Handbook of Robotics

Chapter 32: 3D Vision

Danica Kragic Kostas Daniilidis

January 15, 2015

Contents

Chapter 32

3D Vision

Summary

In this chapter, we describe algorithms for 3D vision that help robots accomplish navigation and grasping. To model cameras, we start with the basics of perspective projection and distortion due to lenses. This projection from a 3D world to a 2D image can be inverted only by using information from the world or multiple 2D views. If we know the 3D model of an object or the location of 3D landmarks, we can solve the pose estimation problem from one view. When two views are available, we can compute the 3D motion and triangulate to reconstruct the world up to a scale factor. When multiple views are given either as sparse viewpoints or a continuous incoming video, then the robot path can be computer and point tracks can yield a sparse 3D representation of the world. In order to grasp objects, we can estimate 3D pose of the end effector or 3D coordinates of the graspable points on the object.

32.1 Introduction

With the rapid progress and cost reduction in digital imaging, cameras became the standard and probably the cheapest sensor on a robot. Unlike positioning (GPS), inertial (IMU), and distance sensors (sonar, laser, infrared) cameras produce the highest bandwidth of data. Exploiting information useful for a robot from such a bit stream is less explicit than in case of GPS or a laser scanner but semantically richer. In the years since the first edition of the handbook, we had significant advances in hardware and algorithms. RGB-D sensors like the Primesense Kinect enabled a new generation of full model reconstruction systems [29] with an arbitrary camera motion. Google's project Tango [18] established the state of the art in visual odometry using the latest fusion methods between visual and inertial data [23]. 3D modeling became a commodity software (see, for example, 123D Catch App from Autodesk) and the widely used open source Bundler [66] has been possible by advances in wide baseline matching and bundle adjustment. Methods for wide baseline matching have been proposed for several variations of pose and structure from motion [37]. Last, the problem of local minima for non-minimal overconstrained solvers has been addressed by a group of method using Branch and Bound global optimization of a sum of fractions subject to convex constraints [31] or an L_∞-norm of the error function [21].

Let us consider the two main robot perception domains: navigation and grasping. 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 *matching* and *structure from motion* algorithm. The result of the trajectory can be used to build a lay out of the environment through dens 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 blind 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 3D map as well as a trajectory of all places it visited, the ultimate *visual simultaneous localization and semantic mapping* problem. In the case of

grasping, the robot detects an object given a learnt representation, and subsequently, it has to estimate the *pose* of the object and in some cases its shape by *triangulation*. When a camera is not mounted on an end-effector, the *absolute orientation* between the hand the object has to be found.

In the next section we will present the geometric foundations for 3D vision and in the last section we describe approaches for grasping.

32.2 Geometric Vision

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

$$\begin{pmatrix} X_{ci} \\ Y_{ci} \\ Z_{ci} \end{pmatrix} = \mathbf{R}_i \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \boldsymbol{T}_i \tag{32.1}$$

where $\mathbf{R}_i$ 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 $\mathbf{R}^T\mathbf{R} = \mathbf{1}$ with determinant one. 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

$$x_i = \frac{X_{ci}}{Z_{ci}} \qquad y_i = \frac{Y_{ci}}{Z_{ci}}. \tag{32.2}$$

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

$$u_i = f\alpha x_i + \beta y_i + c_u \qquad v_i = f y_i + c_v, \tag{32.3}$$

where f is the distance of the image plane to the projection center measured in pixels. It is also called 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 called the image center. 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 modelled polynomially:

$$\begin{aligned} x_i^{dist} &= x_i(1 + k_1 r + k_2 r^2 + k_3 r^3 + \ldots) \\ y_i^{dist} &= y_i(1 + k_1 r + k_2 r^2 + k_3 r^3 + \ldots) \\ \text{where} \quad r^2 &= x_i^2 + y_i^2, \end{aligned}$$

where we temporarily assumed that the image center is at (0,0). The image coordinates (x_i, y_i) in (32.3) have to be replaced with the distorted coordinates (x^{dist}, y^{dist}).

Calibration Recovering the intrinsic parameters when we can make multiple views of a reference pattern like a checker board without variation of the intrinsic parameters has become a standard procedure using tools like the MATLAB calibration toolbox or Zhang's OpenCV calibration function [76]. 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. [59, 58]. When all intrinsic are unknown on the Kruppa equations and several stratified self-calibration approaches [13, 19] which require at least three views. Apart 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

$$\lambda_i \boldsymbol{u}_i = \mathbf{K}_i \begin{pmatrix} \mathbf{R}_i & \boldsymbol{T}_i \end{pmatrix} \boldsymbol{X} = \mathbf{P}\boldsymbol{X} \tag{32.4}$$

where $\lambda_i = Z_{ci}$ is the depth of point $\boldsymbol{X}$ in camera coordinates and $\mathbf{P}$ is the 3x4 projection matrix. The depth λ_i which can be eliminated to obtain two equations relating the world to the pixel coordinates.

32.2.1 Pose estimation or PnP

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 or the Prespective-n-Point problem (PnP). Of course, it presumes the identification of the world points in the image. In robotics, the pose estimation is a variant of the localization problem in a known environment. When grasping objects of known shape PnP yields the target pose for an end-effector module the grasping point positions. We assume that a camera is calibrated and that measurements

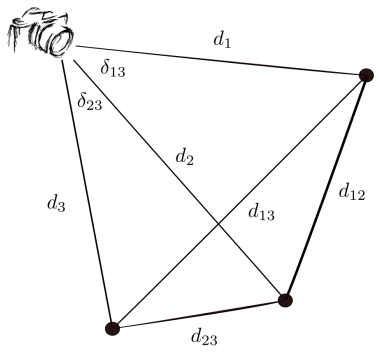

Figure 32.1: Pose estimation problem: A camera seeing 3 points at unknown distances d_1, d_2, and d_3 with known angles between the rays and known point distances d_{12}, d_{13}, d_{23}.

of N points are given in world coordinates $\boldsymbol{X}_{j=1..N}$ and calibrated image coordinates $\boldsymbol{x}_{j=1..N}$. 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. 32.1). Let us denote the squared distance $\|\boldsymbol{X}_i - \boldsymbol{X}_j\|^2$ with d_{ij}^2 and the lengths of $\boldsymbol{X}_j$ with d_j^2. Then cosine law reads

$$d_i^2 + d_j^2 - 2d_i d_j \cos\delta_{ij} = d_{ij}^2 \tag{32.5}$$

If we can recover d_i and d_j the rest will be an absolute orientation problem

$$\mathrm{d}_j \boldsymbol{x}_j = \mathbf{R}\boldsymbol{X}_j + \boldsymbol{T} \tag{32.6}$$

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

Minimal solution 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, so it will have a maximum of eight solutions.

