresponds to the 2D configuration of the assignment matrix $\mathbf{X}$. As in SCNs, we impose a grouping configuration on the network neurons in the form of clusters; neurons of the same row are grouped into one cluster, and the same holds for neurons of the same column. Thus, each neuron belongs to two clusters as shown in Fig. 1. Within each cluster, a WTA activation constraint is imposed; only one neuron per cluster can be active at the end of the network activity with a binary activation level (0 or 1) captured by $\mathbf{X}$ as in [9]. However, during the network activity, and before $\mathbf{X}$ reaches its final state, this constraint is relaxed into a k-winners-take-all (kWTA) constraint, and we allow $\mathbf{X}$ to temporarily contain real values. The connections between neurons are captured by the pairwise affinity matrix $\mathbf{A}$, and as we notice from (1), no connections exists between neurons of the same cluster ($\mathbf{A}_{ia} = 0$) as in SCNs.

The WTA and kWTA constraints we impose within clusters are meant to encourage the one-to-one matching constraint between features in $\mathcal{V}$ and $\mathcal{V}'$. From a biological perspective, this is akin to the local competition among neural assemblies enforced by short inhibitory synaptic connections [16].

The network activity starts by assigning to each neuron its unary affinity value ($\mathbf{X}_{ia} \leftarrow \mathbf{y}_{ia}$). Then, within each row cluster, every neuron receives the max-pooled propagated activity of all other neurons to which it connects as in [1,3]:

$$\mathbf{x}_{ia}^{t+1} \leftarrow \mathbf{x}_{ia}^{t} \sum_{j \in \mathcal{V}} \max_{b \in \mathcal{V}'} \mathbf{x}_{jb}^{t} \mathbf{A}_{ia;jb}, \tag{2}$$

where the superscript t denotes the current iteration. The activity values within this cluster are then normalized to their maximum, and a kWTA operation is applied:

$$\mathbf{x}_{ia}^{t+1} \longleftarrow \mathbf{x}_{ia}^{t} h(\mathbf{x}_{ia}^{t} - \tau) : a \in \mathcal{V}', \tag{3}$$

where $h(.)$ is the unit step function and $\tau \in [0, 1]$ is the kWTA activation threshold. Another iteration is then applied, this time on column clusters. We alternate

between row-wise and column-wise iterations until the convergence of $\mathbf{X}$ or until a fixed maximum number of iterations it attained. Notice that for row clusters, max-pooling and kWTA are applied row-wise, while they are applied column-wise for column clusters.

Finally, an activation threshold is applied, where only neurons with a maximal activation value ($\mathbf{x}_{ia} = 1$) are kept active while others are deactivated ($\mathbf{x}_{ia} \leftarrow 0$). A WTA operation is then applied within every row and column cluster; if more than one neuron is active in a given cluster, they are all deactivated and no winner is declared. This is equivalent to imposing an 'at most' one-to-one matching constraint from $\mathcal{V}$ to $\mathcal{V}'$. The complete matching process we propose is described in Algorithm (1).

Algorithm 1. Proposed matching algorithm.

input : Pairwise affinity matrix $\mathbf{A}$, Unary similarity vector $\mathbf{y}$
output: Assignment vector $\mathbf{x}$
$\mathbf{x} \longleftarrow \mathbf{y}$
repeat
  foreach $i \in \mathcal{V}$ **do**
    foreach $a \in \mathcal{V}'$ **do**
      $\mathbf{x}_{ia}^{t+1} \leftarrow \mathbf{x}_{ia}^{t} \sum_{j \in \mathcal{V}} \max_{b \in \mathcal{V}'} \mathbf{x}_{jb}^{t} \mathbf{A}_{ia;jb}$
    $\mathbf{x}_{ia}^{t+1} \longleftarrow \frac{\mathbf{x}_{ia}^{t+1}}{\max_{a \in \mathcal{V}'} \mathbf{x}_{ia}^{t+1}} : a \in \mathcal{V}'$
    $\mathbf{x}_{ia}^{t+1} \longleftarrow \mathbf{x}_{ia}^{t+1} h(\mathbf{x}_{ia}^{t+1} - \tau) : a \in \mathcal{V}'$
  $\mathbf{x}_{ia}^{t} \longleftarrow \mathbf{x}_{ia}^{t+1}$
  foreach $a \in \mathcal{V}'$ **do**
    foreach $i \in \mathcal{V}$ **do**
      $\mathbf{x}_{ia}^{t+1} \leftarrow \mathbf{x}_{ia}^{t} \sum_{b \in \mathcal{V}'} \max_{j \in \mathcal{V}} \mathbf{x}_{jb}^{t} \mathbf{A}_{ia;jb}$
    $\mathbf{x}_{ia}^{t+1} \longleftarrow \frac{\mathbf{x}_{ia}^{t+1}}{\max_{i \in \mathcal{V}} \mathbf{x}_{ia}^{t+1}} : i \in \mathcal{V}$
    $\mathbf{x}_{ia}^{t+1} \longleftarrow \mathbf{x}_{ia}^{t+1} h(\mathbf{x}_{ia}^{t+1} - \tau) : i \in \mathcal{V}$
until $\mathbf{x}$ converges OR last iteration attained
$\mathbf{x}_{ia} \leftarrow \delta_1^{\mathbf{x}_{ia}} : i \in \mathcal{V}$ and $a \in \mathcal{V}'$ — $\#\delta$ is the Kronecker delta.
WTA: Zero all rows and columns in $\mathbf{X}$ with more than one non-zero element.

