CHAPTER 5

Geometric Verification of Matched Features

Once a set of feature matches has been established, the next step is to verify if those matches also occur in a consistent geometric configuration. By this, we mean that the locations and scales of corresponding features in both images are related by a common geometric transformation. The existence of such a transformation can be motivated by the assumption that we observe the same rigid object in both images. If the entire change in the observed feature configurations is caused by a well-defined change of the observing camera (plus some noise), then we should be able to estimate the underlying transformation given enough correspondences. This is examined in detail in Section 5.1. The influences of noise and outliers cause additional problems. Those are dealt with using robust estimation techniques described in Section 5.2.

5.1 ESTIMATING GEOMETRIC MODELS

We seek to express the allowed change between two corresponding feature configurations by an element of the family of linear transformations. Figure 5.1 shows the different levels of linear geometric transformations that are available to us in the image plane. Starting with the square on the far left, the simplest transformation is a pure translation. Adding also a rotation, we arrive at a *Euclidean* transformation, and adding a scale change we get a *similarity* transformation. At this point, the transformation still preserves angles, parallel lines, and distance ratios. This changes when we move to *affine* transformations, which introduce non-uniform scaling and skew and only preserve parallelism and volume ratios. At the far right end, we see the effect of a *projective* transformation, which only preserves intersection and tangency. A projective transformation of a plane onto another plane is called a *homography*.

Depending on the application and the assumptions we can make about the observed scene, different levels of transformations may be suitable. In the following, we therefore derive methods for estimating *similarity transformations*, *affine transformations*, and *homographies*. In all cases, we make the assumption that all feature correspondences are correct (meaning that we do not have to deal with outliers yet), but may be subject to noise.

For the following derivations, it is often convenient to work in *homogeneous coordinates*, which make it possible to express a translation and projection as a single matrix operation. In homogeneous coordinates, each point vector is extended by an additional coordinate w, *e.g.*, $\mathbf{x} = (x, y, w)$. If $w = 0$,

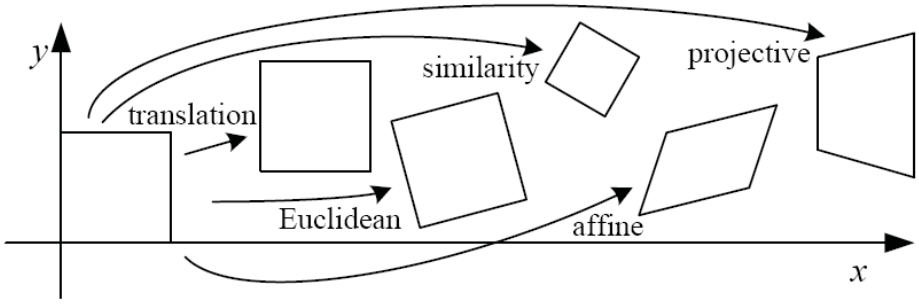

Figure 5.1: The different levels of geometric transformations. Courtesy of Krystian Mikolajczyk.

then the point lies on the *plane at infinity*; else, the point location in ordinary Cartesian coordinates can be obtained by dividing each coordinate by w, *i.e.*, $(x/w, y/w)$.

5.1.1 ESTIMATING SIMILARITY TRANSFORMATIONS

A similarity transformation can already be hypothesized from a single scale- and rotation-invariant interest region observed in both images. Let $f_A = (x_A, y_A, \theta_A, s_A)$ and $f_B = (x_B, y_B, \theta_B, s_B)$ be the two corresponding regions with center coordinates (x, y), rotation θ and scale s. Then we can obtain the transformation from A to B in homogenous coordinates as

$$\mathtt{T}_{\text{sim}} = \begin{bmatrix} ds\cos d\theta & -\sin d\theta & dx \\ \sin d\theta & ds\cos d\theta & dy \\ 0 & 0 & 1 \end{bmatrix}, \text{ where } \begin{array}{rcl} dx & = & x_B - x_A \\ dy & = & y_B - y_A \\ d\theta & = & \theta_B - \theta_A \\ ds & = & s_B/s_A \end{array}. \tag{5.1}$$

If only feature locations are available, we require at least two point correspondences. Then we can compute the two vectors between point pairs in the same image, and we obtain $\mathtt{T}_{\text{sim}}$ as the transformation that projects one such vector onto its corresponding vector in the other image.