We follow here the analysis of the classic solution in [64] and set $d_2 = ud_1$ and $d_3 = vd_1$ and solve all three equations for d_1:

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

which is equivalent to two quadratic equations in u and v

$$d_{12}^2(1 + v^2 - 2v\cos\delta_{13}) = d_{13}^2(u^2 + 1 - 2u\cos\delta_{12}) \tag{32.7}$$

$$d_{13}^2(u^2 + v^2 - 2uv\cos\delta_{23}) = d_{23}^2(1 + v^2 - 2v\cos\delta_{13}) \tag{32.8}$$

Solving 32.8 for u^2 and substituting in 32.7 allows solving E1 for u because u appears linearly. Substituting u back in 32.8 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 [15, 64]. Popular pose estimation algorithms are based either on an iterative method [38, 42] or linear versions using auxiliary unknowns of higher dimension [61, 2].

A more recent method [39] for n world points expresses 3D points as the barycentric coordinates with respect to four virtual control points:

$$\boldsymbol{X}_i = \sum_{j=1}^{4} \alpha_{ij} \boldsymbol{C}_j \quad \text{where} \quad \sum_{j=1}^{4} \alpha_{ij} = 1.$$

A rigid transformation to the camera coordinate system leaves the barycentric coordinates invariant and a perspective projection yields

$$\lambda_i \boldsymbol{x}_i = \sum_{j=1}^{4} \alpha_{ij} \left(X_{ci}, Y_{ci}, Z_{ci}\right)^T.$$

Eliminating λ_i yields two linear equations for each point

$$\begin{aligned} \sum_{j=1}^{4} \alpha_{ij} C_{x_{cj}} &= \alpha_{ij} x_i C_{z_{cj}} \\ \sum_{j=1}^{4} \alpha_{ij} C_{y_{cj}} &= \alpha_{ij} y_i C_{z_{cj}} \end{aligned}$$

with the coordinate triples of the control points in the camera frame being the 12 unknowns. This is a linear homogeneous system with the solution being the nullspace

of a $2n \times 12$ matrix. The unknown control points are found up to a scale factor which is easily fixed because we know the inter point distances. The pose is found from absolute orientation between control points in the camera and the world frame. This yields a very efficient solution for $n \geq 6$ points but leaves you with the initial choice of the control points as a factor affecting the solution.

In case that $n \geq 4$ points lie on a plane we can compute the homography H between the world and the camera plane [77]. Assuming $Z = 0$ is the world plane the homography reads

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \sim \underbrace{K \begin{pmatrix} r_1 & r_2 & T \end{pmatrix}}_{H} \begin{pmatrix} X \\ Y \\ W \end{pmatrix}$$

where $r_{1,2}$ are the first two columns of the rotation matrix and $\sim$ denotes the projective equivalence, namely, for any two points $\boldsymbol{p}$ and $\boldsymbol{p}'$ in the projective plane $\boldsymbol{p} \sim \boldsymbol{p}'$ iff $\boldsymbol{p} = \lambda \boldsymbol{p}'$ for real $\lambda \neq 0$. Hence the first two columns of $K^{-1}H$

$$K^{-1}H = \begin{pmatrix} h'_1 & h'_2 & h'_3 \end{pmatrix}$$

have to be orthogonal. We seek thus an orthogonal matrix R that is the closest to $\begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix}$:

$$\arg\min_{R \in SO(3)} \| R - \begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix} \|_F^2$$

If the SVD of

$$\begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix} = USV^T$$

then the solution is [17]

$$R = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(UV^T) \end{pmatrix} V^T \tag{32.9}$$

The diagonal matrix is a projection from the orthogonal group $O(3)$ to the special orthogonal group $SO(3)$.

Last, we present a method [24] for n points that computes all local minima of the over constrained PnP problem. This involves solving the first derivatives explicitly with respect to the pose unknowns. To achieve this, following observation allows the elimination of the depths λ and the translation. Rigid transformation $\lambda \boldsymbol{x} = R\boldsymbol{X} + \boldsymbol{T}$ can be written for n points as a linear system for $\lambda_{j=1..n}$ and the translation $\boldsymbol{T}$:

$$\begin{pmatrix} \boldsymbol{x}_1 & & & -I \\ & \ddots & & \vdots \\ & & \boldsymbol{x}_n & -I \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ \boldsymbol{T} \end{pmatrix} = \begin{pmatrix} \mathbf{R}\boldsymbol{X}_1 \\ \vdots \\ \mathbf{R}\boldsymbol{X}_n \end{pmatrix}.$$

We can solve for the unknown depths-translation vector and back substitute it into a least squares minimization problem with respect to rotation parameters. It turns out that if we use the three Rodriguez parameters as rotation parametrization the necessary conditions for an extremum (vanishing derivatives) turn out to be three cubic equations [24]. Last we would like to point out to the reader that a nonlinear function of the rotation matrix can also be solved as an optimization problem on the Lie-group SO(3) [71, 1, 46] for the case of line correspondences.

32.2.2 Triangulation

When we know both the intrinsics and extrinsics or their summarization in matrix $\mathbf{P}$ 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

$$\begin{aligned} \lambda_1 \boldsymbol{u}_1 &= \mathbf{P}_1 \begin{pmatrix} \boldsymbol{X} \\ 1 \end{pmatrix} \\ \lambda_2 \boldsymbol{u}_2 &= \mathbf{P}_2 \begin{pmatrix} \boldsymbol{X} \\ 1 \end{pmatrix} \end{aligned} \tag{32.10}$$

with known projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ 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, we neeed though to remove the distortion in order to write them as above.

Having correspondences of the same point in two cameras with known projection matrices $\mathbf{P}_l$ and $\mathbf{P}_r$ 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.

$$\begin{pmatrix} x\mathbf{P}_l(3,:) & -\mathbf{P}_l(1,:) \\ y\mathbf{P}_l(3,:) & -\mathbf{P}_l(2,:) \\ x\mathbf{P}_r(3,:) & -\mathbf{P}_r(1,:) \\ y\mathbf{P}_r(3,:) & -\mathbf{P}_r(2,:) \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{0}, \tag{32.11}$$

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

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 4x4 matrix above using SVD. A thorough treatment of triangulation is the classic [20].

