Mirror Match: Reliable Feature Point Matching Without Geometric Constraints

Jonas Toft Arnfred,*† Stefan Winkler,* Sabine Süsstrunk†

* Advanced Digital Sciences Center (ADSC), University of Illinois at Urbana-Champaign (UIUC), Singapore

† École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract—**Many algorithms have been proposed to solve the problem of matching feature points in two or more images using geometric assumptions to increase the robustness of the matching. However, these assumptions do not always hold; in particular, few methods address the problem of reliable matching in cases where it is unknown whether the images have any corresponding areas or objects in the first place.**

We propose two algorithms for matching feature points *without* the use of geometric constraints. The first relies on the fact that any match between two images should be better than all possible matches within a single image. The second algorithm extends this idea by using community structure in the similarity graph of feature points to find reliable correspondences. To evaluate the algorithms experimentally, we introduce a simple method to generate a large amount of test cases based on a set of image pairs with viewpoint changes. Our results show that the proposed algorithm is generally superior to traditional approaches in finding correct correspondences.

I. Introduction

When finding correspondence pairs between two or more images, feature points in one image cannot always be uniquely matched with feature points in another image. This is partly due to the fact that the discriminatory power of modern feature descriptors such as *SIFT* [1] or *SURF* [2] is traded off with invariance to light and perspective changes. Additionally, corners detected by feature detectors such as *SIFT* and *Harris* [3] are only approximations of the ideal feature point in an image which is uniquely identifiable. These deficiencies often lead to suboptimal results when matching feature points by finding the nearest neighbors in terms of descriptor similarity. Therefore, Lowe introduced *SIFT* together with a matching method tailored to finding unique matches [1]. The method, *NN-Ratio*, assesses the uniqueness of a given match by looking at the two nearest neighbors of each feature point and calculating the matching score as the ratio of similarities; only matches with a ratio above a given threshold are retained. Lowe's *NN-Ratio* has been shown to outperform nearest neighbor approaches that use only the similarity of the descriptors as a threshold [4].

More recent methods for matching feature points consider the geometric configuration of feature points and match them based on assumptions about the geometric relationship between images. These assumptions are then used to select a subset of matches according to different constraints, e.g. angular constraints [5], epipolar constraints [6], [7], or pairwise constraints [8], [9].

Alternatively we can pick out different regions in each image and pair regions according to the number of correspondences between them. This allows for the filtering of all correspondences that do not match points within paired regions. Examples include *Isodata* [10], which uses clustering to find regions, or [11], which designate areas based on maximally stable extremal regions (MSER).

Solutions that rely on geometric assumptions about the images fall short in many practical applications. For example, images with partial or no overlap often result in feature points that match to arbitrary points in the other image; this introduces noise that makes it hard to pick the right subset of correct matches. Similarly, adjacent objects in one image that are separated in another will void any global geometric assumptions of the geometric relationship between two images. Furthermore, methods using geometric constraints all require an initial set of correspondences. If this set of correspondences can be narrowed down to the most probable correct matches, any geometric matching algorithm will benefit in terms of speed and accuracy in cases where geometric assumptions do hold true. For these reasons reliable non-geometric matching methods are essential.

The two methods we propose in this paper are designed to be free of assumptions about image geometry. They improve on Lowe's *NN-Ratio* by attempting to match only feature points that have a unique correspondence. The first algorithm, *Mirror Match (MM)*, is inspired by a simple but novel idea: If a given feature point in one image is better matched with other feature points from the *same* image than points in the other image, then any matches from this feature point to points in the other image are considered unreliable and should be discarded. This approach carries no implicit assumptions about the geometric consistency of matches and as such can easily be extended with other geometric solutions when appropriate or necessary. *Mirror Match with Clustering (MMC)* builds upon this idea by trying to group similar feature points and matching them separately from the feature points that have unique correspondences. Our experimental results show that both approaches generally outperform existing methods when tested on partially overlapping images.

The paper is organized as follows. Section II describes the proposed *Mirror Match (MM)* and *Mirror Match with Clustering (MMC)* algorithms. Section III presents the dataset and its use for the evaluation. Section IV discusses the benchmarking results obtained. Section V concludes the paper.