5.1.2 ESTIMATING AFFINE TRANSFORMATIONS

Similar to the above, an affine transformation can already be obtained from a single *affine covariant* region correspondence. Recall from Chapter 3 that for estimating the elliptical region shape, we had to compute the region's second-moment matrix $\mathtt{M}$. The transformation that projects the elliptical region onto a circle is given by the square root of this matrix $\mathtt{M}^{1/2}$. We can thus obtain the transformation from region $f_A = (x_A, y_A, \theta_A, \mathtt{M}_A)$ onto region $f_B = (x_B, y_B, \theta_B, \mathtt{M}_B)$ by first projecting f_A onto a circle, rotating it according to $d\theta = \theta_B - \theta_A$, and then projecting the rotated circle back

onto f_B. This leads to the following transformation:

$$\mathtt{T}_{\text{aff}} = \mathtt{M}_B^{-1/2}\mathtt{R}\mathtt{M}_A^{1/2}, \text{ with } \mathtt{R} = \begin{bmatrix} \cos d\theta & -\sin d\theta & 0 \\ \sin d\theta & \cos d\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.2}$$

Alternatively, we can estimate the affine transformation from three or more (non-collinear) point correspondences. If more than three such correspondences are available, we can use all of them in order to counteract the influence of noise and obtain a more accurate transformation estimate. This is done as follows. We start by writing down the affine transformation we want to estimate (in non-homogeneous coordinates). This transformation is given by a 2×2 matrix $\mathtt{M}$ and a translation vector $\mathbf{t}$, such that

$$\begin{aligned} \mathbf{x}_B &= \mathtt{M}\mathbf{x}_A + \mathbf{t} \\ \begin{bmatrix} x_B \\ y_B \end{bmatrix} &= \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_A \\ y_A \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}. \end{aligned} \tag{5.3}$$

We can now collect the unknown parameters into one vector $\mathbf{b} = [m_1, m_2, m_3, m_4, t_1, t_2]^\top$ and write the equation in matrix form for a number of point correspondences $\mathbf{x}_{A_i}$ and $\mathbf{x}_{B_i}$:

$$\begin{aligned} \mathtt{A}\mathbf{b} &= \mathbf{X}_B \\ \begin{bmatrix} & & \cdots & & & \\ x_{A_i} & y_{A_i} & 0 & 0 & 1 & 0 \\ 0 & 0 & x_{A_i} & y_{A_i} & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} &= \begin{bmatrix} \cdots \\ x_{B_i} \\ y_{B_i} \\ \cdots \end{bmatrix}. \end{aligned} \tag{5.4}$$

If we have exactly three point correspondences, $\mathtt{A}$ will be square, and we can obtain the solution from its inverse as $\mathbf{b} = \mathtt{A}^{-1}\mathbf{X}_B$. If more than three correspondences are available, we can solve the equation by building the pseudo-inverse of $\mathtt{A}$:

$$\mathbf{b} = (\mathtt{A}^\top\mathtt{A})^{-1}\mathtt{A}^\top\mathbf{X}_B = \mathtt{A}^\dagger\mathbf{X}_B. \tag{5.5}$$

It can be shown that this solution minimizes the estimation error in the least-squares sense. The results of an affine estimation procedure on a real-world recognition example are shown in Figure 5.2.

5.1.3 HOMOGRAPHY ESTIMATION

A homography, *i.e.*, a projection of a plane onto another plane, can be estimated from at least four point correspondences. When using more than four correspondences, this again has the advantage that we can smooth out noise by searching for a least-squares estimate. Compared to the affine estimation above, the estimation becomes a bit more complicated since we now need to work with

Figure 5.2: Example results of affine transformation estimation for recognition: (top left) model images; (bottom left) test image; (right) estimated affine models and supporting features. From Lowe [1999]. Copyright © 1999 IEEE.