32.2.3 Moving Stereo

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

$$\begin{aligned} \boldsymbol{u}_{li} &\sim \mathbf{P}_l \boldsymbol{X}_i & (32.12)\\ \boldsymbol{u}_{ri} &\sim \mathbf{P}_r \boldsymbol{X}_i & (32.13) \end{aligned}$$

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

$$\boldsymbol{X}_0 = \mathbf{R}_1 \boldsymbol{X}_1 + \boldsymbol{T}_1 \quad (32.14)$$

where $\boldsymbol{X}_0$ are point coordinates with respect to the world coordinate system, usually assumed aligned 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 $(\mathbf{R}_1, \boldsymbol{T}_1)$. To estimate the motion of the rig, we have to solve two correspondence problems, first, between 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 enable the solution of the triangulation problem at each time instance. Motion can be obtained then by solving equations (32.14) for $(\mathbf{R}_1, \boldsymbol{T}_1)$, 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. The most popular visual odometry system today is `libviso` [32] and is based on a moving stereo rig.

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

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

The standard way [25, 17] to solve this problem is to eliminate the translation by subtracting the centroids yielding

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

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 $n \times 3$ matrix $A_{1,2}$ we can formulate the following minimization of the Frobenius norm

$$\min_{R \in SO(3)} \|A_2 - \mathbf{R}A_1\|_F$$

which is known as the Procrustes problem. It can be shown [17] that the solution is obtained through SVD as in 32.9 where $\mathbf{U}, \mathbf{V}$ are obtained from the singular value decomposition

$$A_2 A_1^T = USV^T.$$

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

32.2.4 Structure from Motion

Relax now the assumption that projection matrices are known and remain with measuring and matching corresponding points $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. This is the well known structure from motion problem or more precisely structure and 3D-motion from 2D motion. In photogrammetry, it is well known as relative orientation problem. Even after eliminating the λ's from equations (32.12) or by writing them in projective equivalence form

$$\begin{aligned} \boldsymbol{u}_1 &\sim \mathbf{P}_1 \begin{pmatrix} \boldsymbol{X} \\ 1 \end{pmatrix} \\ \boldsymbol{u}_2 &\sim \mathbf{P}_2 \begin{pmatrix} \boldsymbol{X} \\ 1 \end{pmatrix} \end{aligned} \quad (32.15)$$

we realize that we if $(\boldsymbol{X}, \mathbf{P}_1, \mathbf{P}_2)$ is a solution than $(\mathbf{H}\boldsymbol{X}, \mathbf{P}_1\mathbf{H}^{-1}, \mathbf{P}_2\mathbf{H}^{-1})$ is a solution, too, where H an invertible 4x4 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 practice is common

$$\begin{aligned} \boldsymbol{u}_1 &\sim \begin{pmatrix} \boldsymbol{I} & \mathbf{0} \end{pmatrix} \boldsymbol{X} \\ \boldsymbol{u}_2 &\sim \mathbf{P}_2 \boldsymbol{X} \end{aligned} \quad (32.16)$$

we remain with the same ambiguity where $\mathbf{H}$ is of the form

$$\mathbf{H} \sim \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ h_{41} & h_{42} & h_{43} & h_{44} \end{pmatrix} \quad (32.17)$$

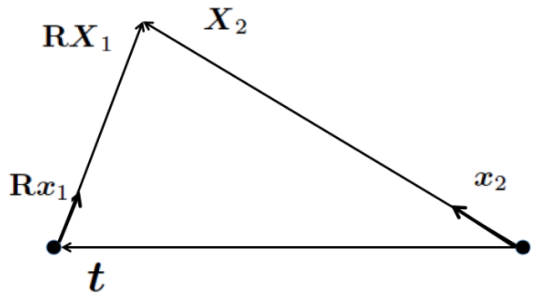

Figure 32.2: A point is perspectively projected to calibrated image vectors $R\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ which are coplanar with baseline $\boldsymbol{T}$.

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:

$$\begin{aligned} \boldsymbol{u}_1 &\sim \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \end{pmatrix} \boldsymbol{X} \\ \boldsymbol{u}_2 &\sim \begin{pmatrix} \mathbf{R} & \boldsymbol{T} \end{pmatrix} \boldsymbol{X} \end{aligned} \tag{32.18}$$

and the only remaining ambiguity is the scale ambiguity where $\mathbf{H}$ looks like an identity matrix except $h_{44} = s \neq 1$ being the scale factor. In other words if $(\mathbf{R}, \boldsymbol{T}, \boldsymbol{X})$ is a solution then $(\mathbf{R}, s\boldsymbol{T}, 1/s\boldsymbol{X})$ is a solution, too. These remarks generalize in multiple views. Because, in robotics the $(\mathbf{R}, \boldsymbol{T})$ matrices correspond to location and $\boldsymbol{X}$) to mapping of the environment, the problem has the more proper term SLAM: Simultaneous localization and mapping. However, because the term SLAM has been used with a variety of sensors like sonar and laser range scanners, the term monocular SLAM is better suited to describe structure from motion from multiple views [8].

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 intersection of the two rays $\mathbf{R}\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is that the two rays are coplanar with the baseline $\boldsymbol{T}$:

$$\boldsymbol{x}_2^T (\boldsymbol{T} \times \mathbf{R}\boldsymbol{x}_1) = 0, \tag{32.19}$$

which is the epipolar constraint, see Fig. 32.2. To avoid the scale ambiguity we assume that $\boldsymbol{T}$ is a unit vector. We proceed by summarizing the unknowns into one matrix

$$\mathbf{E} = \hat{\boldsymbol{T}}\mathbf{R} \tag{32.20}$$

where $\hat{\boldsymbol{T}}$ is the 3x3 skew-symmetric matrix to the vector $\boldsymbol{T}$. The E matrix is called the essential matrix. The epipolar constraint reads then

$$\boldsymbol{x}_2^T \mathbf{E} \boldsymbol{x}_1 = 0 \tag{32.21}$$

which is the equation of a line in the $\boldsymbol{x}_2$ plane with coefficients $E\boldsymbol{x}_1$ or a coefficient of a line in the $\boldsymbol{x}_1$ plane with coefficients $E^T\boldsymbol{x}_2$. 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. The epipoles are the intersections of the baseline with the two image planes, hence $\boldsymbol{e}_2 \sim \boldsymbol{T}$ and $\boldsymbol{e}_1 \sim -\mathbf{R}^T\boldsymbol{T}$. Looking at the equations of the epipolar lines we can immediately infer that $E^T\boldsymbol{e}_1 = 0$ and $E\boldsymbol{e}_2 = 0$.

The set of all essential matrices

$$\begin{aligned} \mathcal{E} \quad = \quad & \Big\{ \mathbf{E} \in \mathbb{R}^{3\times 3} \mid \mathbf{E} = \widehat{\boldsymbol{T}}\mathbf{R}, \\ & \text{where } \boldsymbol{T} \in \mathbb{S}^2 \text{ and } \mathbf{R} \in \mathrm{SO}(3) \Big\} \end{aligned}$$

has been characterized as a manifold of dimension 5 [74]. It has been proven [27] that

Proposition 1 *A matrix $\mathbf{E} \in \mathbb{R}^{3\times 3}$ is essential if and only if it has two singular values equal to each other and third singular value equal zero.*

We present here Nister's method [54] for recovering an essential matrix from five point correspondences and which gained in popularity because of its suitability for RANSAC methods.

