23. 3-D Vision and Recognition

Kostas Daniilidis, Jan-Olof Eklundh

In this chapter, we describe methods to be applied on a robot equipped with one or more camera sensors. Our goal is to present representations and models for both three-dimensional (3-D) motion and structure estimation as well as recognition. We do not delve into estimation and inference issues since these are extensively treated in other chapters. The same applies to the fusion with other sensors, which we heavily encourage but do not describe here.

In the first part we describe the main methods in 3-D inference from two-dimensional (2-D) images. We are at the point where we could propose a recipe, at least for a small spatial extent. If we are able to track a few visual features in our images, we are able to estimate the self-motion of the robot as well as its pose with respect to any known landmark. Having solutions for minimal case problems, the obvious way here is to apply random sample consensus. If no known 3-D landmark is given then the trajectory of the camera exhibits drift. From the trajectory of the camera, time windows over several frames are selected and a 3-D dense depth map is obtained through solving the stereo problem. Large-scale reconstructions based on camera only do raise challenges with respect to drift and loop closing.

In the second part we deal with recognition as appealed to robotics. The main challenge here is to detect an instance of an object and recognize or categorize it. Since in robotics applications an object of interest always resides in a cluttered environment any algorithm has to be insensitive to missing parts of the object of interest and outliers. The dominant paradigm is based on matching the appearance of pictures. Features are detected and quantized into visual words. Similarity is based on the difference between histograms of such visual words. Recognition has a long way to go but robotics provides the opportunity to explore an object and be active in the recognition process.

With the rapid progress and cost reduction in digital imaging, cameras became the standard and probably the cheapest sensor on a robot. Unlike positioning (global position system, GPS), inertial measurement unit (IMU), and distance sensors (sonar, laser, infrared) cameras produce the highest bandwidth of data. One video camera of very modest resolution yields a bandwidth of 140 Mbits/s (= 30 frames/s × (640 × 480) pixels/frame × 16 bits/pixel). Exploiting information useful for a robot from such a bit stream is less explicit than in the case of GPS or a laser scanner but semantically richer.

Assume for example the scenario that a robot vehicle is given the task of going from place A to place B given as instruction only intermediate visual landmarks and/or

GPS waypoints. The robot starts at A and has to decide where is a drivable path. Such a decision can be accomplished through the detection of obstacles from at least two images by estimating a depth or occupancy map with a *stereo* algorithm. While driving, the robot wants to estimate its trajectory which can be accomplished with a *structure-from-motion* algorithm. The result of the trajectory can be used to build a layout of the environment through *matching* and *triangulation*, which in turn can be used as a reference for a subsequent *pose estimation*. At each time instance the robot has to parse the surrounding environment for risks like pedestrians, or for objects it is searching for like a trash can. It has to become aware of *loop closing* or a reentry if the robot has been kidnaped or blinded for a while. This can be accomplished through *object and scene recognition* yielding the *what* and *where* of objects around the robot. In an extreme scenario, a vehicle can be left to explore a city and build a semantic 3-D map as well as a trajectory of all places it visited, the ultimate *visual simultaneous localization and semantic mapping* problem. In the next section we will deal with 3-D motion and estimation and mapping and in the last section with object recognition.

23.1 3-D Vision and Visual SLAM

In this section, we are going to treat all the above problems assuming that we have only one or two cameras capturing visible light. Though many of those problems can be considerably simplified if we assume the existence of active sensors (for example, based on structured light projection) or if we restrict the configuration space of a robot, we are not going to make use of any such assumptions. Instead we want to increase the understanding of a reasonably general setting and leave to the user the fusion with estimates from other sensors or the reduction of the space of unknowns. The reader is referred to Chaps. 21, 22, and 36 regarding range sensors and to Chap. 20 regarding GPS and inertial sensors. In this chapter we mainly present models, leaving the probabilistic estimation to the chapters about sensor fusion (Chap. 25) and SLAM (Chap. 37). Throughout this section we assume that the correspondence between point features has been solved. The matching problem and the detection of features and their descriptors is treated in the recognition session.

Let us start by introducing the projection of the world to an image plane. Assume that a point in the world (X, Y, Z) has coordinates (X_{ci}, Y_{ci}, Z_{ci}) with respect to the coordinate system of a camera c_i related to each other by the following transformation

$$\begin{pmatrix} X_{ci} \\ Y_{ci} \\ Z_{ci} \end{pmatrix} = \boldsymbol{R}_i \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \boldsymbol{t}_i \,, \tag{23.1}$$

where $\boldsymbol{R}_i$ is a rotation matrix whose columns are the world axes with respect to the camera. The translation vector $\boldsymbol{t}_i$ is starting from the origin of the camera and ending at the origin of the world coordinate system. The rotation matrix is orthogonal $\boldsymbol{R}^\top \boldsymbol{R} = \mathbf{1}$, with determinant 1. We assume that the center of projection is the origin of the coordinate system and that the optical axis is the Z_{ci}-axis of the camera. If we assume that the image plane is the plane $Z_{ci} = 1$ then the image coordinates (x_i, y_i) read

$$x_i = \frac{X_{ci}}{Z_{ci}} \quad y_i = \frac{Y_{ci}}{Z_{ci}} \,. \tag{23.2}$$

In practice, what we measure are the pixel coordinates (u_i, v_i) in the image, which are related to the image coordinates (x_i, y_i) by the affine transformation

$$u_i = f\alpha x_i + \beta y_i + c_u \quad v_i = f y_i + c_v \,, \tag{23.3}$$

where f is the distance of the image plane to the projection center measured in pixels. It is also called the focal length, because they are considered approximately equal. The aspect ratio α is a scaling induced by non-square sensor cells or different sampling rates horizontally and vertically. The skew factor β accounts for a shearing induced by a non-perfectly frontal image plane. The image center c_u, c_v is the point of intersection of the image plane with the optical axis. These five parameters are called intrinsic parameters and the process of recovering them is called intrinsic calibration. Upon recovering them we can talk about a calibrated system and we can work with the image coordinates (x_i, y_i) instead of the pixel coordinates (u_i, v_i). In many vision systems, in particular on mobile robots, wide-angle lenses introduce a radial distortion around the image

center which can be modeled polynomially:

$$x_i^{\text{dist}} = x_i(1 + k_1 r + k_2 r^2 + k_3 r^3 + \dots) ,$$
$$y_i^{\text{dist}} = y_i(1 + k_1 r + k_2 r^2 + k_3 r^3 + \dots) ,$$
$$\text{where } r^2 = x_i^2 + r_i^2 ,$$

where we temporarily assumed that the image center is at (0,0). The image coordinates (x_i, y_i) in (23.3) have to be replaced with the distorted coordinates $(x_{\text{dist}}, y_{\text{dist}})$.

Recovering the intrinsic parameters when we can make multiple views of a reference pattern like a checkerboard without variation of the intrinsic parameters has become a standard procedure using tools like the MATLAB calibration toolbox or *Zhang*'s OpenCV calibration function [23.1]. When intrinsics like the focal length vary during operation and viewing reference patterns is not practically feasible, we rely on the state-of-the-art method by *Pollefeys* et al. [23.2, 3]. When all intrinsics are unknown we can use the Kruppa equations and several stratified self-calibration approaches [23.4, 5] that require at least three views. Apart from radial distortion, the projection relations shown above can be summarized in matrix form. By denoting $\boldsymbol{u}_i = (u_i, v_i, 1)$ and $\boldsymbol{X} = (X, Y, Z, 1)$ we obtain

$$\lambda_i \boldsymbol{u}_i = \boldsymbol{K}_i \, (\boldsymbol{R}_i \;\; \boldsymbol{t}_i) \, \boldsymbol{X} = \boldsymbol{P}\boldsymbol{X} , \tag{23.4}$$

where $\lambda_i = Z_{ci}$ is the depth of point $\boldsymbol{X}$ in camera coordinates and $\boldsymbol{P}$ is the 3×4 projection matrix. The depth λ_i can be eliminated to obtain two equations relating the world to the pixel coordinates.

23.1.1 Pose Estimation Solution

When we have landmarks in the world with known positions $\boldsymbol{X}$, and we can measure their projections, the problem of recovering the unknown rotation and translation in the calibrated case is called pose estimation. Of course, this presumes the identification of the world landmarks in the image. In robotics, pose estimation is rather known as a variant of localization in a known environment. We assume that a camera is calibrated and that measurements of N points are given in world coordinates $\boldsymbol{X}_{j=1,\dots,N}$ and calibrated image coordinates $\boldsymbol{x}_{j=1,\dots,N}$. Let us assume two scene points and denote the known angle between their projections $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ as δ_{12} (Fig. 23.1). Let us denote the squared distance $\|\boldsymbol{X}_i - \boldsymbol{X}_j\|^2$ by d_{ij}^2 and the lengths of $\boldsymbol{X}_j$ by d_j^2. Then the cosine law reads

$$d_1^2 + d_2^2 - 2d_1 d_2 \cos\delta_{12} = d_{12}^2 . \tag{23.5}$$

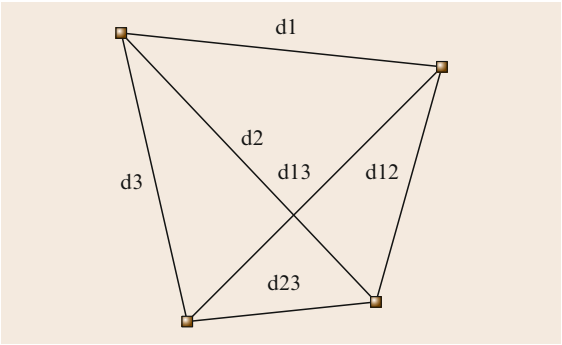

Fig. 23.1 Pose estimation problem: a camera seeing three points at unknown distances d_1, d_2, and d_3 with known angles between the rays and known point distances d_{12}, d_{13}, and d_{23}

If we can recover d_1 and d_2 the rest will be an absolute orientation problem

$$d_j \boldsymbol{x}_j = \boldsymbol{R}\boldsymbol{X}_j + \boldsymbol{t} \tag{23.6}$$

to recover the translation and rotation between the camera and the world coordinate system.