II. Matching Methods

A. *Mirror Match (*MM*)*

The central idea behind *MM* is to match features of n images by taking every feature from all n images and matching them against every other feature from the same set. We can then discard the correspondences that match two points within the same image. Algorithm 1 details the implementation of *MM*. In the acquisition stage we gather all features in the set of images. In the matching stage these features are matched using k-nearest neighbors. For any given feature f_i the two most similar neighbors are returned, and we calculate the ratio between them as proposed in [1]. Any correspondence with a ratio above the threshold supplied will be discarded. Finally in the filter stage we check that matches are from different images and discard all matches that are not symmetric.

Algorithm 1 Mirror Match (*MM*)

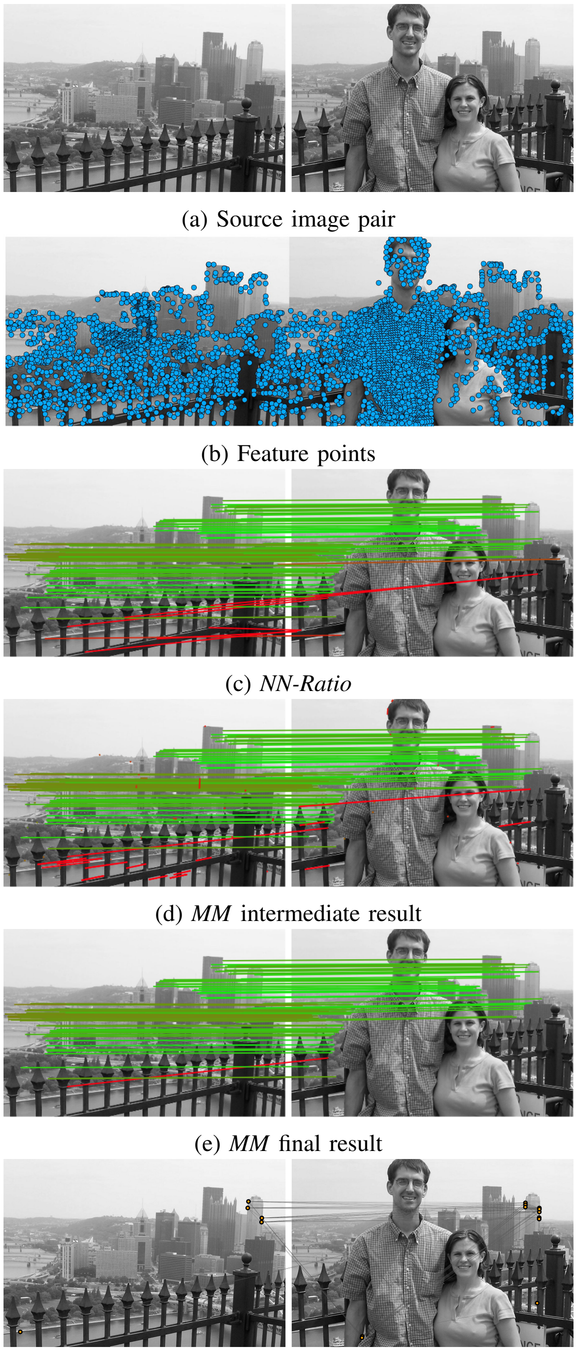
```
Require: $images$ : set of images, $t \in \mathbb{R}$
  $M_{init} \leftarrow \varnothing$, $M_{final} \leftarrow \varnothing$, $F \leftarrow \varnothing$
  for all $I_i \in images$ do                                  ▷ Acquisition Stage
    $F \leftarrow F \cup getF(I_i)$
  end for
  for all $f_i \in F$ do                                       ▷ Matching Stage
    $f_m, f_n \leftarrow get2NearestNeighbors(f_i, F \setminus \{f_i\})$
    $ratio \leftarrow distance(f_i, f_m)/distance(f_i, f_n)$
    if $ratio < t$ then
      $M_{init} \leftarrow M_{init} \cup (f_i, f_m)$
    end if
  end for
  for all $(f_i, f_j) \in M_{init}$ do                         ▷ Filter Stage
    if $(f_j, f_i) \in M_{init} \land getImg(f_i) \neq getImg(f_j) \land (f_j, f_i) \notin M_{final}$ then
      $M_{final} \leftarrow (f_i, f_j)$
    end if
  end for
  return $M_{final}$
```

