Chapter 2
Knowing a Good Feature When You See It: Ground Truth and Methodology to Evaluate Local Features for Recognition

Andrea Vedaldi, Haibin Ling, and Stefano Soatto

Abstract. While the majority of computer vision systems are based on representing images by local features, the design of the latter has been so far mostly empirical. In this Chapter we propose to tie the design of local features to their systematic evaluation on a realistic ground-truthed dataset. We propose a novel mathematical characterisation of the co-variance properties of the features which accounts for deviation from the usual idealised image affine (de)formation model. We propose novel metrics to evaluate the features and we show how these can be used to automatically design improved features.

2.1 Introduction

Local features are the building blocks of many visual recognition systems. They are deterministic functions of the image (i.e., statistics) designed to minimize the effect of various "nuisance factors" such as illumination and viewpoint, while at the same time remaining representative of the object or category at hand.

Local features are typically designed by exploiting common sense, sometime drawing inspiration from current knowledge of the human visual system, without a direct tie to the task at hand. So, we cannot say that any of the existing features is the best possible one could design for the specific task of recognition. And it could not be otherwise. Elementary decision-theoretic considerations reveal that the best

Andrea Vedaldi
University of California at Los Angeles, Los Angeles, USA
e-mail: vedaldi@cs.ucla.edu

Haibin Ling
Temple University, Philadelphia, USA
e-mail: hbling@temple.edu

Stefano Soatto
University of California at Los Angeles, Los Angeles, USA
e-mail: soatto@cs.ucla.edu

R. Cipolla, S. Battiato, & G.M. Farinella (Eds.): Computer Vision, SCI 285, pp. 27–49.
springerlink.com

possible feature is the trivial one – the image itself – as no deterministic function of the data can "create information," even without getting into too much detail on what "information" means in the context of visual recognition.

So why would anyone want to use local features, let alone design or compare them? For one, they seem to work, and it is worthwhile trying to understand why and how.[1] Given that we are not going to design features for optimality in the end-to-end task, can we at least *test their effectiveness?* How do we compare two features? How can we say that one is better than the other?

So far, all comparisons of local features have been *empirical*. That is, their effectiveness is measured by recognition performance in an end-to-end task, where the features are one element of the decision process, together with the classifier and the dataset. An empirical test can tell which one is the better feature among the group being tested, but it tells us nothing on how a given feature can be improved, or how performance generalizes to different classifiers and different data sets.

In this Chapter we introduce a different methodology for evaluating features. We call this *rational evaluation*, as opposed to empirical, even though it naturally entails an experiment.

The first thing we need is *ground truth.* If features were designed for optimality in an end-to-end task (in which case they would have to be co-designed with the classifier), then any labeled training set, along with standard decision-theoretic tools, would suffice. But features are not co-designed with the classifier, so they should be evaluated independently of it. For that we need ground truth. In this Chapter we describe a way to design ground-truthed data to evaluate the effectiveness of a given feature based on its underlying (explicit or implicit) invariance assumptions. Such data consists of *synthetic images*, generated with a model that strictly includes the model underlying the invariance assumptions of a given feature. While ultimately an end-to-end system should be evaluated on the recognition task performed on real images, there is no straightforward way to distill the role of features unless proper ground truth is available.

Once we have ground truth, we need to elucidate the various components of the feature design process, that includes a choice of image domain (the "feature detector"), a choice of image statistic computed on such a domain (the "feature descriptor"), and a choice of decision function ("feature matching") that becomes the elementary tool of the classifier downstream.

The effect of this procedure is not just a number to rank existing features based on how well they perform, when coupled with a given classifier, on a given dataset. A rational comparison also provides ways to improve the design of the feature, as we illustrate with an example. A similar approach could be followed to design better descriptors, and also better detector.

This Chapter is part of a three-prong approach we have been developing for designing and evaluating local features: In [15] we provide a reliable open-source

[1] Even though, theoretically, one could "learn away" nuisances with a super-classifier that would take the raw images as input, such a classifier may be too hard to design, or require too much data to train, especially for adversarial nuisances such as occlusions.

implementation of some of the most common local features. In this manuscript we describe a methodology to compare local features. Finally, in [14] we provide code to generate synthetic test images, as well as a number of already rendered samples.

2.2 Empirical Studies of Local Features

Because of their prominent role in recognition systems, local features have been the subject of considerable attention in the Computer Vision community. Due to the difficulty of extracting adequate ground truth, however, direct evaluation (i.e., not part of an end-to-end system) has been mostly limited to planar scenes [9] designed to fit the conditions for which the features were designed. While local features are usually designed to be invariant to a simple class of transformations (say affine, or projective, corresponding to the assumption of planar scenes), it is the behavior of the feature in the presence of violations of such assumptions that determines its effectiveness. Therefore, it is important that the ground truth reflects conditions that supersede the underlying assumptions.

The need to test features on more challenging data has driven some to employ synthetic datasets [12, 5], although the resulting images lacked in visual realism. More realistic data was used by [2] to infer ground truth via stereo. This procedure, however, is difficult to scale up to be representative of the complexity and variability of natural scenes. The most extensive collection of real objects to-date is [10], where a selection of (uncluttered) objects was placed on a calibrated turntable in front of a blue screen. Thousands of features were mapped from small-baseline image pairs to wide-baseline views in a semi-automated fashion. A semi-synthetic data set was produced in [13] by gathering range images acquired with a laser scanner and generating a number of artificial views by rotating the data. [17] recognized the importance of obtaining wide-baseline feature deformation data for the study of viewpoint-invariant features and used structure from motion to estimate re-projection of point features from a large number of views of real scenes. Unfortunately this technique provides only limited ground truth information (i.e., sparse 3-D points estimated from the images themselves) and is laborious to collect, especially for controlled experimental conditions. To this date, however, there is no extensive data set that can scale up to an arbitrarily large number of scenes, where the geometry of the scene, its reflectance, the illumination, sensor resolution, clutter, and lens artifacts can be controlled and analyzed by the user.

In order to make a useful tool for evaluating features, however, it is not sufficient to generate (even a lot of) synthetic scenes with ground truth. We have to develop a methodology that allows us to evaluate different aspects of the feature matching process in isolation if we want to rationally improve the design of existing features. The following section does just that. While the nomenclature we introduce may seem like a burden to the reader at first, it will make the evaluation process more rigorous and unequivocal.

2.2.1 Some Nomenclature

The image is a function from a domain (the lattice, or the real plane) to a range (the positive reals, possibly quantized into levels, or the three color channels). A *local feature* is a local image statistic. That is, a deterministic function of the image restricted to a neighborhood. A neighborhood is a compact, simply connected subset of the image domain, which is often referred to as a *"region."* A local feature that does not depend on a particular parameter or function is said to be *invariant* to it. A desirable feature is one that is invariant to phenomena that are independent of the identity of the object or category of interest, often called *nuisance factors*. A feature is called *distinctive* if, when considered as a function of the object or category of interest, it is not a constant. A desirable feature is one that provides a "signature" of the object of interest. We focus our attention on the two most common nuisance factors, illumination and viewpoint, and seek for features that are distinctive of the shape and reflectance properties of the object or category of interest. Conceptually, the design of such features can be broken down into steps:

Detection. Given an image, the *co-variant detector*, or simply "detector", selects a number of image regions. It is designed to extract the same (deformed) regions as the image deforms under a viewpoint change. A specific detector (SIFT, Harris-Laplace, Harris-Affine) is compatible by design with a certain family of such deformations (usually a group, e.g., similarities, affinities [9]). Section 2.3.1 develops a formal model of this step.