The cosine law has two unknowns d_1 and d_2 so with three points we should be able to solve for the pose estimation problem. Indeed, three points yield a system of three quadratic equations in three unknowns, with a maximum of eight solutions.

We follow here the analysis of the classic solution in [23.6] and set $d_2 = u d_1$ and $d_3 = v d_1$ and solve all three equations for d_1:

$$d_1 = \frac{d_{23}^2}{u^2 + v^2 - 2uv\cos\delta_{23}} ,$$
$$d_1 = \frac{d_{13}^2}{1 + v^2 - 2v\cos\delta_{13}} ,$$
$$d_1 = \frac{d_{12}^2}{u^2 + 1 - 2u\cos\delta_{12}} ,$$

which is equivalent to two quadratic equations in u and v, one we call E3 involving d_{23} and d_{13} and one we call E1, involving d_{13} and d_{12}. Solving E3 for u^2 and substituting in E1 allows E1 to be solved for u without involving radicals. Substituting u back in E3 yields a quartic in v, which can have as many as four real roots. For each v we obtain two roots for u through any of the quadratic equations, yielding a maximum of eight solutions [23.6, 7]. Popular pose estimation algorithms are based either on an iterative method [23.8, 9] or linear versions using auxiliary unknowns of higher dimension [23.10, 11].

23.1.2 Triangulation

When we know both the intrinsics and extrinsics or their summarization in the matrix $\boldsymbol{P}$ and we measure a point we cannot recover its depth from just one camera position. Assuming that we have the projection of the same point $\boldsymbol{X}$ in two cameras

$$\lambda_1 \boldsymbol{u}_1 = \boldsymbol{P}_1 \begin{pmatrix} \boldsymbol{X} \\ 1 \end{pmatrix} ,$$
$$\lambda_2 \boldsymbol{u}_2 = \boldsymbol{P}_2 \begin{pmatrix} \boldsymbol{X} \\ 1 \end{pmatrix} , \quad (23.7)$$

with known projection matrices $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ we can recover the position $\boldsymbol{X}$ in space, a process well known as triangulation. Observe that we can achieve triangulation without decomposing the projection matrices into intrinsic and extrinsic parameters, although we need to remove the distortion in order to write them as above.

Having correspondences of the same point in two cameras with known projection matrices $\boldsymbol{P}_\mathrm{l}$ and $\boldsymbol{P}_\mathrm{r}$ we can solve the two projection equations for the world point $\boldsymbol{X}$. It is worth noting that each point provides two independent equations so that triangulation becomes an overconstrained problem for two views. This is not a contradiction since two rays do not intersect in general in space unless they satisfy the epipolar constraint as presented in the next paragraph. The following matrix in the left-hand side has in general rank 4 unless the epipolar constraint is satisfied, in which case it has rank 3.

$$\begin{pmatrix} x\boldsymbol{P}_\mathrm{l}(3,:) & -\boldsymbol{P}_\mathrm{l}(1,:) \\ y\boldsymbol{P}_\mathrm{l}(3,:) & -\boldsymbol{P}_\mathrm{l}(2,:) \\ x\boldsymbol{P}_\mathrm{r}(3,:) & -\boldsymbol{P}_\mathrm{r}(1,:) \\ y\boldsymbol{P}_\mathrm{r}(3,:) & -\boldsymbol{P}_\mathrm{r}(2,:) \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \boldsymbol{0} , \quad (23.8)$$

where $\boldsymbol{P}(i,:)$ means the i-th row of matrix $\boldsymbol{P}$.

Obviously, the homogeneous system above can be transformed into an inhomogeneous linear system with unknowns (X, Y, Z). Otherwise it can be solved by finding the vector closest to the null-space of the 4×4 matrix above using singular value decomposition (SVD). A thorough treatment of triangulation is the classic [23.12].

23.1.3 Moving Stereo

Imagine now that a rigid stereo system consisting of cameras c_l (left) and c_r (right)

$$\boldsymbol{u}_{\mathrm{l}i} \sim \boldsymbol{P}_\mathrm{l} \boldsymbol{X}_i , \quad (23.9)$$
$$\boldsymbol{u}_{\mathrm{r}i} \sim \boldsymbol{P}_\mathrm{r} \boldsymbol{X}_i , \quad (23.10)$$

is attached to a moving robot and observe this system at two time instances

$$\boldsymbol{X}_0 = \boldsymbol{R}_1 \boldsymbol{X}_1 + \boldsymbol{t}_1 , \quad (23.11)$$

where $\boldsymbol{X}_0$ are point coordinates with respect to the world coordinate system, usually assumed to be with one of the camera instances, and $\boldsymbol{X}_1$ are the coordinates of the same point with respect to the camera rig, after a motion $(\boldsymbol{R}_1, \boldsymbol{t}_1)$. To estimate the motion of the rig, we have to solve two correspondence problems, first between the left and right image, and second between left (or right) at the first time instance and left (or right, respectively) at the second time instance. The left-to-right correspondence enables the solution of the triangulation problem at each time instance. Motion can then be obtained by solving (23.11) for $(\boldsymbol{R}_1, \boldsymbol{t}_1)$, a problem called absolute orientation. Alternatively one can avoid the second triangulation and solve the pose estimation problem between triangulated points in 3D and points in the left image only. In the robotics context, this is the setup most similar to the simultaneous localization and mapping (SLAM) (Chap. 37) problem when range sensors are used, and here we call it binocular SLAM.

Absolute Orientation

The treatment for moving stereo will be short and the reader is referred to a similar treatment in the chapter about range sensing. We assume that correspondences between two time instances have been established based on tracking in the images so that we can formulate equations of the form

$$\boldsymbol{X}_2 = \boldsymbol{R}\boldsymbol{X}_1 + \boldsymbol{t} .$$

The standard way [23.13, 14] to solve this problem is to eliminate the translation by subtracting the centroids, yielding

$$\boldsymbol{X}_2 - \bar{\boldsymbol{X}}_2 = \boldsymbol{R}(\boldsymbol{X}_1 - \bar{\boldsymbol{X}}_1) .$$

We need at least three points in total to obtain at least two non-collinear mean-free $\boldsymbol{X} - \bar{\boldsymbol{X}}$ vectors. If we concatenate the mean free vectors for n points into an $3\times n$ matrix $\boldsymbol{A}_{1,2}$ we can formulate the following minimization of the Frobenius norm

$$\min_{R\in SO(3)} \|\boldsymbol{A}_2 - \boldsymbol{R}\boldsymbol{A}_1\|_F ,$$

which is known as the Procrustes problem. It can be shown [23.14] that the solution is obtained through SVD

as

$$\boldsymbol{R} = \text{sign}(\det(\boldsymbol{U}\boldsymbol{V}^\top))\boldsymbol{U}\boldsymbol{V}^\top \,, \tag{23.12}$$

where $\boldsymbol{U}$, $\boldsymbol{V}$ are obtained from the singular value decomposition

$$\boldsymbol{A}_2\boldsymbol{A}_1^\top = \text{USV}^\top \,.$$

Solutions are usually obtained with RANSAC by sampling triples of points and verification with the Procrustes method.

23.1.4 Structure from Motion (SfM)

Relax now the assumption that the projection matrices are known and focus on measuring and matching the corresponding points $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. This is the well-known structure-from-motion problem or, more precisely, structure and 3-D motion from 2-D motion. In photogrammetry, this is well known as the relative orientation problem. Even after eliminating the λ's from (23.9) or by writing them in projective equivalence form

$$\begin{aligned} \boldsymbol{u}_1 &\sim \boldsymbol{P}_1 \begin{pmatrix} \boldsymbol{X} \\ 1 \end{pmatrix} \,, \\ \boldsymbol{u}_2 &\sim \boldsymbol{P}_2 \begin{pmatrix} \boldsymbol{X} \\ 1 \end{pmatrix} \,, \end{aligned} \tag{23.13}$$

we realize that, if $(\boldsymbol{X}, \boldsymbol{P}_1, \boldsymbol{P}_2)$ is a solution, than $(\boldsymbol{H}\boldsymbol{X}, \boldsymbol{P}_1\boldsymbol{H}^{-1}, \boldsymbol{P}_2\boldsymbol{H}^{-1})$ is also a solution, where $\boldsymbol{H}$ is an invertible 4×4 real matrix or in other words a collineation in $\mathbb{P}^3$. Even if we align the world coordinate system with the coordinate system of the first camera, which is common in practice,

$$\begin{aligned} \boldsymbol{u}_1 &\sim (\boldsymbol{1} \;\; \boldsymbol{0})\, \boldsymbol{X} \,, \\ \boldsymbol{u}_2 &\sim \boldsymbol{P}_2 \boldsymbol{X} \,, \end{aligned} \tag{23.14}$$

we retain the same ambiguity, where $\boldsymbol{H}$ is of the form

$$\boldsymbol{H} \sim \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ h_{41} & h_{42} & h_{43} & h_{44} \end{pmatrix} \,, \tag{23.15}$$

with $h_{44} \neq 0$. This ambiguity is possible when the projection matrices are arbitrary rank-3 real matrices without any constraint on their elements. If we assume that we have calibrated our cameras then the projection matrices depend only on displacements

$$\begin{aligned} \boldsymbol{u}_1 &\sim (\boldsymbol{1} \;\; \boldsymbol{0})\, \boldsymbol{X} \,, \\ \boldsymbol{u}_2 &\sim (\boldsymbol{R} \;\; \boldsymbol{t})\, \boldsymbol{X} \,, \end{aligned} \tag{23.16}$$

and the only remaining ambiguity is the scale ambiguity, where $\boldsymbol{H}$ looks like an identity matrix except with $h_{44} = s \neq 1$ being the scale factor. In other words, if $(\boldsymbol{R}, \boldsymbol{t}, \boldsymbol{X})$ is a solution, then $(\boldsymbol{R}, s\boldsymbol{t}, 1/s\boldsymbol{X})$ is a solution, too. These remarks generalize to multiple views. Because in robotics the $(\boldsymbol{R}, \boldsymbol{t})$ matrices correspond to location and $\boldsymbol{X}$ to mapping of the environment, the problem is more properly described as simultaneous localization and mapping (SLAM). However, because the term SLAM has been used with a variety of sensors, such as sonar and laser range scanners, the term monocular SLAM is better suited to describe structure from motion based on multiple views [23.15].