To sum up, the network behavior consists in each neuron adding up its input signals, which are the max-pooled weighted activities of other neurons. Then, a non-linear activation function is applied to this neuron, taking into account the activity level of other members of its cluster. This is akin to the classic accumulate-and-fire neuron model of McCulloch-Pitts [15].

4 Experimental Evaluation

In order to evaluate our model, we compare its matching accuracy against a number of state-of-the-art models on a synthetic benchmark. Synthetic datasets

are typically used for assessing performance of matching algorithms because they allow better control of test parameters.

The synthetic dataset is built as follows. Two graphs $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$ are constructed, where $\mathcal{V}, \mathcal{V}' \subset \mathbb{R}^2$ and $\mathcal{E}, \mathcal{E}' \subset \mathbb{R}$. Then, n_{in} points that we call inliers are generated from a uniform random distribution on $[0, 1]^2$, and are added to $\mathcal{V}$. These inliers are also copied to $\mathcal{V}'$ after the addition of a Gaussian noise $\mathcal{N}(0, \sigma^2)$. After that, we add n_{out} outliers, generated from the same uniform random distribution $[0, 1]^2$, to each of $\mathcal{V}$ and $\mathcal{V}'$. Pairwise similarities are computed as follows:

$$s_E(e_{ij}, e'_{ab}) = \exp(-\big|\|v_i - v_j\| - \|v'_a - v'_b\|\big|). \tag{4}$$

Unary similarities are always set to one $s_V(v_i, v'_a) = 1$ so that points are matched using only their pairwise geometric information. The kWTA activation threshold is set to $\tau = 0.98$ in all of our experiments. We noticed that in most cases, convergence is attained after 5 to 10 iterations. However, as in [3], a theoretical guarantee for convergence is not yet proved but is worth exploring.

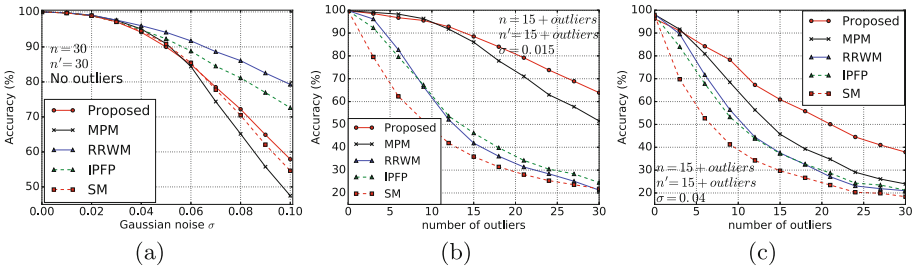

Fig. 2. Experimental comparison of the proposed model's accuracy with several state-of-the-art models on a synthetic dataset. In (a), no outliers are present, and the standard deviation σ of the Gaussian noise is varied. In (b) and (c), the number of outliers is varied for a fixed value of σ. The same number of outliers shown on the horizontal axis is added to both sets $\mathcal{V}$ and $\mathcal{V}'$.

We compare our model to MPM [3], RRWM [4], IPFP [13] and SM [12]. We are only interested in finding matches between inliers in $\mathcal{V}$ and $\mathcal{V}'$, outliers are used to represent clutter. We use the models' mean accuracy as a convenient performance criterion. Accuracy is measured as the ratio of the number of correct matches to the total number of inliers. Comparisons results are shown in Fig. 2. We notice that in the presence of outliers, our model's accuracy becomes significantly better than other models' as the number of outliers increases. This is an interesting property since clutter and deformation are ubiquitous in natural images. This robustness is due to the max-pooling and the kWTA operations that we apply to reduce the effect of false matches on the final result. Notice also that accuracy of our model is still higher than MPM's and SM's when no outliers are present, but lower than that of RRWM and IPFP. However, as stated

in [3], comparing accuracies in the absence of outliers is a less realistic situation as outliers are almost always present in natural images, and robustness against clutter is essential in such situations.