Canonization. The co-variant regions are *canonized*, i.e., deformed to a standard shape. This process compensates (in part or entirely) for deformations induced by the companion transformations. It is often assumed that such transformations form a group, and therefore they can be undone (inverted).

Description. The *descriptor* computes a statistic of the image on the canonized co-variant regions. This process may eliminate, or render the descriptor insensitive to, additional deformations which are not removed by canonization.

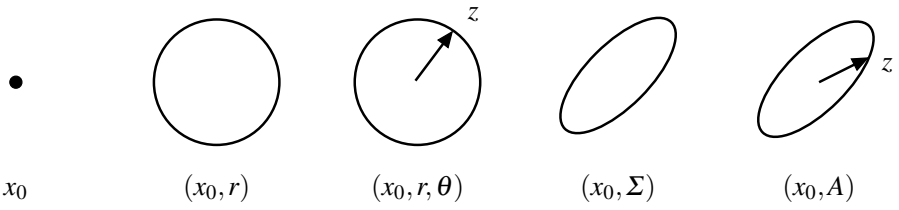
Matching. A *similarity measure* is used to compare invariant descriptors to match regions in different images.

2.3 Constructing a Rigorous Ground Truth

In Section 2.3.1 we introduce an idealized model of the output of co-variant detectors and in Section 2.3.2 a model of feature correspondences. These will be used in the empirical analysis in Section 2.3.3 and 2.3.4.

2.3.1 Modeling the Detector

Viewpoint has a direct effect on the *geometry* of local features, resulting in a deformation of their shape and appearance. The purpose of a *(co-variant) detector* is to select regions that warp according to, and hence track, image deformations induced by viewpoint changes.



frame	companion warps	fixed subset
point	homeomorphisms	translations
disk	similarities	translations and scalings
oriented disk	similarities	similarities
ellipse	affinities	affinities up to residual rotation
oriented ellipse	affinities	affinities

Fig. 2.1 Feature frames. Top. The figure depicts the five classes of feature frames, together with their parameters and the selected point z used to represent orientation. From left to right: point, disk, oriented disk, ellipse, oriented ellipse. Bottom. Association of frame types to companion warps used in this Chapter.

There is a correspondence between the type of regions extracted by a detector and the deformations that it can handle. We distinguish transformations that are (i) compatible with and (ii) fixed by a detector. For instance, a detector that extracts disks is compatible with, say, similarity transformations, but is not compatible with affine transformations, because these in general map disks to other type of regions. Still, this detector does not fix a full similarity transformation, because a disk is rotationally invariant and that degree of freedom remains undetermined. These ideas are clarified and formalized by the next definitions.

Frames. Typically one models the output of a detector as image regions, i.e., as subsets of the image domain [9] . However, many popular detectors produce "attributed regions" instead (for example the SIFT detector [6] produces oriented disks rather than just disks). Since such attributed regions are ultimately used to specify image transformations, in this work we refer to them as "frames." Thus a *frame* is a set $\Omega \subset \mathbb{R}^2$ possibly augmented with a point $z \in \Omega$. For example, a disk is a set $\Omega = \{|x - x_0|_2 < r\}$ and an oriented disk is the combination (Ω, z) of a disk and a point $z \in \Omega$, $z \neq x_0$ representing its orientation[2] (as the line connecting the center x_0 to z). Here we consider the following classes of frames (see Figure 2.1), that capture the output of most detectors found in the literature:

- **Points.** Points are determined by their coordinates x_0.
- **Disks.** Disks are determined by their center x_0 and radius r.

[2] We prefer to use a point z rather than specifying the orientation as a scalar parameter because this representation is easier to work with and can be easily generalized to more complex feature frames.

- **Oriented disks.** Oriented disks are determined by their center x_0, radius r and orientation θ.
- **Ellipses.** Ellipses are determined by their center x_0 and the moment of inertia (covariance) matrix

$$\Sigma = \frac{1}{\int_\Omega dx} \int_\Omega (x - x_0)(x - x_0)^\top dx.$$

  Note that Σ has three free parameters.
- **Oriented ellipses.** Oriented ellipses are determined by the mapping $A \in GL(2)$ which brings the oriented unit circle Ω_c onto the oriented ellipse $\Omega = A\Omega_c$.

Frames fix deformations. Each type of frame (point, disk, oriented disk, etc.) can be used to fix (and undo, by canonization) certain image transformations. In fact, given a pair of frames Ω_1 and Ω_2, the equation $\Omega_2 = w\Omega_1$ determines (partially or entirely) the warp w. Therefore, a frame Ω acts as a reference frame to specify deformations. This fact is captured by the following definitions:

- **Frame deformations.** For what concerns our discussion, an image deformation (warp) w is simply a transformation $\mathbb{R}^2 \to \mathbb{R}^2$ of the image domain, and wI denotes the image $I(w^{-1}(x))$. Such deformations apply to frames as well: Given a frame (Ω, z), the warped frame $w(\Omega, z)$ is the pair $(w\Omega, w(z))$. Note that, if present, the selected point z is moved too; later we will use the shorthand notation $w\Omega$, still meaning that the warp applies to both the set and the selected point z.
- **Closed, complete, and free frames.** Frames are *closed* under the deformations $\mathcal{W}$ if warping a frame by $w \in \mathcal{W}$ does not change their type. For example, disks and oriented disks are closed under similarity transformations and ellipses and oriented ellipses are closed under affine transformations. We say that a frame is *complete* for a certain set of transformation $\mathcal{W}$ if the equation $\Omega_2 = w\Omega_1$ admits at most one solution $w \in \mathcal{W}$. We also say that the frames are *free* on the set $\mathcal{W}$ (as in "free generators") if such an equation has a solution for all possible pairs of frames Ω_1 and Ω_2.

When analyzing a detector, it is important to specify both the type of frames it produces and the class of transformations that are assumed, which we call *companion warps*. Notice in fact that each frame type can be used in connection with different types of transformation, so both choices must be specified. In the rest of the Chapter we focus on the most natural cases, summarized in Figure 2.1. For instance, from the table we read that disks are used in conjunction with similarity transformations (their companion warps), but are expected to fix only a subset of them.[3]

[3] Notice also that frames (i) are closed under the companion warps, (ii) complete for a subset of these, and (iii) free on the complete subset. Property (iii) is not always satisfied by real detector. For instance, maximally stable extremal regions [8] have arbitrary shape Ω, but their companion warps are just affine transformations. This means that the equation $\Omega_1 = w\Omega_2$ may not have a solution.

2.3.2 Modeling Correspondences

In the previous section we have modeled the detector as a mechanism that extracts (co-variant) frames. Operatively, the output of the detector is used to establish frame-to-frame correspondences between multiple images of the same object. For evaluation purposes, it is therefore necessary to extract sets of corresponding frames. This idea is captured by the following definitions.