Epipolar Geometry

This is probably one of the most studied problems in computer vision. We constrain ourselves to the calibrated case, which is most relevant to robotics applications. The necessary and sufficient condition for the two rays $\boldsymbol{R}\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ to intersect is that the two rays are coplanar with the baseline $\boldsymbol{t}$:

$$\boldsymbol{x}_2^\top (\boldsymbol{t} \times \boldsymbol{R}\boldsymbol{x}_1) = 0 \,, \tag{23.17}$$

which is the epipolar constraint (Fig. 23.2). To avoid the scale ambiguity we assume that $\boldsymbol{t}$ is a unit vector. We proceed by summarizing the unknowns into one matrix

$$\boldsymbol{E} = \hat{\boldsymbol{t}}\boldsymbol{R} \,, \tag{23.18}$$

where $\hat{\boldsymbol{t}}$ is the 3×3 skew-symmetric matrix to the vector $\boldsymbol{t}$. The $\boldsymbol{E}$ matrix is called the essential matrix. The epipolar constraint then reads

$$\boldsymbol{x}_2^\top \boldsymbol{E}\boldsymbol{x}_1 = 0 \,, \tag{23.19}$$

which is the equation of a line in the $\boldsymbol{x}_2$ plane with coefficients $\boldsymbol{E}\boldsymbol{x}_1$ or a coefficient of a line in the $\boldsymbol{x}_1$ plane with

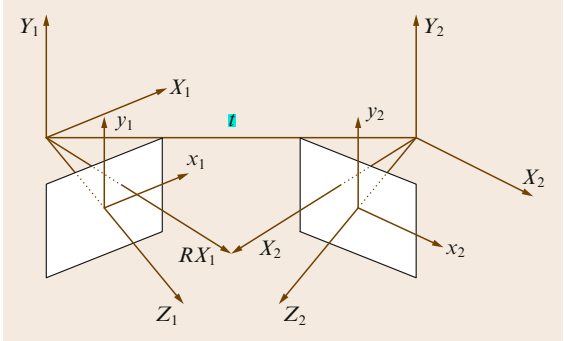

Fig. 23.2 Two views illustrating the coordinate transformations and the perspective projection of the world to a camera

coefficients $\boldsymbol{E}^\top \boldsymbol{x}_2$. These lines are called epipolar and form pencils whose centers are the epipoles $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ in the first and second image plane, respectively. Looking at Fig. 23.2, we can see that the epipoles are the intersections of the baseline with the two image planes, hence $\boldsymbol{e}_2 \sim \boldsymbol{t}$ and $\boldsymbol{e}_1 \sim -\boldsymbol{R}^\top \boldsymbol{t}$. Looking at the equations of the epipolar lines we can immediately infer that $\boldsymbol{E}^\top \boldsymbol{e}_1 = 0$ and $\boldsymbol{E}\boldsymbol{e}_2 = 0$.

The set of all essential matrices

$$\mathcal{E}_2 = \left\{ \boldsymbol{E} \in \mathbb{R}^{3\times3} \mid \boldsymbol{E} = [\boldsymbol{t}]_\times \boldsymbol{R}, \text{ where } \boldsymbol{t} \in \mathrm{S}^2 \text{ and } \boldsymbol{R} \in \mathrm{O}(3) \right\} \tag{23.20}$$

has been characterized as a manifold of degree five. The following result has been proven [23.16].

Proposition 23.1
A matrix $\boldsymbol{E} \in \mathbb{R}^{3\times3}$ is essential if and only if it has two singular values equal to each other and the third singular value equal zero.

We present here a very recent method introduced by *Nister* [23.17] for recovering an essential matrix from five point correspondences, which has gained in popularity because of its suitability for RANSAC methods.

Minimal Case

We expand the epipolar constraint in terms of the homogeneous coordinates $\boldsymbol{x}_1 = (x_1, y_1, z_1)$ and $\boldsymbol{x}_2 = (x_2, y_2, z_2)$ (when the points are not at infinity $z_i = 1$):

$$\begin{pmatrix} x_1\boldsymbol{x}_2^\top & y_1\boldsymbol{x}_2^\top & z_1\boldsymbol{x}_3^\top \end{pmatrix} \boldsymbol{E}_\mathrm{s} = 0\,, \tag{23.21}$$

where $\boldsymbol{E}_\mathrm{s}$ is the raw-by-raw stacked version of matrix $\boldsymbol{E}$.

When we use only five point correspondences the resulting linear homogeneous system will have as a solution any vector in the four-dimensional kernel of the data matrix:

$$\boldsymbol{E}_\mathrm{s} = \lambda_1 \boldsymbol{u}_1 + \lambda_2 \boldsymbol{u}_2 + \lambda_3 \boldsymbol{u}_3 + \lambda_4 \boldsymbol{u}_4\,. \tag{23.22}$$

At this point we want the matrix $\boldsymbol{E}$ resulting from $\boldsymbol{E}_\mathrm{s}$ to be an essential matrix satisfying Proposition 23.1. It has been proven [23.16] that

Proposition 23.2
A matrix $\boldsymbol{E} \in \mathbb{R}^{3\times3}$ is essential if and only if

$$\boldsymbol{E}\boldsymbol{E}^\top \boldsymbol{E} = \frac{1}{2}\mathrm{trace}(\boldsymbol{E}\boldsymbol{E}^\top)\boldsymbol{E}\,. \tag{23.23}$$

Though the $\det(\boldsymbol{E}) = 0$ constraint can be inferred from (23.23) we are still going to use it together with (23.23) to use ten cubic equations in $\boldsymbol{E}$. As described in [23.17], one can obtain a tenth-degree polynomial in λ_4. The number of real roots of this polynomial are computed with a Sturm sequence. There is no proof that a real root will exist at all, beyond the physical plausibility of the existence of at least one solution.

Assuming that we have recovered an essential matrix from point correspondences, the next task is to recover an orthogonal matrix $\boldsymbol{R}$ and a unit vector translation $\boldsymbol{t}$ from the essential matrix. It can be shown that, if $\boldsymbol{E} = \mathrm{USV}^\top$ is the singular value decomposition (SVD) of $\boldsymbol{E}$ with $\det(\boldsymbol{U}) > 0$ and $\det(\boldsymbol{V}) > 0$, then $\boldsymbol{t}$ is parallel to the last column of $\boldsymbol{U}$ and $\boldsymbol{R}$ is $\boldsymbol{U}\boldsymbol{R}_{z\pi}\boldsymbol{V}^\top$ or $\boldsymbol{U}\boldsymbol{R}_{z\pi}^\top\boldsymbol{V}^\top$, where $\boldsymbol{R}_{z\pi}$ is a rotation of $\pi/2$ about the z-axis. Each of the rotations is equal to the other followed by a π-rotation around the baseline. The correct pairing of rotation and translation is chosen so that the reconstructed points are in front of the cameras.

Ambiguities

The approach with five point correspondences has a finite number of feasible (feasible means that they may produce multiple interpretations of structures in front of the camera) solutions when the points in the scene lie on a plane (a twofold ambiguity) [23.18] or when the points on the scene and the camera centers lie on a double sheet hyperboloid with the additional constraint that the camera centers lie symmetrically to the main generator of the hyperboloid [23.19]. These are inherent ambiguities which hold for any number of point correspondences when one seeks a solution for an exact essential matrix.

When solving the linear least-squares system for the essential matrix, a planar scene as well as the case of all points and the camera centers lying on a quadric causes a rank deficiency of the system and thus infinite solutions for $\boldsymbol{E}$.

Beyond the ambiguous situations, there is a considerable amount of literature regarding instabilities in the two-view problem. In particularly, it has been shown [23.18,20,21] that a small field of view and insufficient depth variation can cause an indeterminacy in the estimation of the angle between translation and optical axis. An additional small rotation can cause a confounding effect between translation and rotation [23.22]. Moreover, it has been shown that there exist local minima close to the global minimum that can fool any iterative scheme [23.23, 24].

23.1.5 Monocular SLAM or Multiple-View SfM

When we talk about simultaneous localization and mapping we obviously mean over a longer period of time. The question is how to integrate additional frames into our 3-D motion estimation (localization) process.

To exploit multiple frames we introduce rank constraints [23.25]. We assume that the world coordinate system coincides with the coordinate system of the first frame and that a scene point is projected onto $\boldsymbol{x}_i$ in the i-th frame and that its depth with respect to the first frame is λ_1:

$$\boldsymbol{x}_i = \boldsymbol{R}_i(\lambda_1 \boldsymbol{x}_1) + \boldsymbol{t}_i \; . \tag{23.24}$$

Taking the cross product with $\boldsymbol{x}_i$ and writing it for n frames yields a homogeneous system

$$\begin{pmatrix} \hat{\boldsymbol{x}}_2 \times \boldsymbol{R}_2 \boldsymbol{x}_1 & \hat{\boldsymbol{x}}_2 \boldsymbol{t}_2 \\ \vdots & \vdots \\ \hat{\boldsymbol{x}}_n \times \boldsymbol{R}_n \boldsymbol{x}_1 & \hat{\boldsymbol{x}}_2 \boldsymbol{t}_n \end{pmatrix} \begin{pmatrix} \lambda_1 \\ 1 \end{pmatrix} = 0 \tag{23.25}$$

that has the depth of a point in the first frame as an unknown. The $3n \times 2$ multiple view matrix has to have rank one [23.26], a constraint that infers both the epipolar and the trifocal equations. The least-squares solution for the depth can easily be derived as

$$\lambda_1 = -\frac{\sum_{i=1}^{n} (\boldsymbol{x}_i \times \boldsymbol{t}_i)^\top (\boldsymbol{x}_i \times \boldsymbol{R}_i \boldsymbol{x}_1)}{\| \boldsymbol{x}_i \times \boldsymbol{R}_i \boldsymbol{x}_1 \|^2} \; . \tag{23.26}$$

