Pattern Recognition Letters
journal homepage: www.elsevier.com

# A General Framework For Image Feature Matching Without Geometric Constraints

Jonas Toft Arnfred, Stefan Winkler**

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

*ABSTRACT*

Computer vision applications that involve the matching of local image features frequently use *Ratio–Match* as introduced by Lowe and others, but is this really the optimal approach? We formalize the theoretical foundation of *Ratio-Match* and propose a general framework encompassing *Ratio-Match* and three other matching methods. Using this framework, we establish a theoretical performance ranking in terms of precision and recall, proving that all three methods consistently outperform or equal *Ratio-Match*. We confirm the theoretical results experimentally on over 3000 image pairs and show that matching precision can be increased by up to 20 percentage-points without further assumptions about the images we are using. These gains are achieved by making only a few key changes of the *Ratio-Match* algorithm that do not affect computation times.



## 1. Introduction

Matching image points is a crucial ingredient in almost all computer vision applications that deal with sparse local image features, such as image categorization (Bosch et al., 2008), image stitching (Brown and Lowe, 2007), object detection (Zhang et al., 2007), and near duplicate detection (Zhao and Ngo, 2009), to mention just a few examples. All of these rely on accurately finding the correspondence(s) of a point on an object in a query image given one or more target images that might contain the same object. In many applications the target images have undergone transformations with respect to the query image; in stereo vision, the viewpoint is different, while in object recognition and near duplicate detection both the lighting and even the object itself may also be transformed.

In the literature two approaches to feature point matching have been pursued and later merged, namely the *geometric approach* and the *descriptor-centric approach*.

In the purely geometric approach, feature points are matched based on their location in the images. Scott and Longuet-Higgins (1991) and Shapiro and Brady (1992) introduced the use of spectral methods by deriving a coherent set of matches from the eigenvalues of the correspondence matrix. Other examples of this approach include (Sclaroff and Pentland, 1995; Carcassoni and Hancock, 2003).

The descriptor-centric approach on the other hand finds matches by pairing similar keypoints. The first examples of this approach used the correlation of the raw image data immediately surrounding the feature point (Deriche et al., 1994; Baumberg, 2000) to calculate this similarity. Later algorithms were enhanced by invariant feature descriptors, as first introduced by Schmid and Mohr (1997) and later popularized by the work of Lowe (2004) introducing SIFT and Bay et al. (2006) introducing SURF.

A straightforward way of finding a set of correspondences using only feature points is to apply a threshold to the similarity measure of the feature vectors, accepting only correspondences that score above a certain level of similarity (Szeliski, 2010). When we match images with the assumption that the correspondence between two feature points will be unique, we can further increase precision by only matching a feature point to its nearest neighbor in terms of descriptor similarity. Instead of thresholding based on similarity, Deriche et al. (1994) and Baumberg (2000) proposed using the *ratio* of the similarity of the best to second best correspondence of a given point to evaluate how unique it is. Their finding was later tested by several independent teams, all concluding that thresholding based on this ratio is generally superior to thresholding based on similarity (Lowe, 2004; Mikolajczyk and Schmid, 2005; Moreels and Perona, 2007; Rabin et al., 2009). Brown et al. (2005) extended this "ratio-match" idea to deal with a set of images by using not the ratio of the best and second best correspondence, but the average ratio of the best and the second best correspondences across a set of images. Rabin et al. (2009) tried to enhance descriptor matching by looking at the statistical distribution of

**Corresponding author: Tel.: +65-6591-7760; fax: +65-6591-9091;
*e-mail:* stefan.winkler@adsc.com.sg (Stefan Winkler)

local features in the matched images, and only return a match when such a correspondence would not occur by mere chance. Finally, a precursor of the algorithms discussed in this paper was introduced by the authors as *Mirror-Match*, which makes use of the feature points in both images to decide if a match is valid (Arnfred et al., 2013).

A plethora of hybrid solutions have combined descriptor matching with various geometric constraints to improve matching. These constraints are based on assumptions regarding the transformation between query and target images. At the stricter end we have epipolar constraints, assuming that images can be tied by a homography (Torr and Zisserman, 2000; Chum and Matas, 2005), and angular constraints, assuming correspondences are angled similarly (Kim et al., 2008; Schmid and Mohr, 1997). Often these approaches are made computationally feasible by modeling feature correspondences as an instance of graph matching, where each feature is a vertex, and edge values correspond to a geometric relation between two features. Approximate graph matching algorithms can then be used to efficiently establish an isomorphism between the feature graphs of two images (Leordeanu and Hebert, 2005; Torresani et al., 2008; Yarkony et al., 2010; Yuan et al., 2012). Others define image regions and reject or accept correspondences based on the regions they connect (Cho et al., 2009; Wu et al., 2011).

Any matching method relying on geometric constraints is limited by inherent assumptions about the geometric relationship between the two images. Broad assumptions such as the epipolar constraint only apply in simple image transformations. For more complex transformations we need models suitable for each particular case, which restricts them to the subset of images that fit the model. Transformations from one scene to another often feature a change in perspective, background, and sometimes variations within the object itself: a person can change pose, a car model can have different configurations, a flower can bloom etc. When matching these instances we are forced to either create a sophisticated model that represents the variables of transformation within the object, or alternatively find correspondences using an algorithm with no inherent geometric assumptions. Besides, any geometric method acts as a filter on a given set of correspondences. Therefore, if the initial set of purely descriptor-based matches contains fewer incorrect correspondences, the final set can be calculated faster and more accurately.