View sets (multiple views). A *view set* [4] $\mathcal{V}$ is a collection of images $(I_1, \dots, I_n)$ of a scene taken under different viewpoints. Under Lambertian reflection and other assumptions [16], any image $I_j(x)$, $x \in \Lambda$ in a view set is obtained from any other image I_i by a deformation $I_j(x) = I_i(h_{ij}(x)) \doteq (h_{ji} I_i)(x)$. Such a deformation arises from the equation

$$h_{ij}(x) = \pi(R_{ij}\pi_j^{-1}(x) + T_{ij}), \quad x \in \Lambda \tag{2.1}$$

where π is the perspective projection and $\pi_j^{-1}(x)$ is the pre-image of pixel x from viewpoint j and (R_{ij}, T_{ij}) is the camera motion from view j to view i. Also note that, due to occlusions and other visibility artifacts, equations $I_j = h_{ji} I_i$ may have only local validity, but this is sufficient for the analysis of local features.

Co-variant frame sets (correspondences). A *(co-variant) frame set* $\mathcal{F}$ is a selection of frames $(\Omega_1, \dots, \Omega_n)$, one for each image of a view set $\mathcal{V} = (I_1, \dots, I_n)$, that are linked by the same deformations of the view set, i.e.,

$$\Omega_i = h_{ij}\Omega_j$$

where h_{ij} is given by (2.1). It is useful to think of co-variant frames as collections of geometric elements (such as points, regions, bars and so on) that are "attached" to the images and deform accordingly. Co-variant frames define the support of features and, by tracking image deformations, enable canonization.

Frame sets enable canonization. By mapping a co-variant frame Ω_i to a *canonical variant* Ω_0, the equation $\Omega_0 = w_i\Omega_i$ defines a warp w_i which undoes the local image deformation in the sense that the local appearance $w_i I_i$ is constant through the view set $i = 1, \dots, n$. For example, mapping an oriented disk Ω_i to the disk $\Omega_0 = w_i\Omega_i$ of unit radius and orientation $z = (0, 1)$ undoes the effect of a similarity transformation. Doing so for an un-oriented disk does the same up to a residual rotation.

Remark 2.1. Operatively, a detector can attach a frame to the local appearance only if this has enough "structure:" We can associate a disc to a radially symmetric blob, but we cannot (uniquely) associate an oriented disc to it because the image is rotationally symmetric. It should be noted, however, that this is irrelevant to the end of canonization: As long as the most specific frame is attached to each image structure, canonization will make the local appearance constant. For example, we cannot associate an oriented disk to a symmetric blob, but this is irrelevant as the residual rotation does not affect the local appearance by definition.

While so far we have just listed nomenclature, the next section will tie these concepts to the empirical process of evaluating features relative to ground truth.

2.3.3 Ground-Truth Correspondences

The main obstacle to the practical applicability of the concept of co-variant frames given in Section 2.3.1 is that the actual image transformations h_{ij} (2.1) are rarely of the idealized types because world surfaces are seldom flat, so the actual pixel motion $h(x)$ is more complex than a similarity or other simple transformation that we might assume. Furthermore, due to occlusion, folding, visibility and reflectance phenomena, images in a view set are rarely related to one another by simple deformations of their domains.

Therefore, we relax our requirement that the frames represent exactly the image deformations, and look for the best fit. We propose the following operational construction of a ground-truth frame set (i.e., of ground-truth correspondences):

1. We select a *reference view* $I_0 \in \mathcal{V}$ and an initial frame Ω_0 in I_0. Then, given an alternate view $I_i \in \mathcal{V}$, we map the points x of the frame Ω_0 to points $y = h(x)$ of the alternate view. To this end we use the three-dimensional ground truth in order to estimate the actual *motion* of the pixels from (2.1), which does not depend on the local appearance. Note that $h(x)$ is well defined even when some pixels $y = h(x)$ are occluded.
2. We search for the warp $w \in \mathcal{W}$ that best approximates h, for example by solving

$$w = \operatorname*{argmin}_{v \in \mathcal{W}} \int_{\Omega_0} \|h(x) - v(x)\|^2 \, dx. \tag{2.2}$$

   Algorithms that solve efficiently this problem for the transformation classes $\mathcal{W}$ of interest are reported in Appendix 2.5. Notice that one can choose a cost different from (2.2), and we illustrate a different example in (2.3).
3. We map the frame Ω_0 to the frame $\Omega_i = w\Omega_0$ by the estimated warp w.

Occlusions and Foldings

The procedure we have delineated is simple, but can be inadequate if the frame Ω_0 contains an occlusion or a strong depth discontinuity, which induces a highly non-linear or discontinuous motion $h(x)$. In such cases, instead of trying to capture the motion of all pixels simultaneously, one can expect the detector to track only the *dominant motion*, i.e., the motion of the background or the foreground, depending on which one occupies the larger portion of the region, or "patch." To this end, before executing step (2.2) we consider splitting the patch in half. We sort the pixels $x \in \Omega_0$ by depth and we search for a (relative) gap in the sorted values which is bigger than a threshold. If we find it, we restrict equation (2.2) only to the pixels before or after the split, based on majority rule.

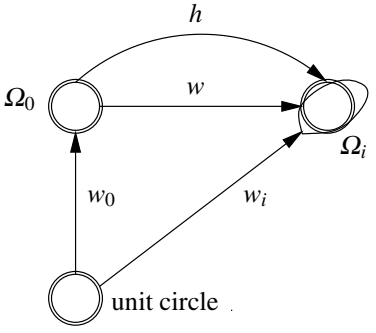


Fig. 2.2 Deviation. The figure illustrates the quality index (2.3). Intuitively, the deviation is the norm of the difference of the true motion h and the estimated motion w, normalized by projecting on the unit disk (canonical configuration). Normalization reflects the fact that features are being canonized before the descriptor is computed.

Quality Indices

The procedure just delineated finds, in each image I_i of a view set, the best matching frame Ω_i. However, not all matches are equal. Some may approximate very well the underlying image transformation, while others may be poor fits, due for instance to occlusions or strong non-linear distortions. For the evaluation of a real-world detector, it is important to assess which of these ground-truth matches are close to the idealized working assumptions, and which are not. To this end, we propose the following quality indices:

Deviation. This index measures the "non-linearity" of the warp. Let $w_0 = (A_0, T_0)$ and $w_i = (A_i, T_i)$ be the affine transformations that map the unit (oriented) circle on the reference frame Ω_0 and the alternate frame Ω_i; let w be the companion warp $\Omega_i = w\Omega_0$ that approximates the true motion $h(x)$. The deviation index is a normalized version of the average square residual $|h(x) - w(x)|^2$, obtained by conjugation with w_i:

$$\mathrm{dev}(w, h, \Omega_i) = \frac{1}{\pi} \int_{\{x:|x|<1\}} |w_i^{-1} \circ (h \circ w^{-1}) \circ w_i(x) - w_i^{-1} \circ w_i(x)|^2 \, dx. \quad (2.3)$$

The formula has a simple interpretation (Figure 2.2). It is the average squared residual $|h(x) - w(x)|^2$ remapped to the canonized version of the frame Ω_i. Noting that, by definition, $w = w_i w_0^{-1}$ and all but h are affine warps, we find (see Appendix 2.5)

$$\mathrm{dev}(w, h, \Omega_i) = \frac{1}{\pi} \int_{\{x:|x|<1\}} |A_i^{-1}(h \circ w_0(x) - w_i(x))|^2 \, dx. \quad (2.4)$$

In practice, we estimate the values of h on the pixels $\widehat{x_i}$ of the region Ω_0; in this case we use the formula

$$\mathrm{dev}(w,h,\Omega_i) \approx \frac{1}{|\Omega_0|} \sum_{\tilde{x}_i \in \Omega_0} |A_i^{-1}(h(\tilde{x}_i) - w(\tilde{x}_i))|^2 \tag{2.5}$$

which preserves its validity even if the region Ω_0 intersects the image boundaries.