Given a depth for each point we can solve for motion by rearranging the multiple-views constraint (23.25) as

$$\begin{pmatrix} \lambda_1^1 \boldsymbol{x}_1^{1\top} \otimes \hat{\boldsymbol{x}}_i^1 & \hat{\boldsymbol{x}}_i^1 \\ \vdots & \vdots \\ \lambda_1^n \boldsymbol{x}_1^{n\top} \otimes \hat{\boldsymbol{x}}_i^n & \hat{\boldsymbol{x}}_i^n \end{pmatrix} \begin{pmatrix} \boldsymbol{R}_i^{\text{stacked}} \\ \boldsymbol{t}_i \end{pmatrix} = 0 \tag{23.27}$$

where $\boldsymbol{x}_i^n$ is the n-th image point in the i-th frame and $\boldsymbol{R}_i, \boldsymbol{t}_i$ is the motion from the first to the i-th frame and $\boldsymbol{R}_i^{\text{stacked}}$ is the 12×1 vector of stacked elements of the rotation matrix $\boldsymbol{R}_i$. Suppose that $\boldsymbol{k}$ is the 12×1 kernel (or closest kernel in a least-squares sense) of the $3n \times 12$ matrix in the left hand side obtained through singular value decomposition and let us call $\boldsymbol{A}$ the 3×3 matrix obtained from the first nine elements of $\boldsymbol{k}$ and $\boldsymbol{a}$ the vector of elements 10–12. To obtain a rotation matrix we follow the SVD steps in the solution of absolute orientation (23.12) to find the closest orthogonal matrix to an arbitrary invertible matrix.

On top of such an approach, a bundle adjustment [23.27] minimizes the sum of all deviations between image coordinates and the backprojections of the points to be reconstructed. For N points and M motions, the sum of squares of $2N(M+1)$ residuals has to be minimized with respect to all 3-D coordinates and all motions modulo a universal scale, yielding $3N+6M-1$ unknowns. *Lourakis* [23.28] exploits the sparse structure of the Jacobian involved in any nonlinear minimization. It is worth mentioning that bundle adjustment, though extremely slow, captures the correlation between motion estimates and structure (3-D points) estimates which is artificially hidden in the iterative scheme in (23.25).

The largest-scale motion estimation and registration of views have been performed by *Teller* [23.29] with a decoupled computation first of relative rotations and finally of relative translations. The above multiple-view SfM techniques can only be applied in a sliding-window mode in time due to their batch nature. *Davison* [23.15] showed the first real-time recursive approach by decoupling the direction of the viewing rays from the depth unknowns.

23.1.6 Dense Depth Maps from Stereo

In this section we treat a particular aspect of the 'M' in the visual SLAM, namely how we can obtain a *dense* map of the environment from two simultaneous (stereo) or consecutive images (structure from motion). We emphasized the density of the map to differentiate this task from visual SLAM approaches establishing only layout maps of landmark points.

We start with a simple explanation, namely, how to extract dense depth maps from a pair of cameras with parallel coordinate axes. Given the projection matrices or the essential matrix of two views we can always rotate the two cameras so that corresponding points lie on the same image row, a process called *rectification* [23.30]. Knowledge of the epipolar geometry is sufficient for rectification. We already described how to triangulate given a correspondence so the focus of this section is really solving the matching problem: for each pixel in the image what is the most similar point in the right image, and vice versa. Assume that we have a *similarity function* between two neighborhoods in the left I_l and right I_r images, respectively, whose central pixels differ by a disparity d. Each similarity measure defines a function of (x_l, d) called the *disparity space image* [23.31, 32]. Recent approaches based on the plane-sweeping method [23.33–35] use

a different domain for computing the correlation, which can be the backprojection of the image on frontoparallel planes and using the interdistance of these layers instead of the disparity. Such a backprojection requires the epipolar geometry and contains a different rectification. The choice of the *similarity* or *matching cost* function and the local *aggregation* directly affects the disparity space image and, depending on the time resources, can become quite nonlinear and elaborate. Of particular concern is the aggregation, which implicitly assumes that disparity is constant over a neighborhood. Such an assumption produces several artifacts at discontinuities unless offseted bilateral aggregation is applied [23.36].

Given a disparity space image, algorithms differ from each other in the way they assign a disparity d to each point x_1 based on a local or global *optimization*. In local algorithms, the decision for x_1 is independent of that for other points in the image and the classical procedure is the greedy one, yielding the most similar pixel in the same scanline. Global algorithms solve either an entire image row (scanline) at once (dynamic programming) [23.31, 36, 39] or formulate a cost functional over the entire image that consists of the data term, a regularizing smoothness term, and a discontinuity-preserving step. Labeling all pixels with a disparity label is an NP-hard problem and the two dominant approximation paradigms are graph cuts [23.40, 41] or belief propagation [23.42–44].

The challenges in disparity computation arise from three circumstances: pixels might be occluded in one of the images (occlusions), image variation might be minimal (texturelessness), and the appearance of the same point/area might change due to perspective foreshortening or even due to violation of the Lambertian assumption. The latter effect is rather prominent in specularities and is accentuated when the baseline increases.

Occlusions are handled by local algorithms through bilateral windows and left–right consistency checks for the same correspondences independent of which of the two frames is used as reference. Occlusions are best processed by the dynamic programming approach (illustrated in Fig. 23.3), which we illustrate because of its real-time appeal. Applying dynamic programming in stereo presumes that disparity is a monotone function of the image positions, which means that, if we have x_{11}, d_1 and x_{12}, d_2 (two point-disparity pairs) and $x_{11} < x_{12}$ then $d_1 < d_2$. A further constraint subsumed by all algorithms is the uniqueness constraint, which dictates that a point in the left can have only one corresponding point in

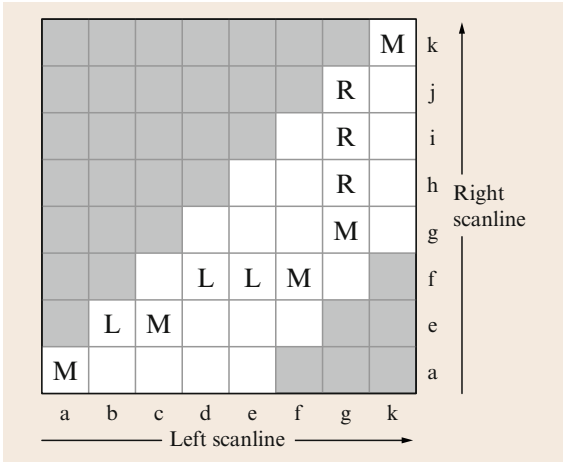

Fig. 23.3 The dynamic programming approach in stereo is based on finding the optimal path in the cost matrix of the picture. Matches are denoted by 'M', while 'L' and 'R' indicate visibility in the left or right image only, respectively [23.37]

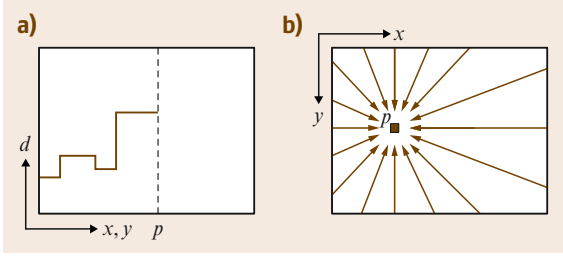

Fig. 23.4 **(a)** Minimum-cost path in disparity space. **(b)** Aggregation of costs in 16 directions [23.38]

the right, and vice versa. Several attempts have been made to overcome its single-scanline optimization character. One of the most successful and ranking highly in the stereo vision benchmark is the semiglobal approach in [23.38, 45] which aggregates the costs of scanlines in many directions (Fig. 23.4).

When multiple concatenated stereo pairs are used, for example, for a given a camera trajectory. We have to combine all depth maps into one 3-D model that is as consistent as possible with all the views. Multiple-view stereo [23.46] takes advantage of the existence of multiple cameras around an object by computing the visual hull of objects and refining depth by applying photoconsistency [23.47]. In the context of robotics, we rarely have such a surround capture but we definitely have multiple views captured by a moving system with estimated motion.

23.2 Recognition

Whatever task a robot performs it must be able to determine where things are in the environment and also identify relevant objects, structures, and events. Vision provides rich information about both *where* and *what*, and visual recognition is therefore an essential capability for robots. However, generic visual object recognition is far from easy. It is not even easy to define, since what constitutes an object in the world is not trivial to define without additional constraints (see, e.g., [23.48]). The problem becomes even more difficult if one talks about classes or categories of objects or, say, places. Visual recognition lies at the heart of computer vision research and a vast number of methods have been proposed to deal with it. Still, much remains unknown about how to realize it.

Fortunately, in many cases in robotics we are not faced with the most general aspects of object recognition or classification. More precisely, the problem is often to find an object that either is known or was recently seen. This is typically the case in tracking, navigation, and manipulation. Other situations concern recognizing landmarks or structures such as roads, or more generally places. These tasks all involve recognition, but mainly in the sense of establishing correspondence between phenomena in different images or between an image and a model. Other cases concern the determination of whether something belongs to a class of objects or establishing whether what is seen is a particular type of object, e.g., a road. This is usually called object classification or categorization and constitutes an even more difficult problem. The most general form of the categorization problem, of course, also includes forming the categories, but that is beyond this treatment.

The literature on computer-based visual object recognition, classification, and categorization is very rich and a multitude of methods have been proposed since the very early days of computer vision. Characteristically certain approaches have waxed and waned in popularity. For instance, the purely statistical pattern recognition methods advocated during the early days, but discounted around 1970–80 when 3-D reconstruction and physical and geometrical modeling were the focus, are again being strongly emphasized, not least due to advances in machine learning and increasingly powerful computers. Surveying all existing techniques would require volumes rather than chapters. In the next few sections we will therefore describe some of the methods that seem particularly useful in the type of robot applications that we have mentioned. The focus on these applications has resulted in many interesting approaches being left out. A comprehensive set of references has been given as entry points to several of these techniques and to the extensive literature on the subject. Moreover, good surveys can be found in the slides from the short course by *Fei Fei* et al. [23.49] or in *Pinz* [23.50], even though they treat categorization rather than recognition.