Minimal case We expand the epipolar constraint in terms of 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 $z_i = 1$) and obtain

$$\begin{pmatrix} x_1\boldsymbol{x}_2^T & y_1\boldsymbol{x}_2^T & z_1\boldsymbol{x}_3^T \end{pmatrix} E_s = 0 \tag{32.22}$$

where E_s is the raw by raw stacked version of matrix 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:

$$E_s = \lambda_1\boldsymbol{u}_1 + \lambda_2\boldsymbol{u}_2 + \lambda_3\boldsymbol{u}_3 + \lambda_4\boldsymbol{u}_4. \tag{32.23}$$

At this point we want the matrix E resulting from E_s to be an essential matrix satisfying Proposition 1. It has been proven [27] that

Proposition 2 *A matrix $\mathbf{E} \in \mathbb{R}^{3\times 3}$ is essential if and only if*

$$\mathbf{E}\mathbf{E}^T\mathbf{E} = \frac{1}{2} trace(\mathbf{E}\mathbf{E}^T)\mathbf{E}. \tag{32.24}$$

Though the $\det(\mathbf{E}) = 0$ constraint can be inferred from (32.24) we are still going to use it together with (32.24) to obtain ten cubic equations in the elements of $\mathbf{E}$. As described in [54], 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 beyond physical plausibility of the existence of at least one solution that a real root will exist at all. Several alternative 5-point solvers have been proposed since Nister's paper [40, 36, 69, 3] and an extensive list including code has been established by Pajdla's group [1].

Assuming that we have recovered an essential matrix from point correspondences, the next task is to recover an orthogonal matrix $\mathbf{R}$ and a unit vector translation $\boldsymbol{T}$ from the essential matrix. If $E = U\mathrm{diag}(\sigma, \sigma, 0)V^T$, there are four solutions for the pair $(\widehat{T}, R)$:

$$\begin{aligned}
(\widehat{\boldsymbol{T}_1}, R_1) &= (UR_{z,+\pi/2}\Sigma U^T, UR_{z,+\pi/2}^T V^T) \\
(\widehat{\boldsymbol{T}_2}, R_2) &= (UR_{z,-\pi/2}\Sigma U^T, UR_{z,-\pi/2}^T V^T) \\
(\widehat{\boldsymbol{T}_1}, R_2) &= (UR_{z,+\pi/2}\Sigma U^T, UR_{z,-\pi/2}^T V^T) \\
(\widehat{\boldsymbol{T}_2}, R_1) &= (UR_{z,-\pi/2}\Sigma U^T, UR_{z,+\pi/2}^T V^T)
\end{aligned}$$

where R_z denotes rotation around the z-axis. The four solutions can be split into two two-fold ambiguities:

Mirror ambiguity: If T is a solution, then $-T$ is a solution, too. There is no way to disambiguate from the epipolar constraint: $\boldsymbol{x}_2^T((-\boldsymbol{T}) \times R\boldsymbol{x}_1) = 0$.

Twisted pair ambiguity: If R is a solution, then also $R_{T,\pi}R$ is a solution. The first image is "twisted" around the baseline 180 degrees.

These ambiguities are resolved by checking if depths of triangulated points are positive.

Critical Ambiguities The approach with five point correspondences has a finite number of feasible[2] solutions when the points in the scene lie on a plane (a two fold ambiguity) [49] 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 [50]. These are inherent ambiguities which hold for any number of point correspondences when one seeks a solution for an exact essential matrix.

When someone is 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 $\mathbf{E}$.

Beyond the ambiguous situations, there is a considerable amount of literature regarding instabilities in the two view problem. In particularly, it has been shown [49, 30, 14] 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 between translation and rotation [7]. Moreover, it has been shown, that there exist local minima close to the global minimum that can fool any iterative scheme [67, 55].

3-point SfM Minimal solutions based on 5 points are still too slow to be used on mobile platforms where additional information like a reference gravity vector might be obtained from an IMU. We present here a recent solution using a reference direction and only 3 points [53].

We are given three image correspondences from calibrated cameras, and a single directional correspondence like the gravity vector or a vanishing point. This problem is equivalent to finding the translation vector $\mathbf{t}$ and a rotation angle θ around an arbitrary rotation axis.

Let us choose the arbitrary rotation axis to be $\mathbf{e}_2 = [0, 1, 0]^\top$. After taking the directional constraint into account, from the initial five parameters in the essential matrix, we now only have to estimate three. We can use the axis-angle parameterization of a rotation matrix to rewrite the essential matrix constraint as follows:

$$\mathbf{p}_{2i}{}^T \tilde{E} \mathbf{p}_1 = 0, \tag{32.25}$$

where

$$\tilde{E} = \widehat{\tilde{\mathbf{t}}}(I + \sin\theta\widehat{\mathbf{e}}_2 + (1 - \cos\theta)\widehat{\mathbf{e}}_2^2),$$

and $\tilde{\mathbf{t}} = (x, y, 1)$.

Each image point correspondence gives us one such equation, for a total of three equations in three unknowns (elements of $\mathbf{t}$ and θ). To create a polynomial system, we set $s = \sin\theta$ and $c = \cos\theta$, and add the trigonometric constraint $s^2 + c^2 - 1 = 0$, for a total of four equations in four unknowns. In order to reduce the number of unknowns, we choose the direction of the epipole by assuming that the translation vector $\tilde{\mathbf{t}}$ has the form $[x, y, 1]^\top$. This means that for each $\tilde{\mathbf{t}}$ that we recover, $-\tilde{\mathbf{t}}$ will also need to be considered as a possible solution.

Once we substitute for $\tilde{E}$ in equation (32.25), the resulting system of polynomial equations has the following

[1] `cmp.felk.cvut.cz/minimal/5_pt_relative.php`

[2] feasible means that they may produce multiple interpretations of structures in front of the camera

form:

$$a_{i1}xs + a_{i2}xc + a_{i3}ys + a_{i4}yc + a_{i5}x - a_{i2}s + a_{i1}c + a_{i6} = 0 \tag{32.26}$$

for $i = 1, .., 3$, and the equation

$$s^2 + c^2 - 1 = 0. \tag{32.27}$$

This polynomial system can be solved in closed form and has up to four solutions. The total number of possible pose matrices arising from our formulation is therefore at most 8, when we take into account the fact that we have to consider the sign ambiguity in translation.