Visibility. This is the portion of the frame Ω_i that falls inside the image boundaries.

Occlusion. This is the portion of the region Ω_0 that is occluded in the alternate view I_i. Occluded pixels $x \in \Omega_0$ are determined empirically by checking whether their pre-image from the reference view I_0 and the alternate view I_i correspond, i.e.,

$$R_{i0}\pi_0^{-1}(x) + T_{i0} \neq \pi_i^{-1}(h(x)).$$

Splitting. This is the portion of frame Ω_0 which is accounted for in the estimation of the dominant motion and ranges from 1 (complete frame) to $1/2$ (half frame).

Figure 2.4 illustrates the quality indices for a number of co-variant frames.

2.3.4 Comparing Ground-Truth and Real-World Correspondences

One may regard a real-world detector as a mechanism that attempts to extract co-variant frames from the local appearance only. This task is difficult because, while the definition of correspondences (co-variant frames) is based on the knowledge of the ground-truth transformations h_{ij}, these are not available to the detector, and cannot be estimated by it as this would require operating on multiple images simultaneously [16].

There exist several mechanisms by which detectors are implemented in practice. The simplest one is to randomly extract a large number of feature frames so that eventually some frame sets will be filled "by chance". Albeit very simple, this process poses a high computational load on the matching step. More refined approaches, such as Harris, SIFT, attempt to attach feature frames to specific patterns of the local image appearance (for example SIFT attaches oriented disks to "image blobs"). This enables the detector to explicitly "track" image transformations while avoiding the exhaustive approach of the random detectors. In general, constructing a co-variant detector requires that it be possible to associate co-variant frames to images based on the (local) appearance only. So, for example, we can associate disks to image "blobs," as long as we make sure that, as the blobs move, the disks move according.

No matter what the underlying principle on which a detector is based, the quality of the correspondences established by a real-world detector can be expected to be much lower than the ideal correspondences introduced in Section 2.3.3, which, under the limited expressive power of the regions extracted (e.g., disks are limited to similarity transformations), optimally approximate the actual image

transformations. Thus ground-truth frame sets can be used to compare and evaluate the performance of the real-world detectors.

To asses the performance of a detector, we therefore measure how much the approximate co-variant frame $\tilde{\Omega}_i$ extracted by the detector deviates from the ground truth co-variant frame Ω_i defined in Section 2.3.3. To this end, we use the same criterion introduced in 2.3, and compare the deviation of the ground truth and estimated motions. Consider first the simpler case in which frames are complete for the companion transformations $\mathcal{W}$ (for example oriented disks for similarity transformations). Let $w_i = (A_i, T_i)$ and $\tilde{w}_i = (\tilde{A}_i, \tilde{T}_i)$ be the unique (by hypothesis) warps in $\mathcal{W}$ that bring the oriented unit circle to the frames Ω_i and $\tilde{\Omega}_i$. Let $w = \tilde{w}_i w_i^{-1}$ be the transformation mapping Ω_i to $\tilde{\Omega}_i$; the desired transformation h is the identity $\mathbf{1}$ and by plugging back into eq. (2.3) (see Appendix 2.5) we obtain the *oriented matching deviation*

$$\mathrm{dev}(w, \mathbf{1}, \tilde{\Omega}_i) = \frac{1}{4}\|\tilde{A}_i^{-1} A_i - \mathbf{1}\|_F^2 + |\tilde{A}_i^{-1}(T_i - \tilde{T}_i)|^2 \tag{2.6}$$

where $\|\cdot\|_F$ is the Frobenius matrix norm.

In case the frames are not oriented, w_i and $\tilde{w}_i$ are known only up to right rotations R and $\tilde{R}$ and we have

$$\mathrm{dev}(w, \mathbf{1}, \tilde{\Omega}_i) = \frac{1}{4}\|\tilde{R}^\top \tilde{A}_i^{-1} A_i R - \mathbf{1}\|_F^2 + |\tilde{A}_i^{-1}(T_i - \tilde{T}_i)|^2 \tag{2.7}$$

where we used the fact that the Euclidean norm $|\cdot|$ is rotationally invariant. We obtain the *un-oriented matching deviation* by minimizing over R and $\tilde{R}$ (see Appendix 2.5)

$$\mathrm{dev}(w, \mathbf{1}, \tilde{\Omega}_i) \quad = \quad \frac{1}{4}\left(\|\tilde{A}_i^{-1} A_i\|_F^2 + 2(1 - \mathrm{tr}[\Lambda])\right) \quad + \quad |\tilde{A}_i^{-1}(T_i \quad - \quad \tilde{T}_i)|^2 \tag{2.8}$$

where Λ is the matrix of the singular values of $\tilde{A}_i^{-1}\tilde{A}_i$.

2.3.5 The Data

Based on the concepts that we have introduced in the previous sections, we now describe a new dataset to evaluate and learn visual invariants. The dataset is composed as follows:

View Sets

View sets are obtained from a number of three dimensional scenes shot from different vantage points (Figure 2.3). Each image comes with accurate geometric ground truth information in the form of a *depth map*. This data can be acquired by means of special instrumentation (e.g., a dome and a laser scanner), but in this work we propose to use high quality synthetic images instead. This has the advantages that (a) no special instrumentation is needed; (b) much more accurate ground truth can be generated; (c) automated data extraction procedures can be easily devised. Our

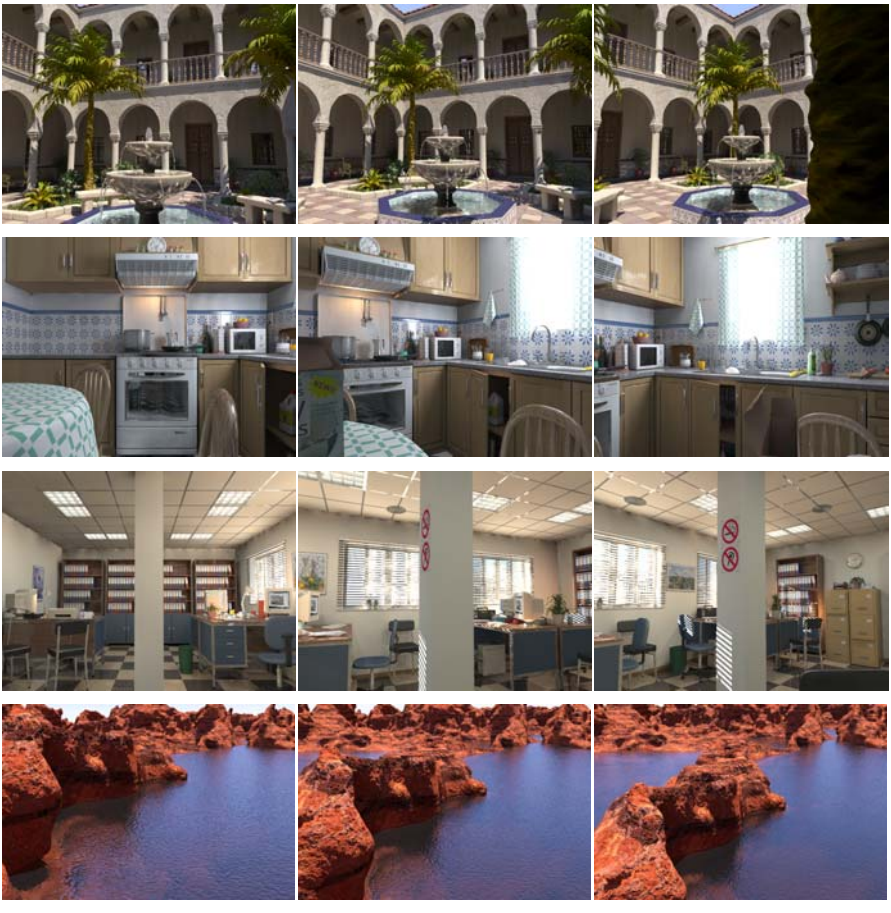


Fig. 2.3 View sets. We show a small portion of a few view sets. These are synthetic rendering of scenes from [11] and come with accurate ground truth. Each image requires several CPU hours to be generated. The data set, which required a large computer cluster to be computed, is available to the public at [14].