The methods we propose in this paper are designed to be free from assumptions about image geometry. They extend and improve on *Ratio-Match* (Lowe, 2004) and the authors' *Mirror-Match* (Arnfred et al., 2013) by generalizing both algorithms to a framework of matching methods. We go on to formally establish a ranking based on how different methods within the framework compare in terms of precision and recall. Our experimental evaluations confirm the theoretical results and show that *Ratio-Match* is generally a sub-optimal choice as a matching algorithm.

In our previous paper (Arnfred et al., 2013), we introduced *Mirror-Match* and *Mirror-Match with Clustering*, two algorithms that outperform the state of the art. The novel contributions of the present paper consist of presenting these algorithms together with several related existing algorithms in a general and comprehensive framework. We further develop the theoretical foundations for comparing the algorithms and use these to formally prove a ranking in terms of performance for the different algorithms. This also enables us to understand why *Mirror-Match* performs better than *Ratio-Match* in the first place. In addition we benchmark all algorithms within the framework extensively on a much larger dataset containing over 3000 image pairs.

The paper is organized as follows. Section 2 introduces the original *Ratio-Match* and extends it to introduce the proposed framework. Section 3 compares the various methods of the framework theoretically. Section 4 presents an experimental evaluation and discusses the results obtained. Section 5 concludes the paper.

## 2. Matching Framework

### *2.1. Definitions*

The proposed framework is inspired by *Ratio-Match* as introduced by Deriche et al. (1994) and later used by Baumberg (2000) and Lowe (2004). *Ratio-Match* is motivated by the observation that nearest-neighbor feature matching is not necessarily the best strategy (Lowe, 2004; Mikolajczyk and Schmid, 2005). The distance between the feature descriptors of two nearest neighbors might tell us on a global level how much they resemble each other, but it does not tell us if other feature points are equally similar. *Ratio-Match* makes use of the ratio between the nearest and second nearest neighbor as a heuristic to determine the confidence of the match. Matches are returned only if this ratio is lower than a given threshold $\tau$, filtering out feature points that are ambiguous because others match almost equally well.

The underlying assumption in *Ratio-Match* is that the point we seek to match in the query image has only one true correspondence in a given target image or no matches at all. In both cases we can infer that the *second* nearest neighbor in the target image is not a true correspondence. We consider the distance between the second nearest neighbor and the feature point as the baseline. It tells us how similar the descriptors of two feature points can be when they are not a true correspondence. Some feature points might have very unique descriptors with large distances to false correspondences, while others may be generic with plenty of similar points. Knowing the baseline for all features allows us to be lenient in the first case and cautious in the second. In practice *Ratio-Match* scores a match by dividing the distance to the nearest neighbor with the distance to the second nearest neighbor (the baseline) to estimate how distinct the correspondence is from a false match.

In what follows we will use the following nomenclature:

- Let $f_q$ be a feature point in the query image.
- Let $\mathcal{F}$ be a set of features. $\mathcal{F}_t$ denotes all features from the target image.
- Let $\tau \in [0 \dots 1]$ be a threshold used to decide whether to keep a match.

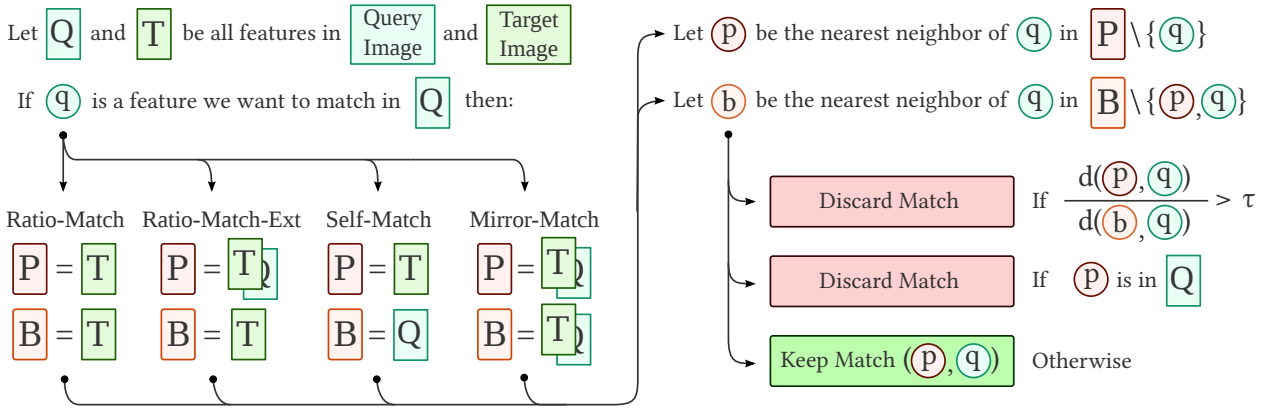



Fig. 1: Flow chart of the feature matching framework. $\tau$ is the ratio threshold, and $d(x, y)$ is the distance between two feature descriptors $x$ and $y$.

- Let the *proposed match* be the nearest neighbor of a query feature $f_q$ picked from a set of feature points that we call the *proposal set* $\mathcal{F}_p$, which does not contain the query feature.

- Let the *baseline match* be the nearest neighbor of the query feature $f_q$ picked from a set of feature points that we call the *baseline set* $\mathcal{F}_b$, which contains neither the proposed match nor the query feature. In *Ratio-Match* the baseline match is the second nearest neighbor.

### 2.2. Framework of Matching Methods

We can generalize *Ratio-Match* by expanding on the idea of baseline and proposal sets. With *Ratio-Match* these two sets are created from features in the target image, but this is not the only option. If we use the features in the query image as well as the combined features of both images, we end up with six possible permutations of a *Ratio-Match*-like algorithm. We illustrate these variants in Figure 2 and will go on to prove theoretically and demonstrate empirically that *Ratio-Match* is among the least performant of the pack.