Figure 1 illustrates the benefits of *MM* using an example image pair from the Gallagher dataset [12]. With *NN-Ratio* (Figure 1c), many incorrect matches occur in the fence towards the bottom of the image. When we match all feature points together, many of these incorrect matches are eliminated, because points in the fence match with other points in the fence in the same image (Figures 1d and 1e).

B. *Mirror Match with Clustering (*MMC*)*

In contrast to *MM*, *MMC* diverges from traditional non-geometric feature matching by clustering feature points by similarity. This process yields partitions of fairly similar feature points that we can match using the same approach as *MM*. Algorithm 2 shows the pseudocode implementation of *MMC*.

We use the Louvain Method [13] for clustering feature points, since it is relatively fast and performs well [14], does not require parameters [13], and does not emphasize partitions



(a) Source image pair

(b) Feature points

(c) *NN-Ratio*

(d) *MM* intermediate result

(e) *MM* final result

(f) *MMC* Partition Example

Fig. 1: Feature matching with *MM* and *MMC*. Dots represent feature points; green/red lines indicate correct/incorrect matches, respectively; black lines represent edges in the feature graph. (c) Result of *NN-Ratio* matching. (d) All matches found by *MM*, including intra-image matches. (e) Final *MM* result. (f) Example of a partition of feature points after clustering, which includes similar feature points from the same building across both images.

Algorithm 2 Mirror Match with Clustering (*MMC*)

```
Require: $images$ : set of images, $t \in \mathbb{R}$, $\alpha \in \mathbb{R}$
  $M \leftarrow \varnothing$, $F \leftarrow \varnothing$
  for all $I_i \in images$ do                          ▷ Gather features
      $F \leftarrow F \cup getFeatures(I_i)$
  end for
  $A \leftarrow getAdjacencyMatrix(f_1, f_2, \dots, f_n)$
  $A_{norm} \leftarrow normalize(A)$
  for all $(i, j) \in indices(A)$ do                   ▷ Prune edges
      if $A_{norm}[i, j] < \alpha$ then
          $A_{norm}[i, j] \leftarrow 0$
      end if
  end for
  $P \leftarrow cluster(A_{norm})$
  for all $p \in P$ do                    ▷ p is a set of feature points
      $M \leftarrow M \cup getMatches(p, t, F)$
  end for
  return matches
```

of equal size, as opposed to spectral clustering or k-means [15], for example. While the Louvain clustering algorithm does not require any parameters in itself, it tends towards clustering all feature points together in the same partition if the graph is very connected. To ensure that the graph is well clustered, the adjacency matrix is pruned so that only edges above a certain threshold are kept. From empirical analysis, retaining the top 2.5% of edges with the highest similarity seems to work well in practice. Figure 2 shows the result of clustering the feature points as a graph.

The partitions group feature points by similarity, which means that repetitive structures such as buildings often appear in larger partitions, as exemplified in Figure 1f.

The matching algorithm for *MMC*, *getMatches*, finds matches within all partitions with more than two elements using the *MM* approach. However, as can be seen in the example in Figure 2, many of the partitions contain only two feature points from different images linked by one edge. In such a case, we compare the similarity of the these two feature points with their second best match and remove matches where this ratio lies above a certain threshold, like in the *NN-Ratio* algorithm. For example in the case of Figure 1f, we have several feature points from a building in one image grouped together with points from the same building in another image.

III. Experiments

To reliably measure the accuracy of a matching method on real images, we either need a set of image pairs tied by a homography, or we have to manually count the number of inliers. The latter becomes prohibitive for large numbers of (non-trivial) images.