23.2.1 Approaches to Recognition

Early approaches to object recognition considered objects as represented by 3-D models or by a decomposition into surfaces or volumetric primitives. The first attempts by *Roberts* and *Guzman* [23.51] considered simple geometric parts. *Binford* [23.52] introduced generalized cylinders, which were used by *Brooks* [23.53] in the ACRONYM system and later in the model of *Marr* and *Nishihara* [23.54]. These approaches were based on object-centered representations to enable view invariance. *Marr* [23.55] and others assumed that these could be obtained by 3-D reconstruction from 2-D images using stereo or monocular cues, or by direct acquisition of range images. Several such systems were presented, e.g., *Faugeras* et al. [23.56] and *Bolles* et al. [23.57]. *Biederman* [23.58] introduced his recognition-by-components (RBC) theory as a model for human object recognition from 2-D images. This work inspired numerous attempts to implement RBC-based systems on computers. However, extracting the needed geometric primitives from images turned out to be difficult. The first step of edge detection seldom led to robust indications of where to find the parts. In the ACRONYM system this was addressed by advanced reasoning, in other systems, such as those of *Mohan* and *Nevatia* [23.59] and *Zisserman* et al. [23.60], edges were grouped to generate part hypotheses. Another system using the same principle, but going for the object without any intermediate part representation was, proposed by *Nelson* and *Selinger* [23.61]. Increasingly advanced methods for representing and analyzing shapes based on skeletons have also been proposed. These usually require a silhouette curve and therefore also assume prior segmentation. The influence of these techniques on object recognition has consequently been limited at least insofar that they assume object-centered representations.

The general development has instead been towards viewer-centered representations. In fact, there is an on-

going debate on the role of such representations in human recognition (see, e.g., *Tarr* and *Bülthoff* [23.62]). In computer vision, methods of this sort go back to very early pattern-recognition-based approaches. A revived interest in these first arose in work on face recognition, e.g., *Turk* and *Pentland* [23.63]. This was paralleled by advances on neural network models, e.g., *Poggio* and *Edelman* [23.64]. By showing that training a system on different views of a set of objects resulted in excellent recognition rates *Murase* and *Nayar* [23.65] initiated an interest in learning and statistical approaches that has continued and developed substantially ever since. Notably the focus has changed from global to local methods, which was already suggested by *Rao* and *Ballard* [23.66]. Discriminative as well as generative models have been proposed, which in turn has implied that the methods apply to object categorization as well as recognition. Moreover, more explicit representation of structure and parts have also been incorporated. This introduction of what is now often called constellation-based approaches has in fact decreased the difference between part- and appearance-based methods and again implied a return to very early methods such as the spring-loaded template matching proposed by *Fischler* and *Elschlager* [23.67]. Nevertheless, we will in the sequel describe methods as being appearance-based or constellation- or part-based as a way of stressing different aspects of them.

23.2.2 Appearance-Based Methods

Appearance-based approaches to object recognition rely on the extraction of distinctive features in the images. In pattern recognition and image processing features are defined as N-tuples or vectors whose components are functions of the initial pattern variables or a subset of them (from *Haralick* and *Shapiro* [23.68]). Features can be computed both locally and globally in the images, but in robotics applications local features tend to be most appropriate. In matching one needs to find distinguishing features that characterize the patterns one is looking for. Hence, one separates the steps of detecting features and computing feature descriptors.

As described in the section on *Ambiguities* features computed at discrete locations of the images should preferably be invariant to observer and scene motion. Therefore, geometric features, such as points, lines, and curves have been extensively used, both in computer vision and robotics. These types of features can be computed using derivatives of the image function. Also more general derivative-based features can be derived (see *Lindeberg* [23.69]), and in fact sets of differential invariants that completely represent the image function can be defined. However, simple point features and even extended features such as lines only partly represent the visual information at their locations. Therefore, richer descriptors of these locations are often needed to address, e.g., the correspondence problem. Furthermore, invariance, or at least covariance, is desirable not only for motion or small viewpoint changes, but also for illumination variations and large-scale motions and distance changes. These issues have been considered in recent approaches to extraction of features and feature descriptors (*Lowe* [23.70], *Schmid* and *Mohr* [23.71], *Mikolajczyk* and *Schmid* [23.72], Matas et al. [23.73], and Mikolajczyk et al. [23.74]). Building on decades of development of feature extractors this work today provides a set techniques with rather well-studied performance. The features as well as the feature descriptors are generally computed over small regions around some geometric feature, say a point. In that sense region properties are accounted for, for instance, the scale-invariant feature transformation (SIFT) descriptor by Lowe captures local luminance variations. However, to achieve invariance to illumination the actual luminance values are eliminated in these approaches, and color is not used at all. By including color features additional properties are captured, even though computational models for color constancy, i. e., deriving surface color independently of illumination, have turned out to be very difficult. Color features are still valuable for discrimination. In general, the different features used in the literature are complementary in applicability both with regard to imaging conditions and scene content.

Distinctive Features

The idea of finding interest points for matching was pioneered by *Moravec* [23.75]. His method was furthered in the frequently used detector proposed by *Harris* and *Stephens* [23.76], usually called the Harris detector. It is based on the windowed second moment matrix,

$$\boldsymbol{M} = \boldsymbol{E}\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_x^2 \end{pmatrix} = \boldsymbol{E}\left((\nabla I)(\nabla I)^\top\right) . \qquad (23.28)$$

The eigenvalues of the matrix in (23.28) are proportional to the principal curvatures of the autocorrelation function of I. When both these curvatures are high one has a corner in the image, that is moreover stable with respect to lighting variations. Generally, this detector represents the local distribution of orientations. It was independently introduced as such by *Förstner* [23.77]

and *Bigün* and *Granlund* [23.78]. Typically a Gaussian function is used both for windowing and differentiation, and then the operator takes the form

$$\boldsymbol{M} = \boldsymbol{E}(\cdot, \sigma_I \sigma_D) = \sigma_D^2 G(\sigma_I) \begin{pmatrix} I_x^2(\cdot, \sigma_D) & I_x I_y(\cdot, \sigma_D) \\ I_x I_y(\cdot, \sigma_D) & I_x^2(\cdot, \sigma_D), \end{pmatrix} \quad (23.29)$$

where σ_I and σ_D are the scales of the Gaussian kernels for windowing and differentiation. This matrix is positive semidefinite with eigenvalues $\lambda_1, \lambda_2 \geq 0$. Its trace

$$\boldsymbol{M} = \lambda_1 + \lambda_2 \quad (23.30)$$

represents the strength of the operator response. The direction of the eigenvector corresponding to the largest eigenvalue gives the average gradient direction in the neighborhood. Moreover

$$\det(\boldsymbol{M}) = \lambda_1 \lambda_2 \quad (23.31)$$

provides a measure of the spread of directions around the point. An alternative to the Harris detector is given by the Hessian matrix

$$\boldsymbol{H} = \boldsymbol{H}(\cdot, \sigma_I \sigma_D) = \begin{pmatrix} I_{xx}(\cdot, \sigma_D) & I_{xy}(\cdot, \sigma_D) \\ I_{xy}(\cdot, \sigma_D) & I_{xy}(\cdot, \sigma_D) \end{pmatrix}. \quad (23.32)$$

This operator gives somewhat stronger responses at blobs and ridges (see, e.g., *Lindeberg* [23.69] or *Lowe* [23.79] for details), but of course requires second-order derivatives. The eigenvalues are in this case proportional to the principal curvatures of the greylevel function and can again be computed from the trace and the determinant. *Lowe* [23.70] uses this for a test on the ratio of the eigenvalues to reject keypoints that are on edges. We refer to his paper for details.

Robust Methods for Finding Distinctive Points

The response of these operators depend on and are sensitive to scale changes. To find distinctive points suitable for matching one therefore needs to find the points of maximum response in both space and scale, i. e., search for maxima in a 3-D space. *Lindeberg* [23.80] presents a method for this based on the notion of characteristic scale. *Mikolajczyk* and *Schmid* [23.72] and *Lowe* [23.70, 79] later described efficient ways of implementing this. Mikolajczyk and Schmid first use the Harris detector to localize points in 2D and then select the points at which the Laplacian attains a maximum over scales as the final set. This detector, the Harris–Laplacian, is shown to give the best repeatability over

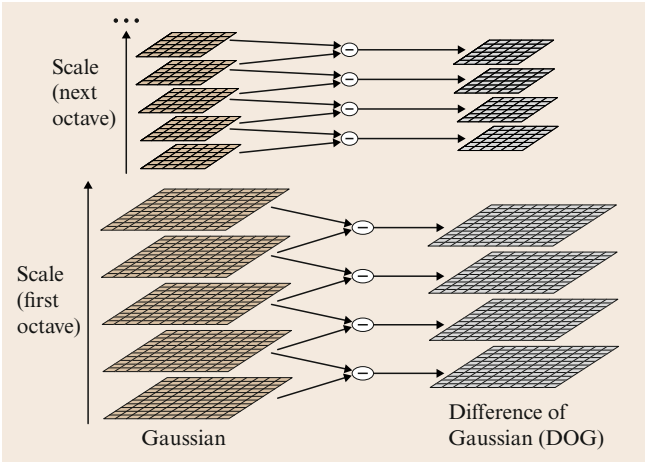

Fig. 23.5 For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian (DOG) images on the right. After each octave, the Gaussian image is downsampled by a factor of 2, and the process is repeated

scale changes among several other common interest point detectors.