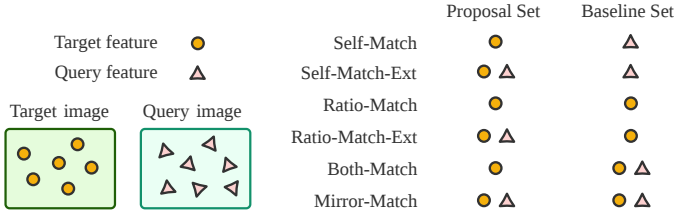



Fig. 2: Graphical representation of the baseline set and proposal set for different methods in the proposed framework.

The algorithms *Ratio-Match*, *Self-Match* and *Both-Match* all find the best match to a given query feature only in the target image. They differ by the feature set used as the baseline set. While *Ratio-Match* uses features from the target image, *Self-Match* draws the baseline set from the query image. Finally *Both-Match* uses the conjunction of features from both images.

Each of these three algorithms has an 'extended' case where the proposal set is replaced with the conjunction of features from both the target image and the query image. The extended version of *Self-Match* is *Self-Match-Ext*, and the extended version of *Ratio-Match* is *Ratio-Match-Ext*. Finally we call the extended version of *Both-Match* "*Mirror-Match*" because we originally introduced it under this name (Arnfred et al., 2013). We will not discuss the *Self-Match-Ext* and *Both-Match* variants in this paper since we prove them to be equivalent with *Self-Match* and *Mirror-Match* respectively in Section 3.

Figure 1 shows the flow chart of the generalized matching framework when proposal set and baseline set are defined according to Figure 2. Algorithm 1 illustrates in more detail how the framework can be implemented.

**Algorithm 1** Generalized matching algorithm for two images.

**Require:** $I_q, I_t$ : images, $\tau \in [0, 1]$
  $\mathcal{F}_q = get\_features(I_q)$
  $\mathcal{F}_t = get\_features(I_t)$
  $\mathcal{F}_{proposal-all} = get\_proposal\_features(\mathcal{F}_q, \mathcal{F}_t)$
  $\mathcal{F}_{baseline-all} = get\_baseline\_features(\mathcal{F}_q, \mathcal{F}_t)$
  $\mathcal{M} = \varnothing$
  **for all** $f_q \in \mathcal{F}_q$ **do**
    $\mathcal{F}_p = \mathcal{F}_{proposal-all} \setminus \{f_q\}$
    $f_p \leftarrow getNearestNeighbor(f_q, \mathcal{F}_p)$
    $\mathcal{F}_b = \mathcal{F}_{baseline-all} \setminus \{f_q, f_p\}$
    $f_b \leftarrow getNearestNeighbor(f_q, \mathcal{F}_b)$
    $r \leftarrow distance(f_q, f_p) / distance(f_q, f_b)$
    **if** $(r < \tau) \wedge (f_p \in \mathcal{F}_t)$ **then**
      $\mathcal{M} \leftarrow \mathcal{M} \cup \{f_q, f_p\}$
    **end if**
  **end for**
  **return** $\mathcal{M}$

The main distinguishing factor between the algorithms in the framework is the final ratio between the distance of the two nearest neighbors of a query feature, which determines if we keep or discard a match.

We can define a calculation of this ratio that is common to all algorithms in the framework, which allows us to compare the algorithms theoretically. To do so, we introduce the uniqueness ratio $r$, based on the concept of a nearest neighbor.

Given a feature $f_i$ and a set of features $\mathcal{F}$, the nearest neigh-

bor of $f_i$ in $\mathcal{F}$ is calculated as follows:

$$\arg\min_{f_j \in \mathcal{F}} d(f_i, f_j).$$

Here, $d(f_i, f_j)$ is the distance between feature descriptors. With SIFT and SURF this is the Euclidean distance of the feature vectors (Lowe, 2004; Bay et al., 2006), whereas BRIEF, BRISK, and FREAK use the Hamming distance (Leutenegger et al., 2011; Calonder et al., 2010; Alahi et al., 2012).

Let $f_p \in \mathcal{F}_p$ and $f_b \in \mathcal{F}_b$ be the nearest neighbors of a feature $f_q$ in $\mathcal{F}_p$ and $\mathcal{F}_b$. The uniqueness ratio is defined as follows:

$$\begin{aligned} r &= \mathrm{r}(f_q, \mathcal{F}_p, \mathcal{F}_b) \\ &= \frac{d(f_q, f_p)}{d(f_q, f_b)}. \end{aligned}$$

Referring again to Figure 1, for some methods it is possible that the proposed match is a feature from the query image, in which case we discard the match as a false correspondence. It also happens that we encounter correspondences with $r > 1$, in which case the match is also discarded. Take for example the case of *Self-Match* where we might find that the nearest neighbor of a feature in the target image is further from the query feature than the nearest neighbor in the query image. In this case the match is discarded.

## 3. Proofs of Algorithm Performance

### 3.1. Assumptions

Under the following three assumptions, which largely reflect the performance of matching feature points in practice, and which will be discussed in more detail in Section 3.6, we can theoretically compare the performance of the different algorithms shown in Figure 1:

1. For any point in the query image there is at most one real correspondence in the target image and no real correspondence in the query image (as assumed by *Ratio-Match*).
2. The distance between two feature descriptors within the query image is larger than their distance to a real correspondence in the target image. More precisely, given $f_q$, a feature from the query image for which a real correspondence, $f_{match}$, exists in the target image, we assume that $\forall f_i \in \mathcal{F}_q : d(f_q, f_{match}) < d(f_q, f_i)$.
3. For a set of query features $\mathcal{F}_q$ with true correspondences in the target image, the distribution of uniqueness ratios using $\mathcal{F}_t$ as the baseline set is similar to using $\mathcal{F}_q$, since the two images are bound to share part of the same scene in the case of true correspondences.