projective geometry. We can do that by using homogeneous coordinates. The homography transformation from a point $\mathbf{x}_A$ to its counterpart $\mathbf{x}_B$ can then be written as follows:

$$\begin{aligned} \mathbf{x}_B &= \tfrac{1}{z'_B}\mathbf{x}'_B \quad \text{with} \quad \mathbf{x}'_B = \mathtt{H}\mathbf{x}_A \\ \begin{bmatrix} x_B \\ y_B \\ 1 \end{bmatrix} &= \tfrac{1}{z'_B}\begin{bmatrix} x'_B \\ y'_B \\ z'_B \end{bmatrix} \quad \begin{bmatrix} x'_B \\ y'_B \\ z'_B \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}\begin{bmatrix} x_A \\ y_A \\ 1 \end{bmatrix} \end{aligned} \tag{5.6}$$

The simplest way to estimate a homography from feature correspondences is the *Direct Linear Transformation* (DLT) method [Hartley and Zisserman, 2004]. Using several algebraic manipulations, this method sets up a similar estimation procedure as above, resulting in the following matrix equation for the homography parameters $\mathbf{h}$:

$$\mathtt{A}\mathbf{h} = \mathbf{0}$$

$$\begin{bmatrix} x_{B_1} & y_{B_1} & 1 & 0 & 0 & 0 & -x_{A_1}x_{B_1} & -x_{A_1}y_{B_1} & -x_{A_1} \\ 0 & 0 & 0 & x_{B_1} & y_{B_1} & 1 & -y_{A_1}x_{B_1} & -y_{A_1}y_{B_1} & -y_{A_1} \\ & & & & & \dots & & & \\ & & & & & \dots & & & \\ & & & & & \dots & & & \end{bmatrix}\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \end{bmatrix}. \tag{5.7}$$

The solution to this equation is the null-space vector of $\mathtt{A}$. This can be obtained by computing the *singular value decomposition* (SVD) of $\mathtt{A}$, where the solution is given by the singular vector

corresponding to the smallest singular value. The SVD of $\mathtt{A}$ results in the following decomposition:

$$\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{V}^\top = \mathtt{U} \begin{bmatrix} d_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_{99} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{19} \\ \vdots & \ddots & \vdots \\ v_{91} & \cdots & v_{99} \end{bmatrix} \tag{5.8}$$

and the solution for $\mathbf{h}$ is given by the last column of $\mathtt{V}^\top$. As above, this solution minimizes the least-squares estimation error.

Since the homography has only 8 degrees of freedom, we are free to bring the result vector into a canonical form by an appropriate normalization. This could be done by normalizing the result vector by its last entry:

$$\mathbf{h} = \frac{[v_{19}, \ldots, v_{99}]}{v_{99}} \,. \tag{5.9}$$

Although this procedure is often used, it is problematic since v_{99} may also be zero. Hartley and Zisserman [2004] therefore recommend to normalize the vector length instead, which avoids this problem:

$$\mathbf{h} = \frac{[v_{19}, \ldots, v_{99}]}{\|[v_{19}, \ldots, v_{99}]\|} \,. \tag{5.10}$$

It should be noted that there are also several more elaborate estimation procedures based on nonlinear optimizations. For those, we however refer the reader to the detailed treatment in [Hartley and Zisserman, 2004].

5.1.4 MORE GENERAL TRANSFORMATIONS

The transformations discussed in this section can also be interpreted in terms of the camera models they afford. *Affine transformations* can only describe the effects of *affine cameras*, a simplified camera model that only allows for orthographic or parallel projection. They are a suitable representation if the effects of perspective distortion are small, such as when the object of interest is far away from the camera and its extent in depth is comparatively small. Affine transformations can also be used to approximate the effects of perspective projection for small regions, such as the local neighborhood of an interest region.

In order to describe the effects of general *perspective cameras*, a *projective transformation* is needed. Section 5.1.3 presented an approach for estimating *homographies*, which capture the perspective projection of a planar surface. In the case of a more general 3D scene, homographies are no longer sufficient, and we need to check if the point correspondences are consistent with an *epipolar geometry*.