Lindeberg showed that one needs to normalize the Laplacian with the factor σ to obtain scale invariance, and indeed *Mikolajczyk* [23.72] demonstrated experimentally that the extrema of $\sigma^2 \nabla^2 G$ produce the over scale most stable image features of several operators. *Lowe* [23.70] presents an efficient approximate method for computing these extrema using D, the difference of Gaussians (DOG) instead. This scale space computation is shown in Fig. 23.5. Introducing the factor k he writes

$$D(\cdot, \delta) = G(\cdot, kt) - G(\cdot, t). \quad (23.33)$$

From the heat diffusion equation it follows that

$$\frac{\partial G}{\partial t} = t \nabla^2 G. \quad (23.34)$$

Hence

$$\partial \nabla^2 G = \frac{\partial G}{\partial t} \approx \frac{D}{kt - t} \quad (23.35)$$

that is

$$D \approx (k-1)\delta^2 \nabla^2 G. \quad (23.36)$$

Lowe's efficient method is based on using the approximation to find the points using a set of DOG images. The result of these operations is a set of keypoints with associated location (x- and y-coordinates),

scale, and orientation. The described operators are not affinely invariant, but can be made so, see e.g. *Mikolajczyk* et al. [23.74]. However, as pointed out by Lowe one can adapt the descriptor so that this becomes unnecessary in many cases, at least if the scene contains enough distinctive features.

Distinctive Regions

Matas et al.[23.73] proposed a method for finding distinguished regions that has turned out to give good results on wide-baseline matching and also in applications to object recognition. It is based on maximally stable extremal regions (MSER), that is connected components of thresholded images. The appropriate way to threshold will become clear below. The regions are extremal in the sense that all pixels inside the MSER have either higher (bright extremal regions) or lower intensity (dark extremal regions) than all the pixels on its boundary. They are stable in the sense that this property is optimized at threshold selection.

The set of extremal regions have a number of desirable properties, see *Matas* et al. [23.73]. For instance, they are unaffected by monotonic or affine intensity changes and to continuous geometric image transformations that preserve topology. Hence, they are stable with respect to many geometric and photometric changes that are common and not always known well enough to be corrected.

To describe the algorithm we follow *Matas* et al. [23.73]. On a digital image I defined on $\mathbb{D}$ with values in $\mathbb{S}$, typically 0–255 regions are defined as connected components under the topology given be 4-neighborhoods. An extremal region $\mathbb{D}$ is a region such that $p \in \mathbb{D}, q \in \partial\mathbb{D} : I(p) > I(q)$ (maximum intensity region) or $I(p) < I(q)$ (minimum intensity region).

To define a maximally stable extremal region we consider a nested set of regions $D_1 \subset \ldots \subset D_i \subset D_{i+1} \subset \ldots$ Region D_i^* is maximally stable if q has a maximum at i, where

$$q(i) = |D_{i+h}| \backslash |D_{i-h}| / |D_i| \qquad (23.37)$$

and $|.|$ denotes cardinality; h is a parameter of the method.

The nested set of regions is obtained by successively thresholding I with a threshold that transcends $\mathbb{S}$, e.g., steps from 0 to 255. The regions obtained from such an operation can be enumerated in a straightforward manner. First the pixels are sorted by intensity. Then the pixels are placed in the image, either in decreasing or increasing order, and the list of connected components and their areas are maintained using, e.g., an efficient union-find algorithm [23.81]. If $\mathbb{S}$ is a small discrete set, such as $0, \ldots, 255$ this algorithm is almost linear in the number of pixels and allows an efficient implementation. Notably, the extremal regions can be messy, but the MSER operation only retains rather simple regions, which is important for subsequent computation of feature descriptors.

The described method finds salient structure without using derivatives. Other techniques of this nature also exist, e.g., the detector by *Kadir* and *Brady* [23.82], the SUSAN corner detection by *Smith* and *Brady* [23.83] and morphological methods.

Descriptors for Distinctive Points and Regions

As mentioned above interest point as well as region detectors provide information about local regions over which suitable descriptors can be computed. Hence, Lowe uses scale information to determine an elliptical region, Matas et al. the convex hull of the MSER, and Mikolajczyk et al. perform an affine normalization to obtain comparable regions in different images.

Numerous region-based descriptors have been proposed for appearance-based object recognition. These have included methods using the local image patch itself [23.84], its statistics (*Schiele* and *Crowley* [23.85]), and more generally filtered versions of it (*Rao* and *Ballard* [23.66]). The success of the second-order moment matrix in motion and shape computation indicates the importance of utilizing the directional statistics. *Lowe* [23.70] suggested a particularly useful descriptor in his scale-invariant feature transformation (SIFT) features.

These descriptors are computed at the distinctive points found, e.g., by the methods described earlier in this section, which determine keypoints with an associated scale. To compute them an orientation is first assigned to the keypoint so as to achieve invariance to image rotation. Notably, this is different from using rotationally invariant measures, as in *Schmid* and *Mohr* [23.71]. *Lowe* [23.70] computes the orientation by finding peaks in the histograms of gradient orientations, with the entries weighted by gradient magnitude. The peaks correspond to dominant directions of local gradients. More than one peak can be accepted if several exist that almost reach the highest value. As a consequence several feature descriptors can be assigned to a single image point. Lowe sets the threshold to 80% of the maximum and keeps up to two peaks. Experiments show that as many as four peaks can be useful.

We refer readers to [23.70] for more details on the computations.

Given these steps we have a set of keypoints with associated locations, scales, and orientations. At each point in a window around such a keypoint we again compute the gradient magnitude and orientation. Weighting the values with a Gaussian we histogram them in $n \times n$ subregions in a $k \times k$ pattern around the keypoint; see Fig. 23.6. *Lowe* [23.70] in his experiments uses $n = k = 4$. In the figure $n = 4$ and $k = 2$. The number of bins for orientation is eight. More details on the computations and the choice of parameters are given in the paper. An important aspect is the choice of measurement region and how it is related to the scale at which the keypoint is detected. This is discussed in the paper, but also in *Matas* et al. [23.73] and in *Mikolajczyk* et al. [23.74]. In the latter paper affine normalization is introduced in the comparison to affine invariant detectors. Lowe discusses the usefulness of that in his article as well.

The notion of considering directional statistics around keypoints has also been used in character recognition, where of course the idea of representing the objects (the characters) as patterns arranged around a central point is natural.

Patches and Local Histograms as Features

An alternative and a complement to using features of the kind described above is to use the image patches directly as local descriptors. Such methods can of course be preceded by a step for determining interest points to limit the amount of computation. *Agarwal* and *Roth* [23.86] proposed a method based on a codebook of patches that later was furthered by *Leibe* et al. [23.84] and several other authors.

The patch preserves local image structure, but it has been shown that also histograms of the information in the patch might be sufficient for object recognition and localization. *Swain* and *Ballard* [23.87] showed examples of recognition based on comparisons of color information. However, they used red–green–blue (RGB) data directly, which turns out to be sensitive to illumination variations. *Schiele* and *Crowley* [23.85] instead used histograms of the output of receptive field computations, either first-order Gaussian derivative operators or differential invariants at three scales. *Schneiderman* and *Kanade* [23.88] showed that efficient recognition of faces and cars could be obtained from histograms of wavelet coefficients. More recently *Linde* and *Lindeberg* [23.89] introduced higher-order histograms and showed that they can be efficiently computed. Generally, these types of methods are simple to use and give good performance provided that sufficient local variations are at hand. A more general framework for such methods has been proposed by *Koenderink* and *van Doorn* [23.90], introducing the notion of locally orderless images. Methods for comparing histograms are well known in statistics. An excellent and concise overview of such techniques is given in *Rubner* and *Tomasi* [23.91].

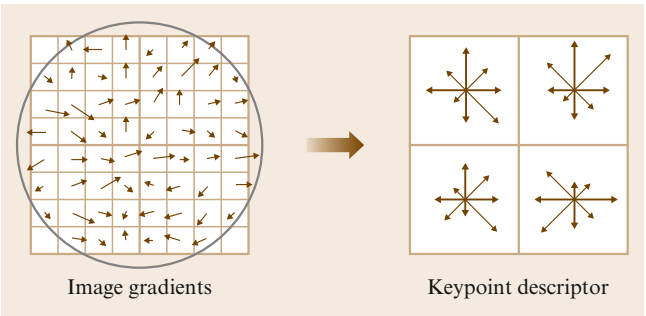

Fig. 23.6 A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the *left*. These are weighted by a Gaussian window, indicated by the *overlaid circle*. The samples are then accumulated into orientation histograms summarizing the contents over 4×4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2×2 descriptor array computed from an 8×8 set of samples, whereas the experiments in this chapter use 4×4 descriptors computed from a 16×16 sample array

23.2.3 Matching

Visual recognition implies matching features or quantities derived from an image to stored representations of objects or images. This is a classical and extensively studied problem that is treated in most standard textbooks on image and signal processing as well as pattern recognition. With new feature-based techniques and with the applications to increasingly large data sets it has attracted considerable interest in recent years and novel contributions have appeared. Some of these will be discussed in the section on constellation based methods. Here we consider some approaches to matching in feature- and appearance-based recognition.

Vocabulary-Based Methods

Straightforward matching using descriptors such as SIFT implies finding nearest neighbors in high-dimensional spaces, which is computationally difficult.

Therefore, Lowe and others use clustering methods to create what can be regarded as an alphabet. Information represented in this way is amenable to general information retrieval methods.

To index an image database *Sivic* and *Zisserman* [23.92] introduced the idea to vector-quantize region descriptors into clusters using k-means and letting these clusters serve as visual *words* in a text retrieval approach. Given a vocabulary generated from a set of training images descriptors are extracted from each new image and assigned to the nearest cluster. In this way matches are immediately obtained for each new image. Sivic and Zisserman applied this to video retrieval, but the method is useful also for other sets of images. Text retrieval is performed using term-frequency inverse document frequency (TF-IDF) relevance scoring (*Baeza-Yates* and *Ribiero-Neto* [23.93]). This technique has been generalized by *Nister* and *Stewenius* [23.94] in the method described next.

Recognition Using Vocabulary Trees

Nister and *Stewenius* [23.94] introduced a hierarchical TF-IDF. In this approach a vocabulary tree is defined by hierarchically defined words. They obtain a very efficient lookup of visual words and can therefore use a larger vocabulary that in turn is shown to give improved retrieval quality (Fig. 23.7).