data is largely based on publicly available 3-D scenes developed by [11] and generated by means of the freely available POV-Ray ray tracer.[4] Currently we work with a few such scenes that include natural as well as man-made environments; for each scene we compute a large number of views (from 300 to 1000) together with their depth map and camera calibration. The camera is moved to cover a large volume of space (it is more important to sample the camera translations rather than the camera rotations as additional orientations can be simulated exactly in post-processing by simple homographies).

[4] We actually use a customized version to export the required ground truth data. Patches are available from [14].

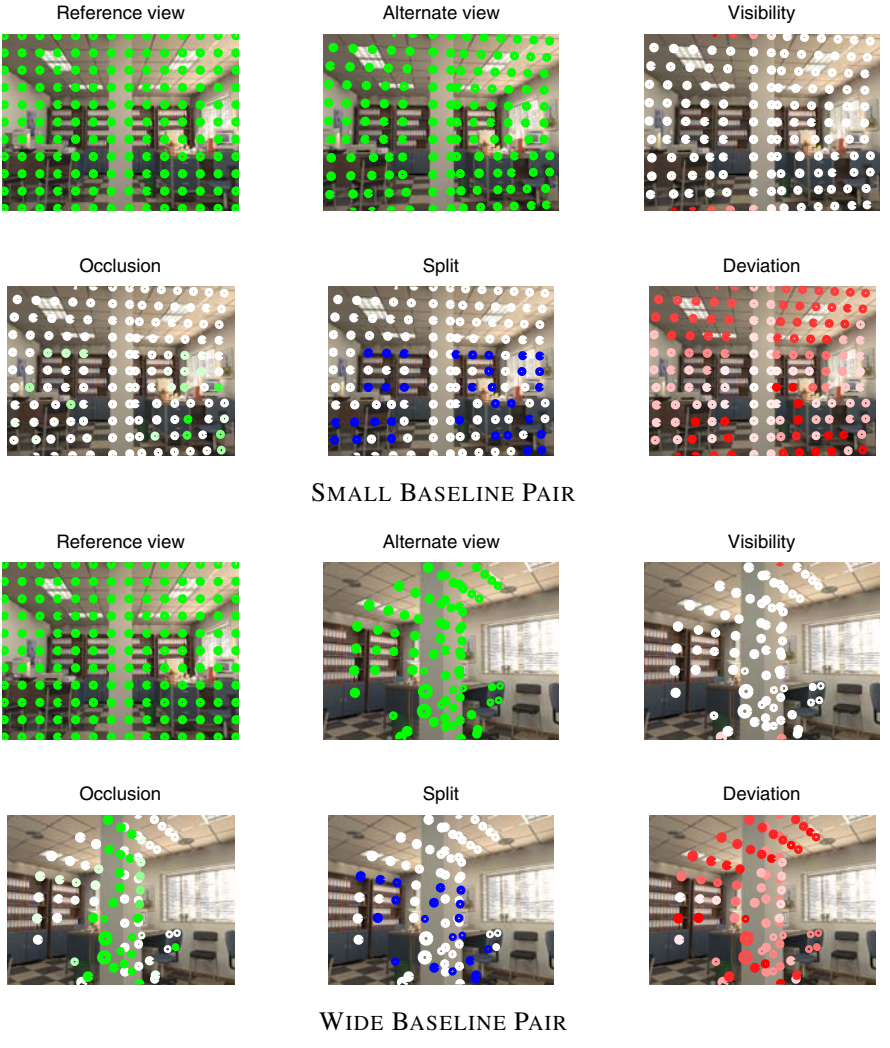


Fig. 2.4 Quality indices. We show examples of the four quality indices (visibility, occlusion, split, and deviation) for a selection of of features in a small baseline pair (top) and wide baseline pair (bottom). Quality indices signal if, and for what reason, a certain match deviates from the idealized working assumptions. Brighter colors indicate higher quality patches.

Frame Sets

For each view set we compute a number of co-variant frame sets (Figure 2.5). This happens as follows:

- We choose a detector (e.g., SIFT) and a number of *reference views* in the view set.
- We run the detector on each reference view to extract reference frames.

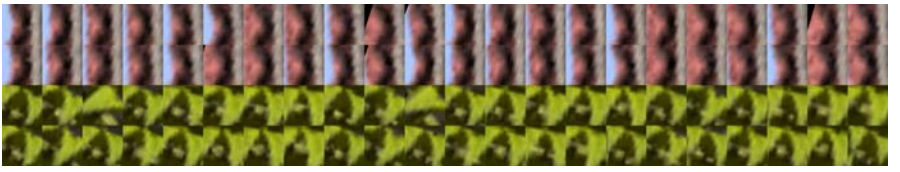


Fig. 2.5 Frame sets (correspondences). We show portion of two ground-truth frame sets (Section 2.3.1) as canonized patches. Each patch is obtained by un-warping to a canonical configuration the corresponding co-variant frame. Note that, due to complex reflectance and geometric phenomena, canonization never yields perfectly aligned patches.

- We re-map each reference frame to all other views as explained in Section 2.3.3 and we compute the four quality indices. The resulting collection of frames is a co-variant frame set. Based on the quality indices, frames can be filtered out in order to generate data of varying difficulty.
- Optionally, we run the detector on each non-reference view as well and we match each co-variant frame to a detected frame by minimizing the quality index introduced in Section 2.3.4. We then record the matching score and matched frames along with the co-variant frame set. This is the approximation of the co-variant frame set obtained from the real-world detector.

In practice, only a few reference views (from 2 to 4) are selected for each view set. This alone is sufficient to generate several thousand frame sets, and most frame sets count dozens of frames from different views. Eventually it is easy to generate data in the order of millions frames. The data comes with quality indices so that interesting subsets can be easily extracted.

2.4 Learning to Compare Invariant Features

The data developed in Section 2.3 can be used to:

1. Learn natural deformation statistics, similarly to [13], but in a wide-baseline setting.
2. Evaluate/learn detectors that compute good approximations of co-variant frames.
3. Evaluate/learn descriptors, given either the co-variant frame sets or the frame sets matched to the output of any practical co-variant detector.
4. Evaluate/learn similarity measures between descriptors.

Here we limit ourselves to the last task for the purpose of illustration of the use of the dataset. While the improvements we expect are limited, since we are only operating on the last ingredient of the feature matching pipeline, the results are readily applicable to existing systems.