32.2.5 Multiple views SfM

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

To exploit multiple frames we introduce rank constraints [44]. We assume that the world coordinate system coincides with the coordinate system of the first frame and that a scene point is projected to $\boldsymbol{x}_i$ in the i-th frame and that its depth with respect to the 1st frame is λ_1:

$$\lambda_i \boldsymbol{x}_i = \mathbf{R}_i(\lambda_1 \boldsymbol{x}_1) + \boldsymbol{T}_i. \tag{32.28}$$

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

$$\begin{pmatrix} \widehat{\boldsymbol{x}_2}\mathbf{R}_2\boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2}\boldsymbol{T}_2 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_n}\mathbf{R}_n\boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2}\boldsymbol{T}_n \end{pmatrix} \begin{pmatrix} \lambda_1 \\ 1 \end{pmatrix} = 0 \tag{32.29}$$

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 [45], a constraint that infers both the epipolar and the trifocal equations. The least squares solution for the depth can easily be derived as

$$\lambda_1 = -\frac{\sum_{i=1}^{n}(\boldsymbol{x}_i \times \boldsymbol{T}_i)^T(\boldsymbol{x}_i \times \mathbf{R}_i\boldsymbol{x}_1)}{\|\boldsymbol{x}_i \times \mathbf{R}_i\boldsymbol{x}_1\|^2}. \tag{32.30}$$

Given a depth for each point we can solve for motion by rearranging the multiple views constraint (32.29) as

$$\begin{pmatrix} \lambda_1^1 \boldsymbol{x}_1^{1T} \otimes \hat{\boldsymbol{x}_i^1} & \hat{\boldsymbol{x}_i^1} \\ \vdots & \vdots \\ \lambda_1^n \boldsymbol{x}_1^{nT} \otimes \hat{\boldsymbol{x}_i^n} & \hat{\boldsymbol{x}_i^n} \end{pmatrix} \begin{pmatrix} \mathbf{R}_i^{stacked} \\ \boldsymbol{T}_i \end{pmatrix} = 0 \tag{32.31}$$

where $\boldsymbol{x}_i^n$ is the n-th image point in the i-th frame and $\mathbf{R}_i, \boldsymbol{T}_i$ is the motion from 1st to the i-th frame and $\mathbf{R}_i^{stacked}$ is the 12x1 vector of stacked elements of the rotation matrix $\mathbf{R}_i$. Suppose that $\boldsymbol{k}$ is the 12x1 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 $\mathbf{A}$ the 3x3 matrix obtained from the first 9 elements of $\boldsymbol{k}$ and $\boldsymbol{a}$ the vector of elements 10 to 12. To obtain a rotation matrix we follow the SVD steps in the solution of absolute orientation 32.14 to find the closest orthogonal matrix to an arbitrary invertible matrix.

Bundle Adjustment On top of such an approach, a bundle adjustment [73] minimizes the sum of all deviations between image coordinates and the backprojections of the points to be reconstructed.

$$\underset{R^f, T^f, X_p}{\arg\min}\ \epsilon^T C^{-1} \epsilon$$

minimized with respect to all $6(F-1)$ motions and $3N-1$ structure unknowns, where ϵ is the vector containing all errors

$$\epsilon_p^f = \begin{pmatrix} x_p^f - \dfrac{R_{11}^f X_p + R_{12}^f Y_p + R_{13}^f Z_p + T_x}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \\ y_p^f - \dfrac{R_{21}^f X_p + R_{22}^f Y_p + R_{23}^f Z_p + T_y}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \end{pmatrix}$$

and C is the error covariance matrix. We will continue with the assumption that $C = I$.

Call the objective function $\Phi(u) = \epsilon(u)^T\epsilon(u)$ with u the vector of unknowns. Given a starting value for the vector of unknowns u we iterate with steps Δu by locally fitting a quadratic function to $\Phi(u)$:

$$\Phi(u + \Delta u) = \Phi(u) + \Delta u^T \nabla\Phi(u) + \frac{1}{2}\Delta u^T H(u) \Delta u$$

where $\nabla\Phi$ is the gradient and H is the Hessian of Φ. The minimum of this local quadratic is at Δu satisfying

$$H\delta u = -\nabla\Phi(u)$$

If $\Phi(u) = \epsilon(u)^T\epsilon(u)$ then

$$\nabla\Phi = 2\sum_i \epsilon_i(u)\nabla\epsilon_i(u)^T = J(u)^T\epsilon$$

where the Jacobian J consists of elements

$$J_{ij} = \frac{\partial \epsilon_i}{\partial u_j}$$

and the Hessian reads

$$\begin{aligned} H &= 2\sum_i \left(\nabla\epsilon_i(u)\nabla\epsilon_i(u)^T + \epsilon_i(u)\frac{\partial^2\epsilon_i}{\partial u^2} \right) \\ &= 2\left(J(u)^T J(u) + \sum_i \epsilon_i(u)\frac{\partial^2\epsilon_i}{\partial u^2} \right) \approx 2J(u)^T J(u) \end{aligned}$$

by omitting quadratic terms inside the Hessian. This yields the Gauss-Newton iteration

$$(J^T J)\Delta u = J^T \epsilon$$

involving the inversion of a $(6F+3N-7)\times(6F+3N-7)$ matrix. Bundle adjustment is about the "art" of inverting efficiently $(J^T J)$.

Let us split the unknown vector u into $u = (a, b)$ following [41] obtaining

- $6F - 6$ motion unknowns a
- $3P - 1$ structure unknonws b

and we will explain this case better if we assume two motion unknowns a_1 and a_2 corresponding to 2 frames, and 3 unknown points b_1, b_2, b_3.

For keeping symmetry in writing we do not deal here with the global reference and the global scale ambiguity.

The Jacobian for 2 frames and 3 points has 6 pairs of rows (one pair for each image projection) and 15 columns/unknowns:

$$J = \frac{\partial\epsilon}{\partial(a,b)} = \begin{pmatrix} & & & & \\ A_1^1 & 0 & B_1^1 & 0 & 0 \\ 0 & A_1^2 & B_1^2 & 0 & 0 \\ A_2^1 & 0 & 0 & B_2^1 & 0 \\ 0 & A_2^2 & 0 & B_2^2 & 0 \\ A_3^1 & 0 & 0 & 0 & B_3^1 \\ \underbrace{0}_{\text{motion}} & \underbrace{A_3^2}_{\text{motion}} & \underbrace{0}_{\text{structure}} & \underbrace{0}_{\text{structure}} & \underbrace{B_3^2}_{\text{structure}} \end{pmatrix}$$

with A matrices being 2×6 and B matrices being 2×3 being Jacobians of the error ϵ_i^f of the projection of the i-th point in the f-th frame. We observe now a pattern emerging in $J^T J$

$$J^T J = \begin{pmatrix} U^1 & 0 & W_1^1 & W_2^1 & W_3^1 \\ 0 & U^2 & W_1^2 & W_2^2 & W_3^3 \\ .. & .. & V_1 & 0 & 0 \\ .. & .. & 0 & V_2 & 0 \\ .. & .. & 0 & 0 & V_3 \end{pmatrix}$$

with the block diagonals for motion and structure separated. Let us rewrite the basic iteration $(J^T J)\Delta u = J^T\epsilon$ as