5 Conclusion and Future Work

In this paper, we proposed a new approach for treating the feature correspondence problem using artificial neural network. We compared our model to state-of-the-art algorithms, and showed that it enjoys a higher robustness to outliers thanks to the application of max-pooling and kWTA operations, and to alternating rows and columns during iterations. This robustness to outliers is an essential property for matching objects in cluttered scenes. Further development of our model will include searching for a better way of choosing final matches than zeroing rows and columns of the assignment matrix containing more than one winner. We think that it is a simple but a brutal procedure that might be excluding some good matches. We shall also test the performance of the model in the context of natural images, which would give a more precise evaluation of the advantage of using this neural network model for solving the correspondence problem.

References

1. Aboudib, A., Gripon, V., Jiang, X.: A study of retrieval algorithms of sparse messages in networks of neural cliques. In: COGNITIVE 2014: The 6th International Conference on Advanced Cognitive Technologies and Applications, pp. 140–146. Venise, Italy, May 2014. https://hal.archives-ouvertes.fr/hal-01058303
2. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Robotics-DL Tentative, pp. 586–606. International Society for Optics and Photonics (1992)
3. Cho, M., Sun, J., Duchenne, O., Ponce, J.: Finding matches in a haystack: a max-pooling strategy for graph matching in the presence of outliers. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2091–2098 (2014)
4. Cho, M., Lee, J., Lee, K.M.: Reweighted random walks for graph matching. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 492–505. Springer, Heidelberg (2010)
5. Cour, T., Srinivasan, P., Shi, J.: Balanced graph matching. In: Schölkopf, B., Platt, J.C., Hoffman, T. (eds.) Advances in Neural Information Processing Systems 19, pp. 313–320. MIT Press (2007). http://papers.nips.cc/paper/2960-balanced-graph-matching.pdf
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
7. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. IEEE Trans. Pattern Anal. Mach. Intell. **18**(4), 377–388 (1996). http://dx.doi.org/10.1109/34.491619
8. Grauman, K., Darrell, T.: The pyramid match kernel: discriminative classification with sets of image features. In: Tenth IEEE International Conference on Computer Vision (ICCV 2005), vol. 2, pp. 1458–1465, October 2005

9. Gripon, V., Berrou, C.: Sparse neural networks with large learning diversity. IEEE Trans. Neural Netw. **22**(7), 1087–1096 (2011)
10. Jiang, H., Yu, S.X., Martin, D.R.: Linear scale and rotation invariant matching. IEEE Trans. Pattern Anal. Mach. Intell. **33**(7), 1339–1355 (2011)
11. Leordeanu, M., Collins, R.: Unsupervised learning of object features from video sequences. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 1142–1149, June 2005
12. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: Tenth IEEE International Conference on Computer Vision (ICCV 2005), vol. 2, pp. 1482–1489, October 2005
13. Leordeanu, M., Hebert, M., Sukthankar, R.: An integer projected fixed point method for graph matching and map inference. In: Advances in Neural Information Processing Systems, pp. 1114–1122 (2009)
14. Marr, D.: Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. Henry Holt and Co., Inc., New York (1982)
15. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. **5**(4), 115–133 (1943)
16. Mountcastle, V.B.: The columnar organization of the neocortex. Brain **120**(4), 701–722 (1997)
17. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: Segmenting, modeling, and matching video clips containing multiple moving objects. IEEE Trans. Pattern Anal. Mach. Intell. **29**(3), 477–491 (2007)
18. Schwenker, F., Sommer, F., Palm, G.: Iterative retrieval of sparsely coded associative memory patterns. Neural Netw. **9**(3), 445–455 (1996). http://www.sciencedirect.com/science/article/pii/0893608095001123
19. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer Science & Business Media, London (2010)
20. Tuytelaars, T., Gool, L.V.: Wide baseline stereo matching based on local, affinely invariant regions. In: British Machine Vision Conference (BMVC 2000), pp. 412–425 (2000)
21. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), pp. 1–8, June 2008
22. Zhou, F., la Torre, F.D.: Factorized graph matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012), pp. 127–134, June 2012