The vocabulary tree is built up by hierarchical k-means clustering, where k defines a branch factor for the tree. A vector forming a branch of the tree represents a visual word. To compute the score of a new image one needs to determine how similar its descriptors are to the paths down the vocabulary tree. Nister and Stewenius propose assigning a weight w_i to each node i in the tree, for example, based on entropy, and then define the query q and the database vector according to the assigned weights

$$q_i = n_i w_i \,,$$
$$d_i = m_i w_i \,,$$

where n_i and w_i are the number of descriptors in the two images. The relevance score is then set to

$$S(q,d) = \left\| \frac{q}{\|q\|} - \frac{d}{\|d\|} \right\| . \tag{23.38}$$

Nister and Stewenius recommend the L_1-norm, but any norm can be used. Furthermore, they suggest computing the weights as

$$w_i = \ln \frac{N}{N_i} \,,$$

where N is the number of images in the database, and N_i is the number of images in the database with at least one descriptor vector path through node. The proposed method has been shown to give real-time high-quality performance for retrieval from databases of thousands of images.

A related method using vocabulary-based pyramid matching is presented in *Grauman* and *Darrell* [23.95]. They compute an approximate partial matching between two sets of feature vectors. It should be added that these methods mainly address the indexing problem and give the set of most similar images. For localization an additional step for more precise matching and alignment may be required.

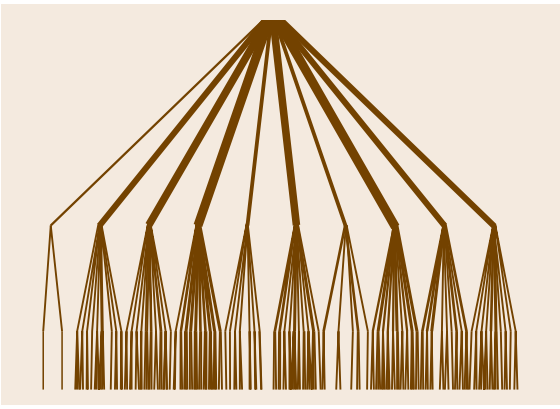

Fig. 23.7 Three levels of a vocabulary tree with branch factor 10 populated to represent an image with 400 features

High-Dimensional Feature Matching

As mentioned above feature matching in high dimensions is computationally difficult. However, useful progress on this problem has also been made. Theoretically it is known that exact identification of the nearest neighbor in high-dimensional spaces in general leads to exhaustive search. Approximative methods have been proposed, e.g., by *Beis* and *Lowe* [23.96] and *Indyk* and *Motwani* [23.97], but they need to trade off speed of computation for quality of approximation. A promising approach to deal with this problem has recently been proposed by *Omercevic* et al. [23.98]. Their method is based on the notion of meaningful nearest neighbors. Such neighbors should be sufficiently close to a query feature that it is an outlier to the background feature distribution. This idea is based on the finding that the tail of the distribution of random outliers can be modeled by

an exponential and that nearest neighbors can therefore be weighted by how much they are similar to the query and dissimilar to the background. Dot products are used to measure the similarity between feature vectors. The authors also introduce a search method based on sparse coding and obtain an approximate method that outperforms other techniques either with respect to speed or accuracy. For instance, while slower than the vocabulary tree method it gave better recognition performance on the same data sets.

23.2.4 Constellation-Based Methods – Recognition by Parts

As mentioned above the success of feature and appearance-based methods together with the difficulty in defining and localizing parts have decreased interest in part-based techniques. However, the obvious advantage of using structural information has led to a renewed focus on the problem and in this subsection we will discuss some recent advances in this area. We will consider methods that use structural relations between the parts, but do not assume that the parts necessarily correspond to geometric primitives of some sort or even geometric parts at all. In fact, it turns out that in this way the distinction between part- and appearance-based becomes rather fuzzy, at least in cases when appearance refers to local parts of the object and not the entire object. Forsyth and Ponce in their book unify such methods by talking about them as being based on structural relations between templates, where the templates can be geometric and defined by 3-D or 2-D shape, or by visual appearance, e.g., given by intensity, color, or texture.

A structural model of an object can be given by a collection of parts together with a representation of how these are connected. Connections between pairs of parts are expressed by an undirected graph $G(V, E)$, where the vertices $v = \{v_1, \ldots, v_n\}$ correspond to the n parts and the edge $(v_i, v_j) \in E$ implies that parts v_i and v_j are connected. An instance of an object is given by a configuration $L = (l_1, \ldots, l_n)$, l_i, being the location of part v_i. There are many ways of parameterizing the locations, including simple image positions or, e.g., positions and joining angles in 2D or 3D for the parts. The idea of using parts in this way is appealing in many ways. Parts provide a modular representation and they may be shared by many objects. Since parts can be simple they are less variable than the entire objects and also often less sensitive to pose variations. Furthermore, occlusion, clutter, and lighting variations are unlikely to influence the recognition of all the parts.

However, it is important to cope with two problems. One has been discussed already, namely that of detecting and localizing the parts. The second one concerns matching, which is generally computationally difficult. *Fischler* and *Elschlager* [23.67] proposed to address it as an energy-minimization problem in the image domain. Such an approach is described by *Felzensschwalb* and *Huttenlocher* [23.99] as follows.

The cost or energy of a given configuration depends on how well a part matches the image data and on how well the parts agree with the model. Given an image, let $m_i(l_i)$ be a measure of the degree of mismatch when part v_i is placed at location l_i in the image. For a pair of connected parts let $d_{ij}(l_i, l_j)$ measure the degree of deformation of the model when part v_i has location l_i and part v_j location l_j. An optimal match of the model to the image can then be defined as

$$L^* = \arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j) \right) . \tag{23.39}$$

This expression gives a configuration that minimizes the sum of match costs for each part and the deformation costs d_{ij} for connected pairs of parts. If the deformation costs only depend on the relative positions of the parts in the pairs the model is invariant to common global transformations.

A problem with this type of formulation is that it generally lacks efficient solutions. If the objects are complicated and especially if the amount of nonrigidity is high the number of parameters required is often also high. Modern approaches to these issues use probabilistic models. Hence, they rely on probabilistic estimates of where parts are rather than on deterministic segmentation steps. The estimates are obtained using learned representations of the appearance of the parts and priors on the configurations. This allows statistical formulations that include both the part finding and configuration problem. However, to initiate such algorithms one still has to find good candidate hypotheses with methods of the type described in the next section.

Let Θ be a set of parameters defining the object and L be the configuration, i. e., the location of each part. Then $P(I \mid L, \Theta)$ is the distribution that measures the likelihood of image I given a viewed object. From Bayes' rule follows that the posterior distribution of the

object configuration L given the model Θ and the image I is

$$P(L \mid I, \Theta) \propto P(I \mid L, \Theta) P(L \mid \Theta) , \qquad (23.40)$$

where $P(L \mid \Theta)$ is the prior probability that an object is at a particular location.

Following *Felzenschwalb* and *Huttenlocher* [23.99], the matching problem formulated as an energy-minimization problem can in this framework be regarded as an MAP estimation problem. They propose an efficient algorithm for this. In their framework the model parameters are $\Theta = (u, E, c)$, where $u = \{u_1, \ldots, u_r\}$ are appearance parameters, E is the set of edges telling what parts are connected (in a graph representation), and $c = \{c_{ij} \mid (v_i, v_j) \in E\}$ are connection parameters. The model parameters are learned from a set of training images using maximum-likelihood estimation. Energy minimization in this formulation is generally NP-hard, but Felzenschwalb and Huttenlocher use the fact that the graph describing the structure has a restricted form to present an efficient algorithm. In other work *Fergus* et al. [23.100] parameterize with respect to appearance, location, and scale, not including any explicit graph representation of the configuration. Their method is aimed at recognizing classes of objects rather than single exemplars.

23.2.5 Place Recognition and Terrain Classification

Localization is a fundamental problem in mobile robotics. Two central aspects of this problem concern continuous pose maintenance and global localization, sometimes called the *robot kidnapping* problem. These problems are generally treated in the context of SLAM. Vision-based techniques form but one class of approaches and the way they are applied largely depends on the type of additional information that is available. In any case, landmark detection and recognition as well as global localization can be addressed using vision and in both cases the problems then share many aspects with object recognition and image retrieval. Hence, several of the techniques presented in the previous sections also apply in these cases.

Scale-invariant key points and SIFT features have been used by *Kosecka* et al. [23.101]. They perform global localization indoors by recognizing locations and exploit information from neighborhood relations from a map using HMMs. *Wolf* et al. [23.102] propose a similar approach. Others have used histogram descriptors to represent the scenes. *Ulrich* and *Nourbakhsh* [23.103] compute color histograms from omnidirectional camera images and match them to stored images in combination with predictions from a topological map. In that way near-real-time performance is obtained through a simple voting process over the color bands. The method is successfully applied to indoor as well as outdoor environments. *Davidson* and *Murray* [23.104] used actively controlled cameras to find landmarks indoors following *Bajcsy*'s active perception paradigm [23.105].

Learning is a central problem in place recognition. In most cases many training images are needed to obtain robust recognition. Moreover, additional information in terms of maps or approximate position is used to facilitate the recognition stage. *Ramos* et al. [23.106] propose a Bayesian approach to deal with these problems. Hence, their method can learn from few (3–10) training images and requires no map. Images are divided into patches and the world is interpreted as a set of places, each having a probabilistic representation. Matching is performed in near real time. They first perform dimensionality reduction on a patch representation of the scenes and then derive a generative probabilistic model of the output in terms of a set of linear mixture models through expectation maximization. The result of this inference process is a mixture of Gaussians that is used in a multiclass classification scheme in which the log likelihood for the model that best explains the given set of patches is selected.