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix}\begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix} = \begin{pmatrix} \epsilon'_a \\ \epsilon'_b \end{pmatrix}$$

and premultiply with

$$\begin{pmatrix} I & WV^{-1} \\ 0 & I \end{pmatrix}\begin{pmatrix} U & W \\ W^T & V \end{pmatrix}\begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix} = \begin{pmatrix} I & WV^{-1} \\ 0 & I \end{pmatrix}\begin{pmatrix} \epsilon'_a \\ \epsilon'_b \end{pmatrix}$$

We find out that motion parameters can be updated separately by inverting a $6F \times 6F$ matrix:

$$(U - WV^{-1}W^T)\Delta a = \epsilon'_a - WV^{-1}\epsilon'_b$$

Each 3D point can be be updated separately by inverting a $3\times$ 3 matrix V:

$$V\Delta b = \epsilon'_b - W^T\Delta a$$

It is worth mentioning that bundle adjustment though extremely slow captures the correlation between motion estimates and structure (3D points) estimates which is artificially hidden in the iterative scheme in (32.29).

The largest scale motion estimation and registration of views has been performed by Teller [72] with a decoupled computation first of relative rotations and finally of relative translations. The above multiple view SfM techniques can also be applied in a sliding window mode in time. Davison [8] showed the first real-time recursive approach by decoupling the direction of the viewing rays from the depth unknowns. For other recursive approaches the reader is referred to the corresponding SLAM chapter.

32.3 3D Vision for Grasping

In this section we will move from the basic geometry required for grasping to the main 3D vision challenges associated with the limited knowledge we might have about the shape of the object as well as the actual selection of 3D grasping poses.

Naturally, object grasping and manipulation is closely related to general scene understanding and problems such as object detection, recognition, categorization and pose estimation. Taking all the above, there are very few approaches that address all the problems in a single system. One example, reported in [43], addresses the problem of enabling transfer of grasp knowledge between object categories, defined using both their physical properties and functionality. This is a challenging problem given that a number of objects with similar physical properties afford different tasks. An example can be a

screwdriver and a carrot that are structurally alike, but only the former can be used as a tool, or a ball and an orange where only the latter affords eating, see Fig. 32.3.

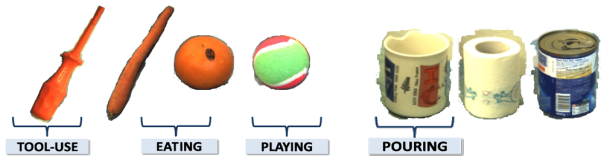

Figure 32.3: Examples of physically similar objects that afford different tasks.

In relation to object grasping in particular, there are methods that assume that full 3D model of the object is available and concentrate on grasp synthesis solely. In addition, many of the approaches conduct experiments in a simulated environment without working with real sensory data. However, the knowledge generated in simulation can also be applied later onto sensory data. Another group of approaches considers grasp synthesis on real sensory data directly, dealing with problems such as noise, occlusions and missing data.

If the object to be grasped is known, there are approaches that store a database of grasp hypotheses, generated either in simulation or through experiments in a real setting. Most of the approaches assume that a 3D mesh of the object is available and the challenge is then to automatically generate a set of feasible grasp hypotheses. This involves sampling the infinite space of possible hand configurations and ranking the resulting grasps according to some quality metric.

To simplify the process, a common approach is to approximate object's shape with a constellation of primitives such as spheres, cones, cylinders, boxes or superquadrics, [51, 28, 10, 60, 16]. The purpose of using shape primitives is to reduce the number of candidate grasps and thus prune the search space for finding the optimal set of grasp hypotheses.

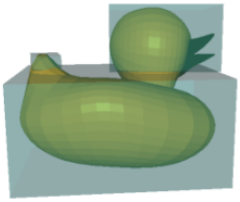

Figure 32.4: Generation of grasp candidates through object shape approximation and decomposition from [28].

One example, shown in Fig. 32.3 and reported in [28], decomposes a point cloud from a stereo camera into a constellation of boxes. Grasp planning is performed directly on the boxes which reduces the number of potential grasps. [11] distinguishes between graspable and non-graspable parts of an object where each part is represented by fitting a superquadric to the point cloud data. [57] approximate an object with a single superquadric and use a Support Vector Machines based approach to search for the grasp that maximizes the grasp quality. [6] models an object as a Markov Random Field (MRF) in which the nodes are points from the point cloud and edges are spanned between the six nearest neighbors of a point. A node in the MRF carries either one of the two labels: a good or a bad grasp location. [9] models the object as a constellation of local multi-modal contour descriptors. The set of associated grasp hypotheses is modeled as a non-parametric density function in the space of 6D gripper poses, referred to as a *bootstrap* density. [56] demonstrates 3D object recognition and pose estimation in a grasping scenario considering cluttered scenes. [75] proposes a metric suitable for predicting grasp stability under pose uncertainty.

There are several approaches that deal specifically with incomplete point clouds. [48] exploits symmetry by fitting a curve to a cross section of the point cloud. [62] concentrates of depth segmentation and sample grasp points from the surface of a segmented object using surface normals. [4] presents a related approach that reconstructs full object shape assuming planar symmetry and generates grasps based on the global shape of the object. [5] makes no prior assumption about the shape of the object and apply shape carving for generating a parallel-jaw gripper grasps. [26] employs heuristics for generating grasp hypotheses dependent on the shape of the point cloud. Recent work in [65] identifies regions that afford force closure grasps by evaluating local curvature of the objects to create an initial opposing grasp with two or three fingers, dependent on the relative size of the object with respect to the hand. [63] uses a stereo-camera setup to generate a 3D representation of a scene with several objects and then generates various top grasps on object candidates. [47] use time-of-flight range data, model objects using 3D Gaussians and rely on finger torque information during grasping to monitor the grasp execution. [70] generate grasp hypotheses based on eigenvectors of the object's *footprints* that are generated by projecting the 3D object point cloud onto the supporting surface. The work of [33] presents a system for general scene un-

derstanding used for grasp planning and execution. The system uses a bottom-up grouping approach where contour and surface structures are used as the basis for grasp planning. The work builds upon previous work presented in [34].