The performance in terms of precision and recall of any algorithm in the proposed framework is uniquely identified by the uniqueness ratio $r$. To show this, let $K$ be the number of possible true correspondences between query and target image. Precision and recall are defined as:

$$\begin{aligned} \text{Precision} &= \frac{\#\text{Correct}}{\#\text{Correct} + \#\text{Incorrect}} \\ \text{Recall} &= \frac{\#\text{Correct}}{K} \end{aligned}$$

Given feature $f_p \in \mathcal{F}_p$ as the nearest neighbor of some query feature $f_q \in \mathcal{F}_q$, we define the set of features $\mathcal{F}_{true}$ as $\left\{ f_q \in \mathcal{F}_q \mid f_p \text{ is a true correspondence of } f_q \right\}$ and $\mathcal{F}_{false}$ as $\mathcal{F}_q \setminus \mathcal{F}_{true}$. We can then define the number of correct and incorrect matches as follows:

$$\begin{aligned} \text{keep\_match}(f_q) &= \begin{cases} 1 & \text{if } \mathrm{r}(f_q, \mathcal{F}_p, \mathcal{F}_b) < \tau \\ 0 & \text{otherwise} \end{cases} \\ \#\text{Correct} &= \sum_{f_q \in \mathcal{F}_{true}} \text{keep\_match}(f_q) \\ \#\text{Incorrect} &= \sum_{f_q \in \mathcal{F}_{false}} \text{keep\_match}(f_q) \end{aligned}$$

If $r$ is identical across two matching methods for every query feature $f_q$, they return identical results.

### 3.2. Ratio-Match and Ratio-Match-Ext

Based on the above assumptions we prove that *Ratio-Match-Ext* is equal to or better than *Ratio-Match* in terms of both precision and recall. Consider the nearest neighbor $f_p$ of a query feature $f_q$ and the two cases where we have either $f_p \in \mathcal{F}_q$ or $f_p \in \mathcal{F}_t$.

For $f_p \in \mathcal{F}_t$ the uniqueness ratio of *Ratio-Match-Ext* is:

$$\begin{aligned} r &= \mathrm{r}(f_q, \mathcal{F}_p, \mathcal{F}_b) \\ &= \mathrm{r}(f_q, \mathcal{F}_q \cup \mathcal{F}_t, \mathcal{F}_t) \\ &= \mathrm{r}(f_q, \mathcal{F}_t, \mathcal{F}_t). \end{aligned}$$

Since $\mathcal{F}_p = \mathcal{F}_t$ for *Ratio-Match*, the two algorithms behave identically when the nearest neighbor is found in the target image.

For $f_p \in \mathcal{F}_q$, *Ratio-Match* gives us the following ratio:

$$\begin{aligned} r &= \mathrm{r}(f_q, \mathcal{F}_p, \mathcal{F}_b) \\ &= \mathrm{r}(f_q, \mathcal{F}_t, \mathcal{F}_t). \end{aligned}$$

That is, *Ratio-Match* calculates the ratio based on the two nearest correspondences in the target image. *Ratio-Match-Ext* on the other hand does not return any correspondence, because the nearest neighbor is in the query image. Since query features with a nearest neighbor in the query image are false correspondences per assumption #1 and #2, this proves that *Ratio-Match-Ext* has superior precision to *Ratio-Match* while maintaining equal recall.

### 3.3. Mirror-Match and Ratio-Match-Ext

Next we show that *Mirror-Match* is equal or better than *Ratio-Match-Ext* in terms of precision and recall. Consider as before the nearest neighbor $f_p$ of a query feature $f_q$. When $f_p$ resides in the query image, both algorithms behave alike and discard the match. However, consider the uniqueness ratio of *Mirror-Match* for the case where $f_p$ resides in the target image:

$$\begin{aligned} r &= \mathrm{r}(f_q, \mathcal{F}_p, \mathcal{F}_b) \\ &= \mathrm{r}(f_q, \mathcal{F}_p, \mathcal{F}_q \cup \mathcal{F}_t) \\ &= \max(\mathrm{r}(f_q, \mathcal{F}_p, \mathcal{F}_t), \mathrm{r}(f_q, \mathcal{F}_p, \mathcal{F}_q)). \end{aligned}$$

For a true correspondence, the uniqueness ratio using $\mathcal{F}_q$ as a baseline is distributed similarly to the ratio when using $\mathcal{F}_t$ according to assumption #3. In this case the algorithm performs like *Ratio-Match-Ext*. However, for a false correspondence with a baseline match in the query image which is closer than the baseline match in the target image, *Mirror-Match* will return a worse uniqueness ratio. This in turns means that *Mirror-Match* is equal to or better than *Ratio-Match-Ext* in terms of precision while maintaining equal recall.

### *3.4. Self-Match and Self-Match-Ext*

Using a similar procedure we can prove the equivalence between *Self-Match* and *Self-Match-Ext*. We look at the nearest neighbor $f_{nn}$ of every query feature $f_q$ and consider the two cases of $f_{nn} \in \mathcal{F}_q$ and $f_{nn} \in \mathcal{F}_t$.