23.3 Conclusion and Further Reading

As main additional sources of reading, we recommend the books by *Hartley* and *Zisserman* [23.5], *Ma* et al. [23.26], *Faugeras* [23.107], and *Faugeras* and *Luong* [23.4]. There is no textbook updated with the most recent results in recognition but the reader is encouraged to browse through the ICCV'05 and CVPR'07 tutorials by Fei-Fei Li, Antonio Torralba, and Rob Fergus. The most representative and real-time implemented system to recognize instances of objects is *Nister*'s [23.94] vocabulary tree approach.

Fig. 23.8 Reconstruction from a camera mounted on a vehicle (after 23.108) obtained from the fusion of 13 depth maps. Each depth map was obtained from 11-view stereo ▶

We will close with a system of motion and dense mapping representing the state of the art: the work in [23.108] represents the state of the art in large-scale dense reconstruction from monocular image sequences without the use of an additional sensor. The camera poses are obtained by applying the algorithm described above [23.17]. A temporary model is obtained and subsequent poses are estimated by applying pose estimation with preemptive RANSAC [23.109] for a time window after which a reinitialization with a novel triple of views is obtained. A depth map is obtained by back-projecting multiple views on planes that sweep space in prominent directions and maximizing a correlation in the sweeping direction. Multiple depth maps are combined (Fig. 23.8) through median fusion and taking visibility into consideration [23.110].

References

23.1 Z. Zhang: A flexible new technique for camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. **22**, 1330–1334 (2000)

23.2 M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch: Visual modeling with a hand-held camera, Int. J. Comput. Vis. **59**, 207–232 (2004)

23.3 M. Pollefeys, L. Van Gool: Stratified self-calibration with the modulus constraint, IEEE Trans. Pattern Anal. Mach. Intell. **21**, 707–724 (1999)

23.4 O. Faugeras, Q.-T. Luong, T. Papadopoulo: *The Geometry of Multiple Images* (MIT Press, Cambridge 2001)

23.5 R. Hartley, A. Zisserman: *Multiple View Geometry* (Cambridge Univ. Press, Cambridge 2000)

23.6 K. Ottenberg, R.M. Haralick, C.-N. Lee, M. Nolle: Review and analysis of solutions of the three-point perspective problem, Int. J. Comput. Vis. **13**, 331–356 (1994)

23.7 M.A. Fischler, R.C. Bolles: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM **24**, 381–395 (1981)

23.8 R. Kumar, A.R. Hanson: Robust methods for estimaging pose and a sensitivity analysis, Comput. Vis. Image Underst. **60**, 313–342 (1994)

23.9 C.-P. Lu, G. Hager, E. Mjolsness: Fast and globally convergent pose estimation from video images, IEEE Trans. Pattern Anal. Mach. Intell. **22**, 610–622 (2000)

23.10 L. Quan, Z. Lan: Linear n-point camera pose determination, IEEE Trans. Pattern Anal. Mach. Intell. **21**, 774–780 (1999)

23.11 A. Ansar, K. Daniilidis: Linear pose estimation from points and lines, IEEE Trans. Pattern Anal. Mach. Intell. **25**, 578–589 (2003)

23.12 R.I. Hartley, P. Sturm: *Triangulation. Computer Vision and Image Understanding* (1997)

23.13 B.K.P. Horn, H.M. Hilden, S. Negahdaripour: Closed-form solution of absolute orientation using orthonormal matrices, J. Opt. Soc. Am. A **A5**, 1127–1135 (1988)

23.14 G.H. Golub, C.F. van Loan: *Matrix Computations* (The Johns Hopkins Univ. Press, Baltimore 1983)

23.15 A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse: Monoslam: Real-time single camera slam, IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007)

23.16 T.S. Huang, O.D. Faugeras: Some properties of the *e* matrix in two-view motion estimation, IEEE Trans. Pattern Anal. Mach. Intell. **11**, 1310–1312 (1989)

23.17 D. Nister: An efficient solution for the five-point relative pose problem, IEEE Trans. Pattern Anal. Mach. Intell. **26**, 756–777 (2004)

23.18 S. Maybank: *Theory of Reconstruction from Image Motion* (Springer, Berlin, Heidelberg 1993)

23.19 S.J. Maybank: The projective geometry of ambiguous surfaces, Philos. Trans. R. Soc. London A **332**(1623), 1–47 (1990)

23.20 A. Jepson, D.J. Heeger: A fast subspace algorithm for recovering rigid motion, Proc. IEEE Workshop on Visual Motion (Princeton 1991) pp. 124–131

23.21 C. Fermüller, Y. Aloimonos: Algorithmic independent instability of structure from motion, Proc. 5th Eur. Conf. Comput. Vis. (Freiburg 1998)

23.22 K. Daniilidis, M. Spetsakis: Understanding noise sensitivity in structure from motion. In: *Visual Navigation*, ed. by Y. Aloimonos. (Lawrence Erlbaum, Hillsdale 1996), pp.61–88

23.23 S.R. Soatto Brockett: Optimal structure from motion: Local ambiguities and global estimates, IEEE Conf. Comput. Vis. Pattern Recog. (Santa Barbara 1998)

23.24 J. Oliensis: A new structure-from-motion ambiguity, IEEE Trans. Pattern Anal. Mach. Intell. **22**, 685–700 (1999)

23.25 Y. Ma, K. Huang, R. Vidal, J. Kosecka, S. Sastry: Rank conditions of the multiple view matrix, Int. J. Comput. Vis. **59**(2), 115–137 (2004)

23.26 Y. Ma, S. Soatto, J. Kosecka, S. Sastry: *An Invitation to 3-D Vision* (Springer, Berlin, Heidelberg 2003)

23.27 W. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon: Bundle adjustment for structure from motion (Springer Verlag 2000) pp. 298–375

23.28 M. Lourakis, A. Argyros: The design and implementation of a generic sparse bundle adjustment software package based on the Levenberg–Marquard method. Technical Report 340, ICS/FORTH (2004)

23.29 S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg: Calibrated, registered images of an extended urban area, Int. Conf. Comput. Vis. Pattern Recogn., Vol. 1 (Kanai 2001) pp. 813–820

23.30 E. Trucco, A. Verri: *Introductory Techniques for 3-D Computer Vision* (Prentice Hall, Upper Saddle River 1998)

23.31 S.S. Intille, A.F. Bobick: Disparity-space images and large occlusion stereo, ECCV **2**, 179–186 (1994)

23.32 R. Szeliski, D. Scharstein: Sampling the disparity space image, IEEE Trans. Pattern Anal. Mach. Intell. **26**(3), 419–425 (2004)

23.33 R. Yang, M. Pollefeys, G. Welch: Dealing with textureless regions and specular highlights: A progressive space carving scheme using a novel photo-consistency measure, Proc. Int. Conf. Comput. Vis. (2003)

23.34 X. Zabulis, A. Patterson, K. Daniilidis: Digitizing archaeological excavations from multiple monocular views, 5th Int. Conf. 3-D Digital Imag. Mod. (2005)

23.35 R.T. Collins: A space-sweep approach to true multi-image matching, IEEE Conf. Comput. Vis. Pattern Recog. (San Fransisco 1996) pp. 358–363

23.36 T. Kanade, M. Okutomi: A stereo matching algorithm with an adaptive window: Theory and experiment, IEEE Trans. Pattern Anal. Mach. Intell. **16**(9), 920–932 (1994)

23.37 D. Scharstein, R. Szeliski: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, Int. J. Comput. Vis. **47**(1/2/3), 7–42 (2002)

23.38 H. Hirschmuller: Stereo vision in structured environments by consistent semi-global matching, Comput. Vis. Pattern Recog. **02**, 2386–2393 (2006)

23.39 O. Veksler: Stereo correspondence by dynamic programming on a tree, Comput. Vis. Pattern Recog. **2**, 384–390 (2005)

23.40 S. Roy, I. Cox: A maximum-flow formulation of the N-camera stereo correspondence problem, Proc. Int. Conf. Comput. Vis. (1998)

23.41 V. Kolmogorov, R. Zabih: Computing visual correspondence with occlusions using graph cuts, Int. Conf. Comput. Vis. **02**, 508 (2001)

23.42 H.-Y. Shum, J. Sun, N.-N. Zheng: Stereo matching using belief propagation, IEEE Trans. Pattern Anal. Mach. Intell. **25**, 787–800 (2003)

23.43 L. Zhang, S.M. Seitz: Estimating optimal parameters for mrf stereo from a single image pair, IEEE Trans. Pattern Anal. Mach. Intell. **29**(2), 331–342 (2007)

23.44 P.F. Felzenszwalb, D.P. Huttenlocher: Efficient belief propagation for early vision, Comput. Vis. Pattern Recog. **01**, 261–268 (2004)

23.45 H. Hirschmuller: Accurate and efficient stereo processing by semi-global matching and mutual information, Comput. Vis. Pattern Recog. **2**, 807–814 (2005)

23.46 S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski: A comparison and evaluation of multi-view stereo reconstruction algorithms, Comput. Vis. Pattern Recog. **1**, 519–528 (2006)

23.47 C.R. Dyer: Volumetric scene reconstruction from multiple views. In: *Foundations of Image Understanding*, ed. by L. Davis (Kluwer, Boston 2001) pp. 469–489

23.48 D.A. Forsyth, J. Ponce: *Computer Vision: A Modern Approach*, Prentice Hall Professional Technical Reference (Prentice Hall, Upper Saddle River 2002)

23.49 L. Fei Fei, R. Fergus, A. Torralba: Recognizing and learning object categories, Short course given at CVPR 2007 (2007)

23.50 A. Pinz: Object categorization, Foundations and Trends in Computer Graphics and Vision **1**(4), 255–353 (2005)

23.51 A. Guzman: Decomposition of a visual scene into three-dimensional bodies. In: *Automatic Interpretation and Classification of Images*, ed. by A. Grasseli (Academic, New York 1965)

23.52 T.O. Binford: Visual perception by computer, Proc. IEEE Conf. Syst. Contr. (Miami 1971)

23.53 R. Brooks: *Model-Based Computer Vision* (Kluwer Academic, Dordrecht 1984)

23.54 D. Marr, K. Nishihara: Representation and recognition of the spatial organization of three-dimensional shapes, Proc. R. Soc. London B **200**, 269–294 (1978)