Mikolajczyk and Schmid [4] introduced a set of test images to compare the performance of feature detectors. The set covers different types of image variations, such as lighting change, blur, rotation, and viewpoint change. Inspired by this dataset (in particular the 'Graf' image set) and motivated by the need for more image pairs featuring viewpoint changes, we have compiled the *Murals* dataset.[1] It comprises 8 image pairs that feature murals taken from two different angles, which makes it possible to relate points in the image pairs with a homography. The images were collected from Flickr's database of photos published under a creative commons licence and have been cropped to show the same motive and resized to 900 × 600 pixels. Figure 4 shows one image from each pair.

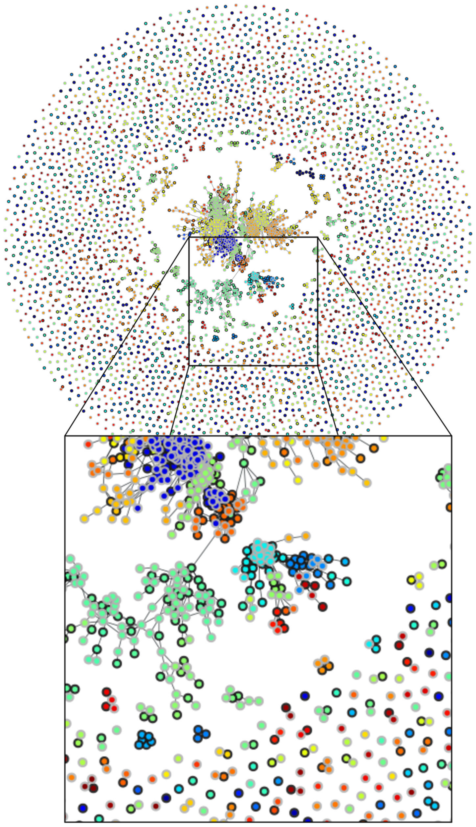


Fig. 2: The partitioned feature graph. Each vertex represents a feature point; lines indicate high similarity between points. A partition is a connected group with the same color. The border color of each node indicates which image it belongs to. Zooming into a section of the graph, the various cluster sizes can be seen, ranging from hundreds of feature points to only two or three.

Fig. 3: Images 1-5 of the Graf set from [4].

We compare the *MM* and *MMC* algorithms to *NN-Ratio* [1] as well as *Isodata* [10] and *Spectral* [9]. *Isodata* and *Spectral* use geometric constraints, whereas *NN-Ratio* does not. The comparison was done using the *Murals* dataset (Figure 4), the *Graf* set (Figure 3) from [4], and two images

[1] The set of source images and homographies as well as the script to generate the cropped test sets based on them can be found at http://vintage.winklerbros.net/murals.html

Fig. 4: Images in the *Murals* dataset.

Test sets were generated from the image pairs by cropping square patches of 250×250 pixels with a random vertical and horizontal offset. Given a source image pair, we produce 100 pairs of patches, which might or might not overlap. This ensures that patch pairs with no overlap still retain a general similarity to each other, while patches that do overlap often only share a small part of their area. Figure 5 shows an example of possible pairs of test image patches produced from a source image pair.

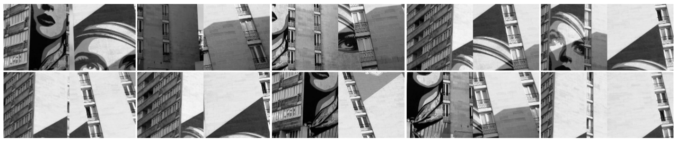


Fig. 5: Sample test patches produced from an image pair.

Producing n such pairs allows us to test not just how well the matching algorithm performs on a variety of overlaps but also how many false positives we get on similar images that do not overlap. In practice the amount of overlap between pairs in a test set will depend on the overlap and viewpoint change in the source image pair. To give a rough idea, Table I shows the overlap of patch pairs created from images 1 and 3 of the *Graf* image set from [4].

TABLE I: Overlap in the set of 100 patch pairs created from two images of the *Graf* image set (Figure 3).