For $f_{nn} \in \mathcal{F}_t$ the uniqueness ratio of *Self-Match-Ext* is:

$$\begin{aligned} r &= \mathrm{r}(f_q, \mathcal{F}_p, \mathcal{F}_b) \\ &= \mathrm{r}(f_q, \mathcal{F}_q \cup \mathcal{F}_t, \mathcal{F}_b) \\ &= \mathrm{r}(f_q, \mathcal{F}_t, \mathcal{F}_b). \end{aligned}$$

Since $\mathcal{F}_p = \mathcal{F}_t$ for *Self-Match*, the two algorithms behave identically when the nearest neighbor is found in the target image.

For $f_{nn} \in \mathcal{F}_q$ we know that the best match for the query feature is in the query image. Since neither algorithm returns a correspondence within the same image, they also behave identically for this case. This proves that for any set of *query features* *Self-Match* returns the same matches as *Self-Match-Ext*.

### *3.5. Both-Match and Mirror-Match*

The proof of the equivalence between *Both-Match* and *Mirror-Match* is almost identical to the one for *Self-Match* and *Self-Match-Ext*. We look at the nearest neighbor $f_{nn}$ of every query feature $f_q$ and consider the two cases of $f_{nn} \in \mathcal{F}_q$ and $f_{nn} \in \mathcal{F}_t$.

For $f_{nn} \in \mathcal{F}_t$ the uniqueness ratio of *Mirror-Match* is:

$$\begin{aligned} r &= \mathrm{r}(f_q, \mathcal{F}_p, \mathcal{F}_b) \\ &= \mathrm{r}(f_q, \mathcal{F}_q \cup \mathcal{F}_t, \mathcal{F}_b) \\ &= \mathrm{r}(f_q, \mathcal{F}_t, \mathcal{F}_b). \end{aligned}$$

Since $\mathcal{F}_p = \mathcal{F}_t$ for *Both-Match*, the two algorithms behave identically when the nearest neighbor is found in the target image.

For $f_{nn} \in \mathcal{F}_q$ we know that the best match for the query feature is in the query image. Since neither algorithm returns a correspondence within the same image, they also behave identically for this case. This proves that for any set of *query features* *Both-Match* returns the same matches as *Mirror-Match*.

### *3.6. Discussion*

We have thus proven that – based on assumptions #1-3 – the algorithms in the framework behave as follows in terms of matching precision:

$$\begin{gathered} \textit{Ratio-Match} \leq \textit{Ratio-Match-Ext} \leq \textit{Mirror-Match} \\ \textit{Self-Match} = \textit{Self-Match-Ext} \\ \textit{Both-Match} = \textit{Mirror-Match} \end{gathered}$$

*Self-Match* is most closely related to *Ratio-Match* in that both the baseline set and the proposal set are created based on only one image. While the proposal set for both algorithms is based on the target image, *Self-Match* uses the query image for the baseline set, whereas *Ratio-Match* sticks with the target image. For cases with a lot of overlap between the target and query image they should perform similarly. However when we match images where the target image may not overlap at all with the query image, using the query image and not the target image as a baseline set seems like a reasonable choice, given that the baseline set would be closer to the feature matched in this case and more strictly rule out false correspondences.

The three assumptions underlying the formal ranking of algorithms presented above have been chosen to reflect conditions under which we would ideally match feature points. However, for each assumption there exist corner cases where it is no longer valid. In this section we discuss these corner cases in order to review the circumstances under which the absolute or relative performance of the algorithms might differ.

The first assumption states that every feature point has at most one unique match, which is often the case for natural images. However, for the recognition of object classes or in images with repetitive content, a point in the query image might have several true correspondences in the target image, so using a baseline match from the target image will lead to a much higher uniqueness ratio. However, as long as the query image does not contain repetitive objects, we can still use *Self-Match* without loss of precision. Conversely the query image could have several similar objects, which could negatively impact performance for all algorithms except *Ratio-Match* in cases where the target image only contains one such object. However, when applied to scene matching, we would normally expect there to be several similar objects in the target image too. Matching any of these points could easily lead to ambiguities. Here the uniqueness assumption plays to our advantage by forcing us to ignore features that will lead to ambiguous matches.

According to the second assumption, feature descriptors behave such that when matched, a true correspondence to a query feature will always be closer in distance than any other feature in the query image. For this assumption to be violated, either the query image has to contain repeating patterns (violating assumption #1), or the two images are so different (i.e. taken from very different viewpoints or under different lighting conditions) that the descriptor distance between two unrelated features ends up being closer than the distance between a feature and its true correspondence. In this case *Ratio-Match-Ext* and *Mirror-Match* would discard the match while *Ratio-Match* would still attempt to match the descriptor to a feature in the target image.

The empirical results of *Self-Match* (Figure 6) support the second assumption. *Self-Match* discards any match where a feature in the query image is more similar to the query feature, yet the algorithm consistently performs equally well or better than *Ratio-Match*, providing further evidence to the notion that for almost all true correspondences the best match is found in the target image. As we would expect, this assumption breaks down when we increase $\tau$ to accept matches with higher unique-

ness ratios, illustrated by the higher recall rates of *Ratio-Match* compared to *Self-Match* in most plots. Once we accept almost all proposed nearest neighbors as matches, we are bound to encounter more non-distinct feature points for which even true correspondences might have better matches amongst the query features.

The third assumption states that the uniqueness ratios using either the query or target image as the baseline set would be similar in distribution, given that the feature point we are matching has a true correspondence. We can support this assumption by looking at the uniqueness ratios returned by *Self-Match* and *Ratio-Match*, since they use the query and target image respectively to calculate them. Figure 3 shows the actual uniqueness ratios measured from 3024 image pairs featuring 3D objects (see Section 4 for more on this dataset). For ratios lower than 0.7, the uniqueness ratios are similar, but as we approach more lenient thresholds, the ratios based on the query image are higher than those for the target image. This means that the third assumption becomes invalid for more lenient thresholds, and we can no longer expect *Mirror-Match* to outperform *Ratio-Match-Ext*.