More concretely, given a frame Ω_0 in a reference view I_0 and an alternate view I_1, we study two problems: (i) how to find the frame Ω_1 of I_1 that matches Ω_0 (Section 2.4.2) and (ii) when to accept a putative match $\Omega_0 \leftrightarrow \Omega_1$ in order to minimize

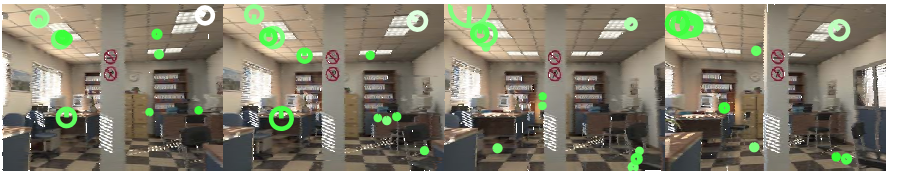


Fig. 2.6 Learning SIFT metric. We show four views of a co-variant frame (the frame on the ceiling lamp) and the ten most similar SIFT features in term of their SIFT descriptors. Shades of green are proportional to the descriptor ϕ_2-similarity to the reference descriptor.

the expected risk of making a mistake (Section 2.4.3). We focus on SIFT features (both detector and descriptor) because of their popularity, but any other similar technique could be studied in this fashion.

2.4.1 Wide Baseline Motion Statistics

[13] studies the statistic of optical flow-based on simulated visual data. However, the analysis is limited to small baseline motion; our data is characterized by a much larger variety of viewing conditions, which enable us to collect statistic on wide-baseline motion.

Here we propose to study the residual of the pixel motion after canonization, i.e., after the companion transformation has been removed, as in (2.2):

$$\forall x : |x| < 1 : \quad r(x) \doteq w_i^{-1} \circ (h \circ w^{-1}) \circ w_i(x) - w_i^{-1} \circ w_i(x)$$

where $w_i(x)$ maps the unit circle to the co-variant frame Ω_0, $h(x)$ is the ground-truth motion and $w(x)$ is the companion transformation.

2.4.2 Learning to Rank Matches

Given a frame Ω_0 of the reference view I_0, its descriptor f_0 and an alternate view I_1, we order the frames $\Omega_1, \Omega_2, \ldots$ of I_1 based on the similarity ϕ of their descriptors $f_1, f_2, \ldots$ to f_0, i.e.,

$$\phi(f_0, f_1) \leq \phi(f_0, f_2) \leq \ldots.$$

Ideally the similarity function ϕ is chosen in such a way that the correct matching frame Ω_i is ranked first.

Normally the similarity of a pair of SIFT descriptors is just their L_1 or L_2 distance, i.e.,

$$\phi_p(f_0, f_1) \doteq \|f_0 - f_1\|_p, \qquad p = 1, 2.$$

Here we show how a similarity ϕ can be learned that outperforms both ϕ_1 and ϕ_2. We do this by setting up a learning problem as follows: Based on our ground-truth data, we sample pairs of corresponding descriptors (f_0, f_1) from a frame set and

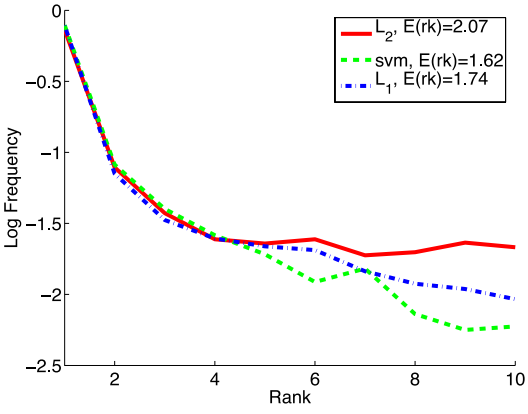


Fig. 2.7 Learn to rank matches. The figure shows the distribution of the rank values of the correct feature match, averaged for frame sets in our database. The SVM-based ranking outperforms the naive ϕ_1 and ϕ_2 ranking, resulting in an expected rank of 1.62, 1.74 and 2.07 respectively.

a pair of non-corresponding descriptors (f_0, f) randomly. We then learn a binary classifier $D(f_0, f_1)$ for the task of deciding whether f_0 and f_1 are the descriptors of corresponding features. Following [3], we assume that the classifier is in the form $[\phi(f_0, f_1) \leq \tau]$ for some function ϕ (for example this is the case for a support vector machine (SVM) but one could use boosting as well [18]) and we use ϕ as a similarity measure.

Re-ranking. Since the goal is to improve ϕ_1 and ϕ_2 (which have already good performance), instead of choosing negative descriptors f completely at random, we select them among the descriptors of the alternate view that have ϕ_p-rank smaller or equal to 10 (Figure 2.6). In testing, the learned similarity ϕ is then used to re-rank these top matches in hope of further improving their ranking. This approach has several benefits: (a) since the computation of ϕ is limited to a few features, testing speed is not a concern; (b) experimentally we verified that the top ten features include very often the correct match; (c) the learning problem has a much more restricted variability because features are ϕ_p-similar by construction.

Learning. We select about 500 frame sets (matched to actual SIFT frames – see Section 2.3.4) and we extract their reference frames Ω_0 and descriptors f_0; for each of them we select about 10 alternate views and we extract the relative co-variant frame Ω_1 and descriptor f_1. In this way, we form about 5,000 positive learning pairs (f_0, f_1). For each positive pair (f_0, f_1), we add about 10 negative pairs (f_0, f) formed as explained for a total of 55,000 examples. The data is used to learn an SVM with polynomial kernel.

Testing. While ϕ is optimized for classification, here we are interested in its ranking performance. Thus testing is done by taking a large number of fresh frame sets and

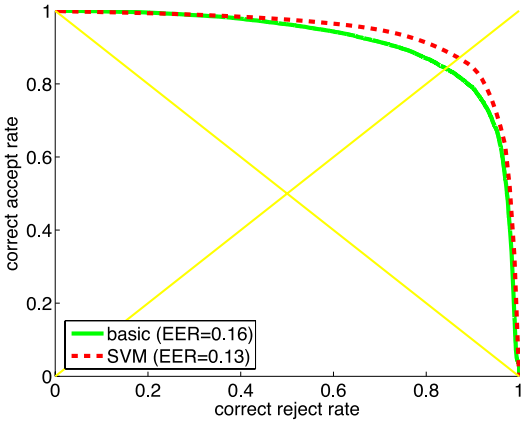


Fig. 2.8 Learning to accept matches. The figure compares the ROC curves of the function $D(f_0, f_1, f_2)$ in its basic form and as learned by an SVM.

averaging the ranking performance of ϕ over them. Figure 2.7 shows that learning can indeed improve the basic similarities.

2.4.3 Learning to Accept Matches

Once putative matches have been proposed, for example based on the similarity metric ϕ, we need to accept or reject them based on some notion of expected risk. In some applications we also want to order matches by reliability [1]. [6] proposes a simple score that can be used to both accept and rank putative matches. With notation similar to the previous section, denote f_0 the descriptor of a reference frame Ω_0 and by f_1 and f_2 the two ϕ-top matches in an alternate view. We define the belief that the match $\Omega_0 \leftrightarrow \Omega_1$ is correct as

$$P(\Omega_0 \leftrightarrow \Omega_1 | f_0, f_1, f_2) = 1 - \frac{\phi(f_1, f_0)}{\phi(f_2, f_0)}.$$

Here we use $\phi = \phi_2$ to be compatible with [6]. This quantity can be directly used to rank and accept matches, the latter by comparing it to a threshold τ, getting the decision function

$$D(f_0, f_1, f_2) = [P(\Omega_0 \leftrightarrow \Omega_1 | f_0, f_1, f_2) \leq \tau]. \tag{2.9}$$

As $D(f_0, f_1, f_2)$ is a decision function, it can also be learned by means of some standard technique, which we do next.