Amount of overlap:	0%	$< 50\%$	$> 50\%$
Number of patch pairs:	21	54	25

Given a potential match between two pixels p_1 and p_2, $m = (p_1, p_2)$, and a homography H relating the two images I_1 and I_2, we can calculate if m is an inlier by checking if the two points satisfy the following criteria:

$$|Hp_1 - p_2| + \left|H^{-1}p_2 - p_1\right| < d_{\max}$$

That is, the distance between p_1 translated to I_2 and p_2 *plus* the distance between p_2 translated to I_1 should be less than a certain threshold (we use $d_{\max} = 5$ pixels here).

IV. Results

Figure 6 shows the results for 100 patch pairs generated from images 1 and 3 of the *Graf* image set (cf. Figure 3). We plot the precision and recall for *MMC*, *MM*, *Isodata*, *Spectral* and *NN-Ratio*, which are calculated as follows:

$$\text{Precision} = \frac{\#\text{Correct Matches}}{\#\text{Correct Matches} + \#\text{False Matches}}$$

$$\text{Recall} = \frac{\#\text{Correct Matches}}{\#\text{Total Possible Matches}}$$

The plot data is obtained by varying ratio thresholds. The results show that *MM* and *MMC* consistently outperform *NN-Ratio*; *MMC* generally lies 2-3 percentage points above *MM* when both are performing at optimal accuracy. Although *Isodata* and *Spectral* exhibit good performance on strict thresholds (small number of matches), that quickly diminishes when more matches are desired.

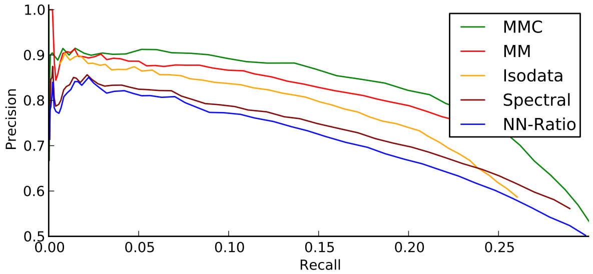


Fig. 6: Results for 100 patch pairs generated from image pair 1&3 from the *Graf* set.

To validate whether these results generalize to other images, we tested the five algorithms on the *Murals* dataset as well as the *Graf* pair tested above. In total 900 different patch pairs were generated from 9 source image pairs. The results are shown in Figure 7.

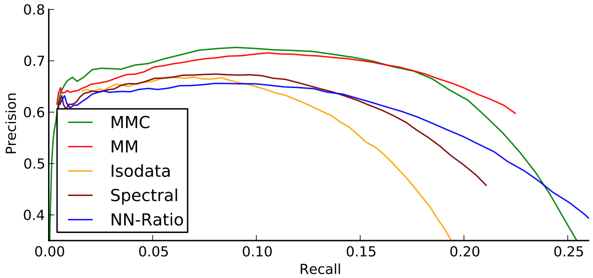


Fig. 7: Results for 900 patch pairs extracted from the *Murals* dataset and the image pair 1&3 from the *Graf* set.

To further investigate the impact of viewpoint changes, we tested the algorithms on the *Graf* image set (Figure 3), which contains 5 images of the same mural taken with gradually increasing viewpoint changes. The first images are almost identical, while the last are taken from very different angles. The results from *MMC* and *NN-Ratio* as shown in Figure 8, confirm that *MMC* is generally superior to *NN-Ratio* across viewpoint changes. The performance of *MM* (not shown in the plot) is similar to *MMC*.

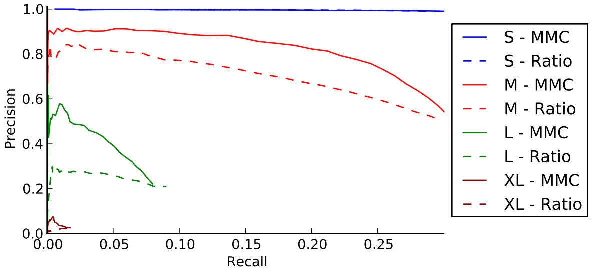


Fig. 8: Results for viewpoint changes using the *Graf* set from [4]. S: img1&2; M: img1&3; L: img1&4; XL: img1&5.

Finally, for an example of a real life use case, Figure 9 shows the results on 100 patch pairs generated from a typical holiday photo shot (Figure 1a) featuring occlusion and a slight viewpoint change from the Gallagher dataset [12]. The performance is comparable to the murals, despite the lack of a simple homographic mapping between the images.