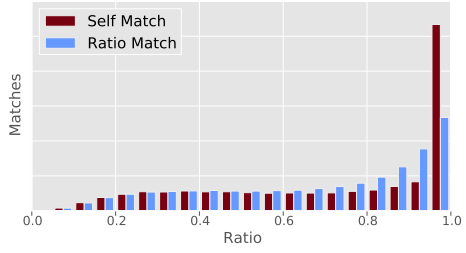



Fig. 3: Normalized histogram of the uniqueness ratios of true correspondences for 3024 image pairs.

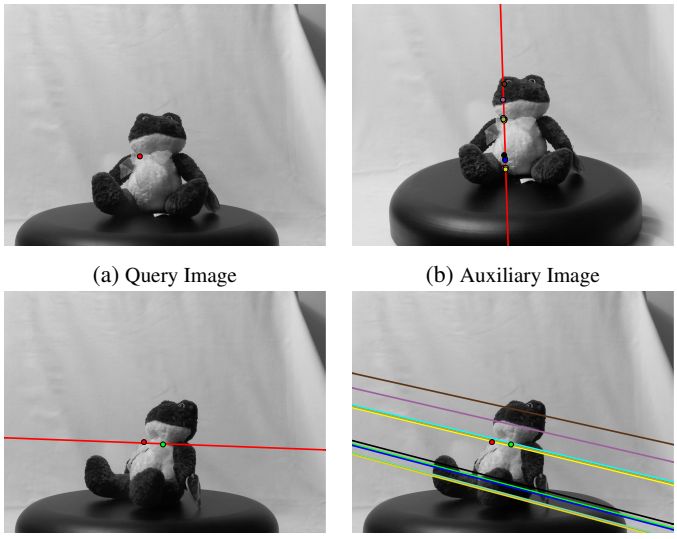
## 4. Experimental Evaluation

### *4.1. Database and Procedure*

We evaluate the matching algorithms of our framework on the 3D Objects database released by Moreels and Perona (2007). Sample images from the dataset are shown in Figure 4. We use images of 84 objects under three different lighting conditions at 12 different angle intervals, resulting in a total of 3024 image pairs.

Fig. 4: Samples from the 3D Objects dataset (Moreels and Perona, 2007).

To validate matches, Moreels and Perona (2007, p.266) proposed a method using epipolar constraints, which is outlined in Figure 5. According to their experiments, these constraints are able to identify true correspondences with an error rate of 2%.



(a) Query Image (b) Auxiliary Image

(c) Target Image (d) Epipolar constraints

Fig. 5: Creating epipolar constraints based on three source images (Moreels and Perona, 2007) (best viewed in color). (a) Query image marked with the position of the feature we are attempting to match. (b) Auxiliary image, taken at the same rotation as the query image but from a higher elevation angle. The line going through the image is the epipolar line of the feature point in the query image. The markers indicate all feature points in the image found near the epipolar line. (c) Target image rotated 45 degrees from the query image. The line overlaid on the image represents the epipolar line corresponding to the feature point shown in the query image. The markers indicate all feature points in the image found near the epipolar line. (d) Target image overlaid with the epipolar lines corresponding to all features shown in (b). A true correspondence should be found within a small distance of one of the intersections of the line in (c) and the lines in (d). In this particular case both feature points shown in (c) and (d) are potential true correspondences.

We use their proposed method to generate the ground truth for the evaluation of our framework.

To compute the total number of possible correspondences, we take all features in a query image and count how many have a feature in the target image which would satisfy the epipolar constraints outlined above. Features with no correspondences are not included in the set of features for testing. This is necessary because a relatively small number of actual true correspondences between folds in the background material were mistakenly counted as false positives when evaluated, which affects the precision of the test in particular in cases with few true correspondences. In practice very few feature points are excluded for this reason.

We evaluate all matching algorithms from our framework on the 3D Objects dataset by matching images at different angular intervals. For each object we pick the query image as the image taken at 10 degrees rotation for calibration stability. We then match this image with the same object turned an additional $\Delta$ degrees, $\Delta \in \{5, 10, \ldots, 60\}$. For every angle interval we compare images taken under 3 different lighting conditions as provided by the dataset.

### *4.2. Results*

Figure 6 shows the performance of the different matching methods in our proposed framework for 12 increasingly bigger angle differences. We use precision-recall plots to facilitate