If the internal camera calibration is known, the corresponding constraints can be expressed by the so-called *essential matrix*, which captures the rigid 6D transformation (3D translation + 3D rotation) of the camera with respect to a static scene. The essential matrix can be estimated

from 5 correspondence pairs. If the internal camera calibration is unknown, we need to estimate the *fundamental matrix*, which can be estimated from 7 correspondence pairs using a non-linear approach or from 8 correspondence pairs using a linear approach. Although those constraints are routinely used in 3D reconstruction, they are, however, only rarely used for object recognition since their estimation is generally less robust. We therefore do not cover them here and refer to [Hartley and Zisserman, 2004] for details.

5.2 DEALING WITH OUTLIERS

The assumption that all feature correspondences are correct rarely holds in practice. In real-world problems, we often have to deal with a large fraction of *outlier correspondences*, *i.e.*, correspondences that are not explained by the chosen transformation model. These outliers can stem from two sources: they can either be caused by wrong or ambiguous feature matches, or they can be due to correct matches that are just not explained by an overly simplistic transformation model (*e.g.*, if an affine transformation model is used to approximate a projective transformation). In both cases, the net effect is the same, namely that the transformed location of a point from one image projected into the other image differs from its correspondence location in that image by more than a certain tolerance threshold.

The problem with outliers is that they can lead to arbitrarily wrong estimation results in connection with *least-squares estimation*. Imagine a simple estimation problem of finding the best-fitting line given a sample of data points. If we use a least-squares error criterion, then moving a single data point sufficiently far away from the correct line will bias the estimated solution towards this point and may move the estimation result arbitrarily far from the desired solution. The same thing will happen with all transformation methods discussed in Section 5.1, since they are also based on least-squares estimation.

In order to obtain robust estimation results, it is therefore necessary to limit the effect of outliers on the obtained solution. For this, we can use the property that the correct solution will result in consistent transformations for all inlier data points, while any incorrect solution will generally only be supported by a smaller, random subset of data points. The recognition task thus boils down to finding a consistent transformation together with a maximal set of inliers supporting this transformation. In the following, we will present two popular approaches for this task: *RANSAC* and the *Generalized Hough Transform*. Both approaches have been successfully used for real-world estimation problems in the past. We will then briefly compare the two estimation schemes and discuss their relative advantages and disadvantages.

5.2.1 RANSAC

RANSAC or *RAndom SAmple Consensus* [Fischler and Bolles, 1981] has become a popular tool for solving geometric estimation problems in datasets containing outliers. RANSAC is a non-deterministic algorithm that operates in a hypothesize-and-test framework. Thus, it only returns a "good" result with a certain probability, but this probability increases with the number of iterations.

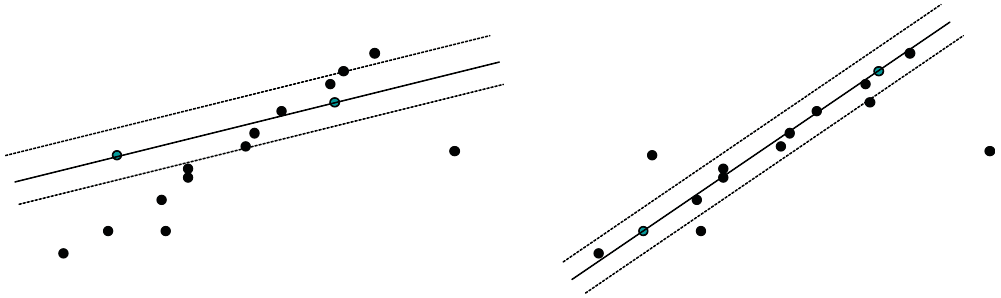

Figure 5.3: Visualization of the RANSAC procedure for a simple problem of fitting lines to a dataset of points in 2D. In each iteration, a minimal set of two points is sampled to define a line, and the number of *inlier points* within a certain distance to this line is taken as its score. In the shown example, the hypothesis on the left has 7 inliers, while the one on the right has 11, making it a better explanation for the observed data. Courtesy of Jinxiang Chai.

Given a set of tentative correspondences, RANSAC randomly samples a minimal subset of m correspondences from this set in order to hypothesize a geometric model (*e.g.*, using any of the techniques described in Section 5.1). This model is then verified against the remaining correspondences, and the number of inliers is determined as its score. This process is iterated until a termination criterion is met. Thus, the RANSAC procedure can be summarized as follows:

1. Sample a minimal subset of m correspondences.
2. Estimate a geometric model $\mathtt{T}$ from these m correspondences.
3. Verify the model $\mathtt{T}$ against all remaining correspondences and calculate the number of inliers I.
4. If $I > I^\star$, store the new model $\mathtt{T}^\star \leftarrow \mathtt{T}$, together with its number of inliers $I^\star \leftarrow I$.
5. Repeat until the termination criterion is met (see below).