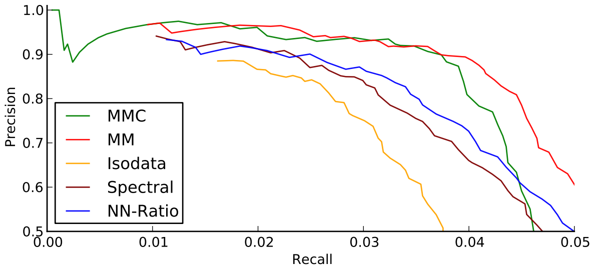


Fig. 9: Results for the image pair in Figure 1a.

In terms of computational complexity, *NN-Ratio*, *MM*, and *Isodata* can be implemented in $O(n \log n)$, where n is the total number of feature points. *Spectral* and our current *MMC* implementation have a complexity of $O(n^2)$ due to the construction of a similarity matrix of the feature points. However, both can be approximated in $O(n \log n)$ using search trees to sparsely populate the adjacency matrix.

In terms of speed, Table II shows the running time of the four algorithms over 100 image pairs of 250×250 pixels. These numbers should be taken with a grain of salt, given that much of the code behind *MMC* and *Isodata* is implemented in Python, whereas *MM* and *NN-Ratio* make use of OpenCV to execute computationally intensive operations in C++, which makes them much faster.

TABLE II: Running times as tested on a Intel® Core$^{\text{TM}}$ i5-3550 CPU @ 3.30 GHz with 8 GB memory.

Algorithm:	*Ratio*	*MM*	*MMC*	*Isomatch*	*Spectral*
Running time:	21s	23s	2722s	1146s	994s

V. Summary

We have addressed the problem of matching feature points without using geometrical constraints, proposing *Mirror Match (MM)* and *Mirror Match with Clustering (MMC)*. The two algorithms share the common idea that feature points should have better matches in another image than in the image they came from to be considered good matches. *MMC* further improves on this idea by using the structure of the similarity graph of the feature points.

The algorithms show promising results when tested on the *Murals* data set. *MM* and *MMC* generally outperform existing matching algorithms *NN-Ratio*, *Spectral* and *Isodata*, and *MMC* outperforms *MM*. We show that this result generalizes to variations in viewpoint change as well as more realistic photos featuring occlusions.

Given the versatility of the proposed algorithms, we are planning to apply them to problems that require high reliability faced with images that might not match, such as near duplicate detection or image retrieval.

*This work is supported by the research grant for ADSC's Human Sixth Sense Programme from Singapore's Agency for Science, Technology and Research (A*STAR).*

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer Vision, European Conference on*, pp. 404–417, 2006.

[3] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[4] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[5] J. Kim, O. Choi, and I. S. Kweon, "Efficient feature tracking for scene recognition using angular and scale constraints," in *Proc. Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, 2008, pp. 4086–4091.

[6] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[7] O. Chum and J. Matas, "Matching with PROSAC – progressive sample consensus," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, vol. 1, 2005, pp. 220–226.

[8] O. Choi and I. S. Kweon, "Robust feature point matching by preserving local geometric consistency," *Computer Vision and Image Understanding*, vol. 113, no. 6, pp. 726–742, 2009.

[9] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, vol. 2, 2005, pp. 1482–1489.

[10] M. Das, J. Farmer, A. Gallagher, and A. Loui, "Event-based location matching for consumer image collections," in *Proc. Content-based Image and Video Retrieval, ACM International Conference on*, 2008, pp. 339–348.

[11] L. Wu, Y. Niu, H. Zhang, H. Zhu, and L. Shen, "Robust feature point matching based on local feature groups (LFGs) and relative spatial configuration," *Journal of Computational Information Systems*, vol. 7, no. 9, pp. 3235–3244, 2011.

[12] A. Gallagher and T. Chen, "Clothing cosegmentation for recognizing people," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2008.

[13] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[14] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Physical Review E*, vol. 80, no. 5, p. 056117, 2009.

[15] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.