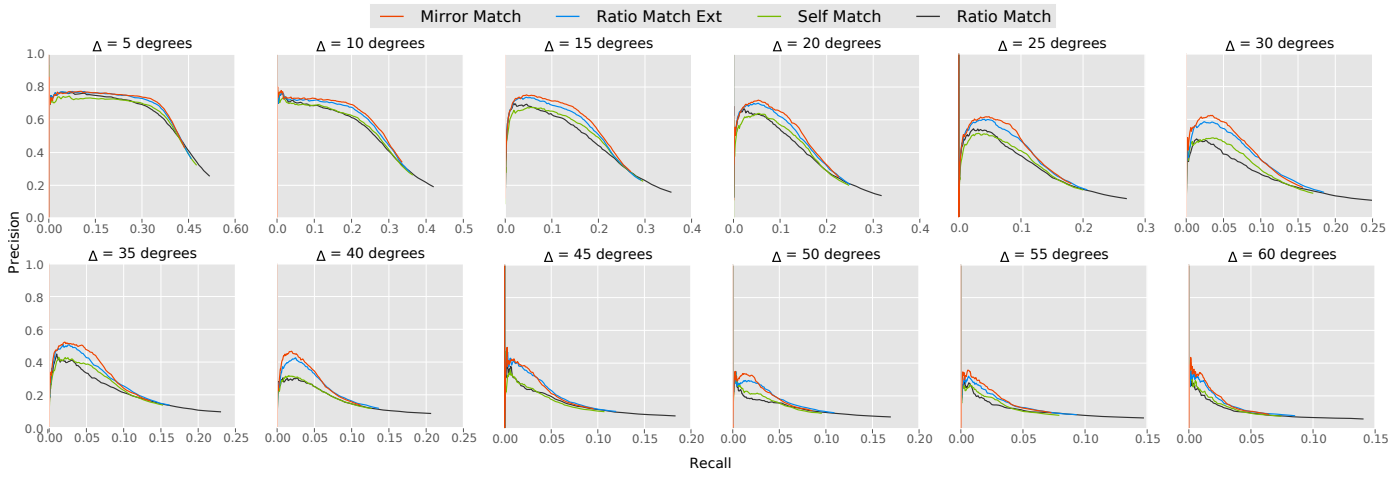



Fig. 6: Results for the 3D objects dataset. Each plot contains the results of 84 objects photographed under 3 different lighting conditions averaged over each image pair.

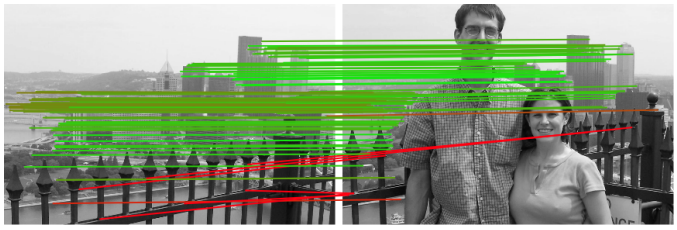
comparisons in terms of precision at similar levels of recall. For each plot we show the average results over all 3D objects, with equal weight given to each image pair.

*Ratio-Match* and *Self-Match* exhibit similar results, while *Mirror-Match* and *Ratio-Match-Ext* outperform both of them, showing the advantage of composing the proposal set of features from both images. *Mirror-Match* fares slightly but consistently better than *Ratio-Match-Ext*. In general we see the largest performance improvements at lower recall and a convergence of precision at higher recall. This is expected since a higher recall is a direct consequence of a more lenient threshold; as we let the threshold approach 1, we lose the benefits of thresholding on the uniqueness ratio, and all methods start approximating the results of a simple nearest neighbor match. For all but *Ratio-Match*, the features with better matches within the same image are still weeded out, which explains the tail end of *Ratio-Match* at high recall where the other algorithms no longer have results.

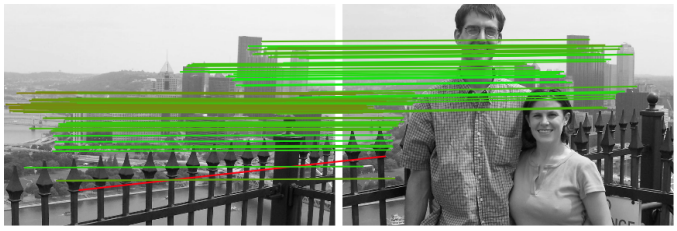
The performance gap between methods increases gradually with angle, reaching its maximum between 25 and 40 degrees, where *Mirror-Match* exhibits about 20 percentage-points higher precision than *Ratio-Match*. At larger viewpoint differences, the performance gain decreases. We suspect this is partly because assumption #2 breaks down when the images are transformed beyond a certain level of recognizability.

As we demonstrated previously (Arnfred et al., 2013), the removal of within-image matches improves performance on image pairs with partial or no overlap; within-image matches may occur for example when an image contains repeated structures, as shown in Figure 7. For no overlap we would expect a matching algorithm to reject matches within regions that have no matching counterparts in the other image.

To summarize, the methods perform as predicted on the 3D Objects dataset, with *Mirror-Match* in general performing better than *Ratio-Match-Ext*, which in turn outperforms *Ratio-Match*. The experimental results also show that the overall performance of *Self-Match* is equal to or better than *Ratio-Match*.



(a) *Ratio-Match*



(b) *Mirror-Match*

Fig. 7: *Ratio-Match* vs. *Mirror-Match* using an example image pair from the Gallagher and Chen (2008) dataset. Green/red lines indicate correct/incorrect matches, respectively. Using *Ratio-Match*, incorrect matches occur in the fence at the bottom of the image, because points on the fence match with other similar points in the same image (a). With *Mirror-Match*, most of these incorrect matches are eliminated (b).

### *4.3. Complexity and Speed*

In terms of computational complexity, the algorithms can be implemented in $O(n \log n)$, if we assume that both the query and the target image have $n$ feature points. For a target image with $m$ features where $m$ is significantly different from $n$, the complexities are as noted in Table 1. In practice the constant factors involved in the actual matching of two images with approximately the same number of feature points are usually small enough that all the algorithms run at very similar speeds. This is also shown in Table 1, which contains the running times as measured while matching 15 objects under three different lighting conditions (42 image pairs were matched in total). The running times are averaged over three separate runs of each algorithm implemented using the same data structures and libraries in Python.