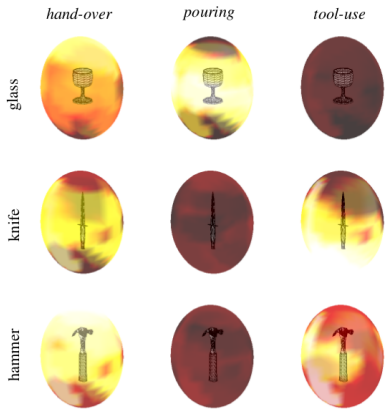

Figure 32.5: Ranking of approach vectors on different objects given a specific task. The brighter an area the higher the rank. The darker an area, the lower the rank. [68]

Most of the recent work concentrates on grasp generalization either by observing human grasping or through off- and on-line learning directly on the robot. [35] demonstrates generalization capabilities using a pouring task scenario. The goal of the approach is to find a part of the object that is most likely to afford the demonstrated action. The learning method method is based on the kernel logistic regression. [22] stores a set of local templates of object that a human is interacting with. If a local part of an object segmented online is similar to a template in the database, the associated grasp hypothesis is executed. [68] approach the problem of inferring a full grasp configuration in relation to a specific task the object is intended for. As in [52], the joint distribution over various grasping variables is modeled as a Bayesian network. Additional variables like task, object category and task constraints are introduced. The structure of this model is learned given a large number of grasp examples generated in a simulator and annotated with grasp quality metrics as well as suitability for a specific task. The learned quality of grasps on specific objects given a task is visualized in Fig. 32.5.

32.4 Conclusion and Further Reading

As main additional sources of reading, we recommend the textbooks by Hartley and Zisserman [19], Ma et al. [45], Faugeras [12], and Faugeras and Luong [13]. The reader is referred to Chapter 5 for fundamentals of estimation, to Chapter 35 for sensor fusion, to Chapter 34 for visual servoing, to Chapter 31 fro Range Sensing, to Chapter 45 for 3D models of the world, and to Chapter 46 for SLAM.

3D vision is a rapidly advancing field and in this chapter we have covered only geometric approaches based on RGB cameras. Although depth sensors will become ubiquitous indoors and might be outdoors as well, RGB cameras remain formidable because of the higher number and larger diversity of features that can be matched and used for pose estimation and 3D-modelling. Long range sensing can still be covered from motion with large translation while active sensors are constrained in terms of energy reflected from the environment.

Bibliography

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] A. Ansar and K. Daniilidis. Linear pose estimation from points and lines. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25:578–589, 2003.

[3] Dhruv Batra, Bart Nabbe, and Martial Hebert. An alternative formulation for five point relative pose problem. In *Motion and Video Computing, 2007. WMVC'07. IEEE Workshop on*, pages 21–21. IEEE, 2007.

[4] Jeannette Bohg, Matthew Johnson-Roberson, Beatriz León, Javier Felip, Xavi Gratal, Niklas Bergström, Danica Kragic, and Antonio Morales. Mind the Gap - Robotic Grasping under Incomplete Observation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2011.

[5] G. M. Bone, A. Lambert, and M. Edwards. Automated Modelling and Robotic Grasping of Unknown Three-Dimensional Objects. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 292–298, 2008.

[6] Abdeslam Boularias, Oliver Kroemer, and Jan Peters. Learning robot grasping from 3-d images with markov random fields. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1548–1553, 2011.

[7] K. Daniilidis and M. Spetsakis. Understanding noise sensitivity in structure from motion. In Y. Aloimonos, editor, *Visual Navigation*, pages 61–88. Lawrence Erlbaum Associates, Hillsdale, NJ, 1996.

[8] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[9] Renaud Detry, Emre Başeski, Norbert Krüger, Mila Popović, Younes Touati, Oliver Kroemer, Jan Peters, and Justus Piater. Learning object-specific grasp affordance densities. In *IEEE Int. Conf. on Development and Learning*, pages 1–7, 2009.

[10] C. Dunes, E. Marchand, C. Collowet, and C. Leroux. Active Rough Shape Estimation of Unknown Objects. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3622–3627, 2008.

[11] S. El-Khoury and A. Sahbani. Handling Objects By Their Handles. In *IEEE/RSJ International Conference on INtelligent Robots and Systems, Workshop on Grasp and Task Learning by Imitation*, 2008.

[12] O. Faugeras. *Three-dimensional Computer Vision*. MIT-Press, Cambridge, MA, 1993.

[13] O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. MIT Press, 2001.

[14] C. Fermüller and Y. Aloimonos. Algorithmic independent instability of structure from motion. In *Proc. Fifth European Conference on Computer Vision*, Freiburg, Germany, 1998.

[15] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[16] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof. Grasp Planning Via Decomposition Trees. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4679–4684, 2007.

[17] G.H. Golub and C.F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, 1983.

[18] Google. Atap project tango.

https://www.google.com/atap/projecttango, 2014.

[19] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge Univ. Press, 2000.

[20] RI Hartley and P Sturm. Triangulation. *Computer Vision and Image Understanding*, 1997.

[21] Richard I Hartley and Fredrik Kahl. Global optimization through rotation space search. *International Journal of Computer Vision*, 82(1):64–79, 2009.

[22] Alexander Herzog, Peter Pastor, Mrinal Kalakrishnan, Ludovic Righetti, Tamim Asfour, and Stefan Schaal. Template-Based Learning of Grasp Selection. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[23] Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, page 0278364913509675, 2013.

[24] Joel A Hesch and Stergios I Roumeliotis. A direct least-squares (dls) method for pnp. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 383–390. IEEE, 2011.

[25] B.K.P. Horn, H.M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal Opt. Soc. Am. A*, A5:1127–1135, 1988.

[26] Kaijen Hsiao, Sachin Chitta, Matei Ciocarlie, and E. Gil Jones. Contact-reactive grasping of objects with partial shape information. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1228 – 1235, October 2010.

[27] T.S. Huang and O.D. Faugeras. Some properties of the e matrix in two-view motion estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11:1310–1312, 1989.

[28] K. Hübner and D. Kragic. Selection of Robot Pre-Grasps using Box-Based Shape Approximation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1765–1770, 2008.

[29] Shahram Izadi, Richard A Newcombe, David Kim, Otmar Hilliges, David Molyneaux, Steve Hodges, Pushmeet Kohli, Jamie Shotton, Andrew J Davison, and Andrew Fitzgibbon. Kinectfusion: real-time dynamic 3d surface reconstruction and interaction. In *ACM SIGGRAPH 2011 Talks*, page 23. ACM, 2011.

[30] A. Jepson and D.J. Heeger. A fast subspace algorithm for recovering rigid motion. In *Proc. IEEE Workshop on Visual Motion*, pages 124–131, Princeton, NJ, Oct. 7-9, 1991.