Data and learning. Data is obtained similarly to Section 2.4.2, with the obvious adaptations. Learning is still performed by an SVM based on a polynomial kernel.

Testing. In Figure 2.8 we plot the ROC curve of (2.9) as τ is varied and the ROC curve of the SVM-based decision function $D(f_0, f_1, f_2)$. The equal error rate is lowered from 0.16 to 0.13 showing again that learning can be used to improve the basic method.

2.5 Discussion

We have presented an extensive, flexible, accurate ground-truthed dataset for matching local invariant features. Together with it, we have presented a methodology to evaluate local features, and illustrated their use to not only evaluate, but also improve current algorithms. Our analysis separates the effects of a feature detector, a descriptor, and a matching algorithm, and our dataset is aimed at facilitating the collection of natural image deformation statistics induced by viewpoint changes, and at incorporating them in the design of better features. A similar procedure can be followed to incorporate natural reflectance, illumination and occlusion statistics, which is obviously beyond the scope of this Chapter. We have demonstrated the use of the dataset to improve on the matching score in matching SIFT features. Albeit the quantitative improvement is not stunning, it is sufficient to illustrate the potential advantage associated in the use of the dataset and the associated methodology for evaluating local features.

Appendix 1: Calculations

Justification of Equation (2.5)

By changing variabile in (2.4) we obtain

$$\begin{aligned}\mathrm{dev}(w,h,\Omega_i) &= \frac{1}{\pi \det A_0} \int_{\Omega_0} |A_i^{-1}(h(\tilde{x}) - w(\tilde{x}))|^2 \, d\tilde{x} \\ &\approx \frac{1}{\pi \det A_0} \sum_{\tilde{x}_i \in \Omega_0} |A_i^{-1}(h(\tilde{x}_i) - w(\tilde{x}_i))|^2.\end{aligned}$$

Note that $\pi \det A_0$ is just the area of the region Ω_0, so we obtain (2.5).

Oriented Matching Deviation and Frobenius Norm

Define the "size" of the linear deformation $A \in GL(2)$ the quantity

$$\|A\|^2 = \frac{1}{\pi} \int_{|x|<1} x^\top A^\top A x \, dx.$$

This is the average of the norm of the vector Ax as x is moved along the unit circle. We have

$$\|A\|^2 = \frac{1}{\pi}\mathrm{tr}\left[A^\top A \int_{|x|<1} xx^\top dx\right] = \frac{1}{4}\mathrm{tr}[A^\top A]$$

so this is just the Frobenius norm of A (up to a scale factor). Now consider the affine deformation $Ax+T$. We define analogously

$$\begin{aligned}\pi\|(A,T)\|^2 &= \int_{|x|<1} |Ax+T|^2\,dx \\ &= \int_{|x|<1} x^\top A^\top Ax\,dx + 2\int_{|x|<1} T^\top Ax\,dx + \int_{|x|<1} T^\top T\,dx.\end{aligned}$$

So the "Frobenius norm" of an affine deformation is

$$\boxed{\|(A,T)\|^2 = \frac{1}{\pi}\int_{|x|<1} |Ax+T|^2\,dx = \frac{1}{4}\mathrm{tr}[A^\top A] + |T|^2.}$$

This also justifies (2.6) because

$$\begin{aligned}\mathrm{dev}(w,\mathbf{1},\tilde{\Omega}_i) &= \frac{1}{\pi}\int_{\{x:|x|<1\}} |\tilde{A}_i^{-1}(w_i(x)-\tilde{w}_i(x))|^2\,dx \\ &= \frac{1}{\pi}\int_{\{x:|x|<1\}} |(\tilde{A}_i^{-1}A_i-\mathbf{1})x+\tilde{A}_i^{-1}(T_i-\tilde{T}_i)|^2\,dx.\end{aligned}$$

Unoriented Matching Deviation

Lemma 2.1. *Let A be a square matrix and Q a rotation of the same dimension and let $U\Lambda V^\top = A$ be the SVD of A. Then the rotation Q which minimizes the quantity* $\mathrm{tr}[QA]$ *is $UV^\top$ and the minimum is* $\mathrm{tr}[\Lambda]$.

Proof. Let $V\Lambda U^\top = A$ be the SVD decomposition of matrix A. We have $\mathrm{tr}[QA] = \mathrm{tr}[L\Lambda]$ where Λ is a diagonal matrix with non-negative entries and $L = U^\top QV$ is a rotation matrix. The trace is equal to $\sum_i L_{ii}\lambda_i$ where $0 \le L_{ii} \le 1$ and $L_{ii} = 1$ for all i if, and only if, L is the identity. So the optimal value of Q is $Q = UV^\top$.

Since the Frobenius norm is rotationally invariant, (2.7) can be written as

$$\|\tilde{R}^\top\tilde{A}_i^{-1}A_iR - \mathbf{1}\|_F^2 = \|\tilde{A}_i^{-1}\tilde{A}_i\|_F^2 - 2\,\mathrm{tr}[Q\tilde{A}_i^{-1}A_i] + 2, \qquad Q = R\tilde{R}_i^\top.$$

Minimizing this expression with respect to Q is equivalent to maximizing the term $\mathrm{tr}[Q\tilde{A}_i^{-1}A_iR]$. Let $V\Lambda U^\top = \tilde{A}_i^{-1}\tilde{A}_i$ be the SVD of $\tilde{A}_i^{-1}A_i$; Lemma 2.1 shows that the maximum is $\mathrm{tr}[\Lambda]$ (obtained for $Q = UV^\top$), yielding (2.8).

Appendix 2: Algorithms

In this Section we derive algorithms to minimize (2.2) in the cases of interest. The purpose of the following algorithm is to align a set of points $x_1^{(1)},\ldots,x_1^{(K)}$ to a set of points $x_2^{(1)},\ldots,x_2^{(K)}$ up to either an affine, rigid, or similarity motion.

Alignment by an Affine Motion

Let $x_2 = Ax_1 + T$ for an affine motion (A,T). We can transform this equation as

$$x_2 = Ax_1 + T = Bx = (x^\top \otimes I_{2\times 2}) \operatorname{vec} B, \qquad x = \begin{bmatrix} x_1 \\ 1 \end{bmatrix}, \quad B = \begin{bmatrix} A & T \end{bmatrix}$$

where $\otimes$ is the Kroneker product and vec is the stacking operator. We obtain one of these equations for each of the points $x_1^{(k)}$, $k = 1,\ldots,K$ to be aligned and solve them in the least-squares sense for the unknown B.

Alignment by a Rigid Motion

We give first a closed-form sub-optimal algorithm. This algorithm is the equivalent as the one proposed in [19], but our development is straightforward.