The RANSAC procedure is visualized in Figure 5.3 for an example of fitting lines to a set of points in the plane. For this kind of problem, the size of the minimal sample set is $m = 2$, *i.e.*, two points are sufficient to define a line in 2D. In each iteration, we thus sample two points to define a line, and we determine the number of inliers to this model by searching for all points within a certain distance to the line. In the example in Figure 5.3, the hypothesis on the left has 7 inliers, while the one on the right has 11 inliers. Thus, the second hypothesis is a better explanation for the observed data and will replace the first one if chosen in the random sampling procedure.

In the above example, RANSAC is applied to the task of finding lines in 2D. Note, however, that RANSAC is not limited to this task, but it can be applied to arbitrary transformation models, including those derived in Section 5.1. In such a case, we define inliers to be those points whose

Algorithm 1 RANSAC

$k \leftarrow 0, \varepsilon \leftarrow m/N, I^\star \leftarrow 0$
while $\eta = (1 - \varepsilon^m)^k \geq \eta_0$ **do**
  Sample m random correspondences.
  Compute a model T from these samples.
  Compute the number I of inliers for T.
  if $I > I^\star$ **then**
    $I^\star \leftarrow I, \varepsilon \leftarrow I^\star/N$, store T.
  end if
  $k \leftarrow k + 1$.
end while

transformation error (*i.e.*, the distance of the transformed point to its corresponding point in the other image), is below a certain threshold.

It can be seen that the more inliers a certain model has, the more likely it is also to be sampled, since any subset of m of its inliers will give rise to a very similar model hypothesis. More generally, an important role in this estimation is played by the true *inlier ratio* $\varepsilon = I^\star/N$ of the dataset, *i.e.*, by the ratio of the inliers of the correct solution to all available correspondences. If this ratio is known, then it becomes possible to estimate the number of samples that must be drawn until an uncontaminated sample is found with probability $(1 - \eta_0)$. We can thus derive a run-time bound as follows. Let ε be the fraction of inliers as defined above, and let m be the size of the sampling set. Then the probability that a single sample of m points is correct is ε^m, and the probability that no correct sample is found in k RANSAC iterations is given by

$$\eta = (1 - \varepsilon^m)^k \,. \tag{5.11}$$

We therefore need to choose k high enough, such that η is kept below the desired failure rate η_0. As the true inlier ratio is typically unknown, a common strategy is to use the inlier ratio of the best solution found thus far in order to formulate the termination criterion. The resulting procedure is summarized in Algorithm 1.

RANSAC has proven its worth in a large number of practical applications, in many cases, yielding good solutions already in a moderate number of iterations. As a result of the rather coarse quality criterion (the number of inliers in a certain tolerance band), the initial solution returned by RANSAC will, however, only provide a rough alignment of the model. A common strategy is therefore to refine this solution further, *e.g.*, through a standard least-squares minimization that operates only on the inlier set. However, as such a step may change the status of some inlier or outlier points, an iterative procedure with alternating fitting and inlier/outlier classification steps is advisable.

Since RANSAC's introduction by Fischler & Bolles in 1981, various improvements and extensions have been proposed in the literature

Figure 5.4: Visualization of the Generalized Hough Transform (GHT) for object recognition. Each local feature matched between model and test image (shown in yellow) defines a transformation of the entire object reference frame (shown in blue). The GHT lets each such feature pair vote for the parameters of the corresponding transformation and accumulates those votes in a binned voting space. In this example, a 3-dimensional voting space is shown for translation x, y and rotation θ. In practice, scale could be added as a 4^{th} dimension of the voting space. Courtesy of Svetlana Lazebnik, David Lowe, and Lowe [2004], left, and from Leibe, Schindler and Van Gool [2008], right, Copyright © 2008 Springer-Verlag.