Table 1: Complexity and average running times over 42 different image pairs with average $n = 237$ and average $m = 247$ feature points as tested on an Intel® Core™ i5-3550 CPU @ 3.30 GHz with 8 GB memory.

| Algorithm | Complexity | Avg Running Time |
|---|---|---|
| *Self-Match* | $O(n \log(nm))$ | 2.62s |
| *Ratio-Match* | $O(n \log(m))$ | 2.53s |
| *Ratio-Match-Ext* | $O(n \log(n+m))$ | 2.49s |
| *Mirror-Match* | $O(n \log(n+m))$ | 2.44s |

## 5. Conclusions

We have proposed a general framework of feature matching methods, building on the ideas behind *Ratio-Match* and *Mirror-Match*, and introducing the additional variants *Self-Match* and *Ratio-Match-Ext*. We formally proved under three assumptions that *Mirror-Match* performs better than or equal to *Ratio-Match-Ext* in terms of precision and recall, which in turn performs better than or equal to *Ratio-Match*.

The theoretical findings are confirmed by our experimental evaluation using images of rotated 3D objects, with *Mirror-Match* often outperforming *Ratio-Match* significantly over 3024 image pairs. These performance gains come for free in terms of both computational complexity as well as actual computing time.

## Acknowledgements

This study was supported by the research grant for the Human-Centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center (ADSC) from Singapore's Agency for Science, Technology and Research (A*STAR).

## References

Alahi, A., Ortiz, R., Vandergheynst, P., 2012. FREAK: Fast retina keypoint, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 510–517.

Arnfred, J.T., Winkler, S., Süsstrunk, S., 2013. Mirror Match: Reliable feature point matching without geometric constraints, in: Proc. IAPR Asian Conference on Pattern Recognition (ACPR), pp. 256–260.

Baumberg, A., 2000. Reliable feature matching across widely separated views, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 774–781.

Bay, H., Tuytelaars, T., Van Gool, L., 2006. SURF: Speeded up robust features, in: Proc. European Conference on Computer Vision (ECCV), pp. 404–417.

Bosch, A., Zisserman, A., Muoz, X., 2008. Scene classification using a hybrid generative/discriminative approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 712–727.

Brown, M., Lowe, D.G., 2007. Automatic panoramic image stitching using invariant features. International Journal of Computer Vision 74, 59–73.

Brown, M., Szeliski, R., Winder, S., 2005. Multi-image matching using multi-scale oriented patches, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 510–517.

Calonder, M., Lepetit, V., Strecha, C., Fua, P., 2010. BRIEF: Binary robust independent elementary features, in: Proc. European Conference on Computer Vision (ECCV), pp. 778–792.

Carcassoni, M., Hancock, E.R., 2003. Spectral correspondence for point pattern matching. Pattern Recognition 36, 193–204.

Cho, M., Lee, J., Lee, K.M., 2009. Feature correspondence and deformable object matching via agglomerative correspondence clustering, in: Proc. IEEE International Conference on Computer Vision (ICCV), pp. 1280–1287.

Chum, O., Matas, J., 2005. Matching with PROSAC – progressive sample consensus, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 220–226.

Deriche, R., Zhang, Z., Luong, Q.T., Faugeras, O., 1994. Robust recovery of the epipolar geometry for an uncalibrated stereo rig, in: Proc. European Conference on Computer Vision (ECCV), pp. 567–576.

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

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

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

Leutenegger, S., Chli, M., Siegwart, R.Y., 2011. BRISK: Binary robust invariant scalable keypoints, in: Proc. IEEE International Conference on Computer Vision (ICCV), pp. 2548–2555.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110.

Mikolajczyk, K., Schmid, C., 2005. A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1615–1630.

Moreels, P., Perona, P., 2007. Evaluation of features detectors and descriptors based on 3D objects. International Journal of Computer Vision 73, 263–284.

Rabin, J., Delon, J., Gousseau, Y., 2009. A statistical approach to the matching of local features. SIAM Journal on Imaging Sciences 2, 931–958.

Schmid, C., Mohr, R., 1997. Local grayvalue invariants for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 530–535.

Sclaroff, S., Pentland, A.P., 1995. Modal matching for correspondence and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 17, 545–561.

Scott, G.L., Longuet-Higgins, H.C., 1991. An algorithm for associating the features of two images. Proceedings of the Royal Society of London. Series B: Biological Sciences 244, 21–26.

Shapiro, L.S., Brady, M.J., 1992. Feature-based correspondence: An eigenvector approach. Image and Vision Computing 10, 283–288.

Szeliski, R., 2010. Computer Vision: Algorithms and Applications. Springer.

Torr, P.H.S., Zisserman, A., 2000. MLESAC: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding 78, 138–156.

Torresani, L., Kolmogorov, V., Rother, C., 2008. Feature correspondence via graph matching: Models and global optimization, in: Proc. European Conference on Computer Vision (ECCV), pp. 596–609.

Wu, L., Niu, Y., Zhang, H., Zhu, H., Shen, L., 2011. Robust feature point matching based on local feature groups (LFGs) and relative spatial configuration. Journal of Computational Information Systems 7, 3235–3244.

Yarkony, J., Fowlkes, C., Ihler, A., 2010. Covering trees and lower-bounds on quadratic assignment, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 887–894.

Yuan, Y., Pang, Y., Wang, K., Shang, M., 2012. Efficient image matching using weighted voting. Pattern Recognition Letters 33, 471–475.

Zhang, J., Marszałek, M., Lazebnik, S., Schmid, C., 2007. Local features and kernels for classification of texture and object categories: A comprehensive study. International Journal of Computer Vision 73, 213–238.

Zhao, W.L., Ngo, C.W., 2009. Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection. IEEE Transactions on Image Processing 18, 412–423.