Let $x_2 = Rx_1 + T$ be a rigid motion (R,T) and assume for the moment that the points are three dimensional. Let $R = \exp(\theta\widehat{r})$ where r, $|r| = 1$ is the axis of rotation, $\widehat{r}$ is the hat operator [7], and $\theta > 0$ is the rotation angle. We use Rodrigues' formula [7] $R = I + \sin\theta\widehat{r} + (1-\cos\theta)\widehat{r}^2$ to get

$$\begin{aligned} x_2 &= Rx_1 + T = x_1 + \sin\theta\,\widehat{r}x_1 + (1-\cos\theta)\widehat{r}^2 x_1 + T, \\ x_1 &= R^{-1}(x_2 - T) = x_2 - T - \sin\theta\,\widehat{r}(x_2 - T) + (1-\cos\theta)\widehat{r}^2(x_2 - T). \end{aligned}$$

Adding the previous equations, collecting $\sin\theta\,\widehat{r}$, and using the trigonometric identity $\tan(\theta/2) = (1-\cos\theta)/\sin\theta$ we obtain

$$\sin\theta\,\widehat{r}\left(x_1 - x_2 + T + \tan\frac{\theta}{2}\,\widehat{r}(x_1 + x_2 - T)\right) = 0.$$

It is easy to check that this condition is equivalent to $x_2 = Rx_1 + T$ for $|\theta| < \pi$. A sufficient condition is

$$x_1 - x_2 + T + \tan\frac{\theta}{2}\,\widehat{r}(x_1 + x_2 - T) = 0$$

which can be rewritten as

$$x_1 - x_2 + \tan\frac{\theta}{2}\,\widehat{r}(x_1 + x_2) + z = 0, \qquad z = T - \tan\frac{\theta}{2}\,\widehat{r}T.$$

Since, no matter what r is, z spans all $\mathbb{R}^3$ as T varies, we can equivalently solve this equation linear in the unknowns $\tan(\theta/2)r$ and z in order to estimate the rigid transformation. As in the previous section, one obtains one of such equations for each of the points $x_1^{(k)}$, $k = 1,\ldots,K$ to be aligned and finds the solution in the least-squares sense.

If there is noise in the model, i.e., if $x_2 = Rx_1 + T + n$, we get the condition

$$x_1 - x_2 + \tan\frac{\theta}{2}\,\widehat{r}(x_1 + x_2) + z + \tan\frac{\theta}{2}\,\widehat{r}n = -n.$$

This means that for moderate rotations (away from $\pm\pi$) minimizing the l^2 residual of this equation is almost equivalent to minimizing the norm of n itself. However if θ approaches $\pm\pi$, then the term $\tan(\theta/2)\widehat{r}n$ will dominate, biasing the estimate.

The formulas work for the 2-D case with little adaptation. In this case we assume that all the points lie on the X-Y plane and the rotation vector is aligned to the Z axis, obtaining

$$x_1 - x_2 - \tan\frac{\theta}{2}\begin{bmatrix} x_{2,1} + x_{2,2} \\ -x_{1,1} - x_{1,2} \end{bmatrix} + z = 0.$$

Finally, the estimate can be refined by the iterative algorithm given in the next section (where one fixes the scale s to the constant 1).

Alignment by a Similarity

There is no closed-form algorithm for this case. Instead, we estimate iteratively the translation T and the linear mapping sR. While the solution to the first problem is obvious, for the second consider the following equation:

$$\min_{s,R} \sum_k (x_2^{(k)} - sRx_1^{(k)})^\top (x_2^{(k)} - sRx_1^{(k)})$$
$$= \min_{s,R} \sum_k |x_2^{(k)}|^2 - 2s \sum_k {x_2^{(k)}}^\top Rx_1^{(k)} + s^2 \sum_k |x_1^{(k)}|^2. \quad (2.10)$$

Rewrite the cost function al as $c - 2bs + as^2$. The optimal value for s given a certain R is $s^* = b/a$ and the optimal value of the cost function is $a + c - 2b^2/a$. Note that only the term b is a function of R, while neither a nor c depend on it. As a consequence, the optimal value of R is obtained by solving the problem

$$\max_R b = \max_R \sum_k {x_2^{(k)}}^\top Rx_1^{(k)} = \max_R \sum_k \mathrm{tr}\left(Rx_1^{(k)} {x_2^{(k)}}^\top\right).$$

Thus we are simply maximizing the correlation of the rotated point $Rx_1^{(k)}$ and the target points $x_2^{(k)}$. By taking the derivative of the trace w.r.t. the rotation angle θ, we immediately find that the optimal angle is $\theta^* = \mathrm{atan}(w_2/w_1)$ where

$$w_1 = \sum_k |x_2^{(k)}||x_1^{(k)}| \cos\theta^{(k)}, \qquad w_2 = \sum_k |x_2^{(k)}||x_1^{(k)}| \sin\theta^{(k)}$$

where $\theta^{(k)}$ is the angle from vector $x_1^{(k)}$ to vector $x_2^{(k)}$.

Thus, in order to estimate R and s, we can first solve for the optimal rotation R^*, and then solve for the scale, which is obtained as

$$s^* = \frac{b}{a} = \frac{\sum_k x_2^{(k),\top} R^* x_1^{(k)}}{\sum_k |x_1^{(k)}|^2}.$$

The convergence of the alternating optimization can be greatly improved by removing the mean from $x_1^{(k)}$, $k = 1, \ldots, K$ as a pre-processing step.

References

1. Chum, O., Matas, J.: Matching with PROSAC – progressive sample consensus. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2005)
2. Fraundorfer, F., Bischof, H.: A novel performance evaluation method of local detectors on non-planar scenes. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2005)
3. Hertz, T., Bar-Hillel, A., Weinshall, D.: Learning distance functions for image retrieval. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2004)
4. Lepetit, V., Lagger, P., Fua, P.: Randomized trees for real-time keypoint recognition. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2005)
5. Ling, H., Jacobs, D.W.: Deformation invariant image matching. In: Proceedings of the International Conference on Computer Vision (2005)
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 2(60), 91–110 (2004)
7. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S.: An Invitation to 3-D Vision. Springer, Heidelberg (2003b)
8. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of the Brithish Machine Vision Conference (2002)
9. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. International Journal of Computer Vision 1(60), 63–86 (2004)
10. Moreels, P., Perona, P.: Evaluation of features detectors and descriptors based on 3D objects. In: Proceedings of the International Conference on Computer Vision (2005)
11. Piqueres, J.V.: The persistence of ignorance (2006), `http://www.ignorancia.org/`
12. Rockett, P.I.: Performance assesment of feature detection algorithms: A methodology and case study on corner detectors. Transaction on Image Processing 12(12) (2003)
13. Roth, S., Black, M.J.: On the spatial statistics of optical flow. In: Proceedings of the International Conference on Computer Vision (2005)
14. Vedaldi, A.: A ground-truthed dataset for evaluation and learning of viewpoint invariant features (2008), `http://vision.ucla.edu/gtvpi`
15. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), `http://www.vlfeat.org/`

16. Vedaldi, A., Soatto, S.: Features for recognition: Viewpoint invariance for non-planar scenes. In: Proceedings of the International Conference on Computer Vision (2005)
17. Winder, S.A.J., Brown, M.: Learning local image descriptors. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2007)
18. Zhou, S.K., Shao, J., Georgescu, B., Comaniciu, D.: BoostMotion: Boosting a discriminative similarity function for motion estimation. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2006)
19. Zhuang, H., Sudhakar, R.: Simultaneous rotation and translation fitting of two 3-D point sets. Transaction on Systems, Man, and Cybernetics 27(1) (1997)