(see [*Proc. IEEE Int'l Workshop "25 Years of RANSAC" in conjunction with CVPR*, 2006] for some examples). Among those are extensions to speed up the different RANSAC stages [Capel, 2005, Chum and Matas, 2005, 2008, Matas and Chum, 2004, 2005, Sattler et al., 2009], to deliver run-time guarantees for real-time performance [Nistér, 2003, Raguram et al., 2008], and to improve the quality of the estimated solution [Chum et al., 2004, 2005, Frahm and Pollefeys, 2006, Torr and Zisserman, 2000]. We refer to the rich literature for details.

5.2.2 GENERALIZED HOUGH TRANSFORM

Another robust fitting technique is the *Hough Transform*. The Hough Transform, named after its inventor P.V.C. Hough, was originally introduced in 1962 as an efficient method for finding straight lines in images [Hough, 1962]. Its basic idea is to take the parametric form of a model (*e.g.*, the equation for a line in 2D) and swap the role of the variables and parameters in order to obtain an equivalent representation in the parameter space such that data points lying on the same parametric model are projected onto the same point in parameter space. Ballard [1981] later on showed how this idea could be generalized to detect arbitrary shapes, leading to the *Generalized Hough Transform* (GHT). The basic idea of this extension is that we can let observed single feature correspondences *vote* for the parameters of the transformation that would project the object in the model image to the correct view in the test image.

For this, we use the single-feature estimation approaches described in Sections 5.1.1 and 5.1.2 for scale invariant and affine invariant transformation models. Similar to the above, we subdivide the parameter space into a discrete grid of accumulator cells and enter the vote from each feature correspondence by incrementing the corresponding accumulator cell value. Local maxima in the Hough voting space then correspond to consistent feature configurations and thus to object detection hypotheses. Figure 5.4 visualizes the corresponding GHT procedure for an example of a rotation invariant recognition problem.

As pointed out by Lowe [1999], it is important to avoid all quantization artifacts when performing the GHT. This can be done, *e.g.*, by interpolating the vote contribution into all adjacent cells. Alternatively (or in addition, depending on the level of noise and the granularity of the parameter-space discretization), we can apply Gaussian smoothing on the filled voting space. This becomes all the more important the higher the dimensionality of the voting space gets, as the influence of noise will then spread the votes over a larger number of cells.

5.2.3 DISCUSSION

Comparing RANSAC with the GHT, there is clearly a duality between both approaches. Both try to find a consistent model configuration under a significant fraction of outlier correspondences. The GHT achieves this by starting from a single feature correspondence and casting votes for all model parameters with which this correspondence is consistent. In contrast, RANSAC starts from a minimal subset of correspondences to estimate a model and then counts the number of correspondences that are consistent with this model. Thus, the GHT represents the uncertainty of the estimation in the model parameter space (through the voting space bin size and optional Gaussian smoothing), while RANSAC represents the uncertainty in the image space by setting a bound on the projection error.

The complexity of the GHT is linear in the number of feature correspondences (assuming a single vote is cast for each feature) and in the number of voting space cells. This means that the GHT can be efficiently executed if the size of the voting space is small, but that it can quickly become prohibitive for higher-dimensional data. In practice, a 4D voting space is often considered the upper limit for efficient execution. As a positive point, however, the GHT can handle a larger percentage of outliers with higher dimensionality ($> 95\%$ in some cases), since inconsistent votes are then spread out over a higher-dimensional volume and are thus less likely to create spurious peaks. In addition, the algorithm's runtime is independent of the inlier ratio.

In contrast, RANSAC requires a search through all data points in each iteration in order to find the inliers to the current model hypothesis. Thus, it becomes more expensive for larger datasets and for lower inlier ratios. On the other hand, advantages of RANSAC are that it is a general method suited to a large range of estimation problems, that it is easy to implement, and that it scales better to higher-dimensional models than the GHT. In addition, numerous extensions have been proposed to alleviate RANSAC's shortcomings for a range of problems [*Proc. IEEE Int'l Workshop "25 Years of RANSAC" in conjunction with CVPR*, 2006].

We have now seen the three key steps that state-of-the-art methods use to perform specific object recognition: local feature description, matching, and geometric verification. In the next chapter, we will give examples of specific systems using this general approach.