[31] Fredrik Kahl, Sameer Agarwal, Manmohan Krishna Chandraker, David Kriegman, and Serge Belongie. Practical global optimization for multiview geometry. *International Journal of Computer Vision*, 79(3):271–284, 2008.

[32] Bernd Kitt, Andreas Geiger, and Henning Lategahn. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *Intelligent Vehicles Symposium (IV)*, 2010.

[33] Gert Kootstra, Mila Popovic, Jimmy Alison Jrgensen, Kamil Kuklinski, Konstantsin Miatliuk, Danica Kragic, and Norbert Kruger. Enabling grasping of unknown objects through a synergistic use of edge and surface information. *Int. Jour. of Robotics Research*, 31(10):1190–1213, 2012.

[34] D. Kraft, N. Pugeault, E. Baseski, M. Popovic, D. Kragic, S. Kalkan, F. Wörgötter, and N. Krueger. Birth of the object: Detection of objectness and extraction of object shape through object action complexes. *Int. Jour. of Humanoid Robotics*, pages 247–265, 2009.

[35] Oliver Kroemer, E Ugur, E Oztop, and Jan Peters. A Kernel-based Approach to Direct Action Perception. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[36] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems. In *BMVC*, pages 1–10, 2008.

[37] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Polynomial eigenvalue solutions to minimal problems in computer vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1381–1393, 2012.

[38] R. Kumar and A. R. Hanson. Robust methods for

estimaging pose and a sensitivity analysis. *Computer Vision and Image Understanding*, 60:313–342, 1994.

[39] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

[40] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 630–633, 2006.

[41] M. Lourakis and 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.

[42] C.-P. Lu, G. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:610–622, 2000.

[43] D. Kragic M. Madry, D. Song. From object categories to grasp transfer using probabilistic reasoning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1716–1723, 2012.

[44] Y. Ma, K. Huang, R. Vidal, J. Kosecka, and S. Sastry. Rank conditions of the multiple view matrix. *IJCV*, 2003.

[45] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2003.

[46] Yi Ma, Jana Košecká, and Shankar Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *International Journal of Computer Vision*, 44(3):219–249, 2001.

[47] A. Maldonado, U. Klank, and M. Beetz. Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2586–2591, October 2010.

[48] Zoltan Csaba Marton, Dejan Pangercic, Nico Blodow, Jonathan Kleinehellefort, and Michael Beetz. General 3D Modelling of Novel Objects from a Single View. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3700 – 3705, October 18-22 2010.

[49] S. Maybank. *Theory of Reconstruction from Image Motion*. Springer-Verlag, Berlin et al., 1993.

[50] S.J. Maybank. The projective geometry of ambiguous surfaces. *Phil. Trans. Royal Soc. London*, A 332(1623):1–47, 1990.

[51] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen. Automatic Grasp Planning Using Shape Primitives. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1824–1829, 2003.

[52] Luis Montesano, Manuel Lopes, Alexandre Bernardino, and José Santos-Victor. Learning object affordances: From sensory–motor coordination to imitation. *IEEE Transactions on Robotics*, 24(1):15–26, 2008.

[53] O. Naroditsky, X.S. Zhou, J. Gallier, S. Roumeliotis, and K. Daniilidis. Two efficient solutions for visual odometry using directional correspondence. *IEEE Trans. Patterns Analysis Machine Intelligence*, 2012.

[54] D. Nister. An efficient solution for the five-point relative pose problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26:756–777, 2004.

[55] J. Oliensis. A new structure-from-motion ambiguity. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:685–700, 1999.

[56] Chavdar Papazov, Sami Haddadin, Sven Parusel, Kai Krieger, and Darius Burschka. Rigid 3d geometry matching for grasping of known objects in cluttered scenes. *Int. Jour. of Robotics Research*, 31(4):538–553, 2012.

[57] R. Pelossof, A. Miller, P. Allen, and T. Jebera. An SVM learning approach to robotic grasping. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3512–3518, 2004.

[58] M. Pollefeys and L.Ṽan Gool. Stratified self-calibration with the modulus constraint. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21:707–724, 1999.

[59] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *Int. Journal of Computer Vision*, 59:207–232, 2004.

[60] Markus Przybylski and Tamim Asfour. Unions of

balls for shape approximation in robot grasping. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1592–1599, Taipei, Taiwan, October 2010. IEEE.

[61] L. Quan and Z. Lan. Linear n-point camera pose determination. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21:774–780, 1999.

[62] Deepak Rao, Quoc V. Le, Thanathorn Phoka, Morgan Quigley, Attawith Sudsang, and Andrew Y. Ng. Grasping novel objects with depth segmentation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2578–2585, Taipei, Taiwan, October 2010.

[63] M. Richtsfeld and M. Vincze. Grasping of Unknown Objects from a Table Top. In *ECCV Workshop on 'Vision in Action: Efficient strategies for cognitive agents in complex environments'*, 2008.

[64] K. Ottenberg R.M. Haralick, C.-N. Lee and M. Nolle. Review and analysis of solutions of the three-point perspective problem. *International Journal of Computer Vision*, 13:331–356, 1994.

[65] Maximo A Roa, Max J Argus, Daniel Leidner, Christoph Borst, and Gerd Hirzinger. Power Grasp Planning for Anthropomorphic Robot Hands. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[66] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[67] S. Soatto and R. Brockett. Optimal structure from motion: Local ambiguities and global estimates. In *IEEE Conf. Computer Vision and Pattern Recognition*, Santa Barbara, CA, June 23-25, 1998.

[68] Dan Song, Carl Henrik Ek, Kai Hübner, and Danica Kragic. Multivariate discretization for bayesian network structure learning in robot grasping. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1944–1950, 2011.

[69] Henrik Stewenius, Christopher Engels, and David Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006.

[70] J. Stückler, R. Steffens, D. Holz, and S. Behnke. Real-Time 3D Perception and Efficient Grasp Planning for Everyday Manipulation TaskS. In *European Conf. on Mobile Robots (ECMR)*, 2011.

[71] Camillo J Taylor and David J Kriegman. Minimization on the lie group so (3) and related manifolds. *Yale University*, 1994.

[72] S. Teller, M. Antone, Z. Bodnar, M. Bosse, and S. Coorg. Calibrated, registered images of an extended urban area. In *Int. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 813–820, Kauai, Hawaii, USA, 2001.

[73] W. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment for structure from motion, 2000.

[74] Roberto Tron and Kostas Daniilidis. On the quotient representation for the essential manifold. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1574–1581, 2014.

[75] Jonathan Weisz and Peter K Allen. Pose Error Robust Grasping from Contact Wrench Space Metrics. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 557–562, 2012.

[76] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:1330–1334, 2000.

[77] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.