2 Basic Relations: Image Sequences – "the World"

Vision is a process in which temporally changing intensity and color values in the image plane have to be interpreted as processes in the real world that happen in 3-D space over time. Each image of today's TV cameras contains about half a million pixels. Twenty five (or thirty) of these images are taken per second. This high image frame rate has been chosen to induce the impression of steady and continuous motion in human observers. If each image were completely different from the others, as in a slide show with snapshots from scenes taken far apart in time and space, and were displayed at normal video rate as a film, nobody would understand what is being shown. The continuous development of action that makes films understandable is missing.

This should make clear that it is not the content of each single image, which constitutes the information conveyed to the observer, but the relatively slow development of motion and of action over time. The common unit of 1 second defines the temporal resolution most adequate for human understanding. Thus, relatively slow moving objects and slow acting subjects are the essential carriers of information in this framework. A bullet flying through the scene can be perceived only by the effect it has on other objects or subjects. Therefore, the capability of visual perception is based on the ability to generate internal representations of temporal processes in 3-D space and time with objects and subjects (synthesis), which are supported by feature flows from image sequences (analysis). This is an animation process with generically known elements; both parameters defining the actual 3-D shape and the time history of the state variables of objects observed have to be determined from vision.

In this "analysis by synthesis" procedure chosen in the 4-D approach to dynamic vision, the internal representations in the interpretation process have four independent variables: three orthogonal space components (3-D space) and time. For common tasks in our natural (mesoscale, that is not too small and not too large) environment, these variables are known to be sufficiently representative in the classical nonrelativistic sense.

As mentioned in the introduction, fast image sequences contain quite a bit of redundancy, since only small changes occur from one frame to the next, in general; massive bodies show continuity in their motion. The characteristic frequencies of human and most animal motion are less than a few oscillations per second (Hz), so that at video rate, at least a dozen image frames are taken per oscillation period. According to sampled data theory, this allows good recognition of the dynamic parameters in frequency space (time constants, eigenfrequencies, and damping). So, the task of visual dynamic scene understanding can be described as follows:

Looking at 2-D data arrays generated by several hundred thousands of sensor elements, come up with a distribution of objects in the real world and of their relative motion. The sensor elements are arranged in a uniform array on the chip, usually. Onboard vehicles, it cannot be assumed that the sensor orientation is known beforehand or even stationary. However, inertial sensors for linear acceleration components and rotational rates are available for sensing ego-motion.

It is immediately clear that knowledge about object classes and the way their visible features are mapped into the image plane is of great importance for image sequence understanding. These objects may be grouped in classes with similar functionality and/or appearance. The body of the vehicle carrying the sensors and providing the means for locomotion is, of course, of utmost importance. The lengthy description of the previous sentence will be abbreviated by the term: the "own" body. To understand its motion directly and independently of vision, signals from other sensors such as odometers, inertial angular rate sensors and linear accelerometers as well as GPS (from the "Global Positioning System" providing geographic coordinates) are widely used.

Image data points carry no direct information on the distance at which their light sources, which have stimulated the sensor signal are in the real world; the third dimension (range) is completely lost in a single image (except maybe for intensity attenuation over longer distances). In addition, since perturbations may invalidate the information content of a single pixel almost completely, useful image features consist of signals from groups of sensor elements where local perturbations tend to be leveled out. In biological systems, these are the receptive fields; in technical systems, these are evaluation masks of various sizes. This now allows a more precise statement of the *vision task*:

By looking at the responses of feature extraction algorithms, try to find objects and subjects in the real world and their relative state to the own body. When knowledge about motion characteristics or typical behaviors is available, exploit this in order to achieve better results and deeper understanding by filtering the measurement data over time.

For simple massive objects (*e.g.*, a stone, our sun and moon) and man-made vehicles, good "dynamic models" describing motion constraints are known very often. To describe relative or absolute motion of objects precisely, suitable reference coordinate systems have to be introduced. According to the wide scale of space accessible by vision, certain scales of representation are advantageous:

- Sensor elements have dimensions in the micrometer range (μm).
- Humans operate directly in the meter (m) range: reaching space, single step (body size).
- For projectiles and fast vehicles, the range of immediate reactions extends to several hundred meters or kilometers (km).
- Missions may span several hundred to thousands of kilometers, even one-third to one-half around the globe in direct flight.

- Space flight and lighting from our sun and moon extend up to 150 million km as a characteristic range (radius of Earth orbit).
- Visible stars are far beyond these distances (not of interest here).

Is it possible to find one single type of representation covering the entire range? This is certainly not achievable by methods using grids of different scales as often done in "artificial intelligence"- approaches. Rather, the approach developed in computer graphics with normalized shape descriptions and overall scaling factors is the prime candidate. Homogeneous coordinates as introduced by [Roberts 1965, Blinn 1977] also allow, besides scaling, incorporating the perspective mapping process in the same framework. This yields a unified approach for computer vision and computer graphics; however, in computer vision, many of the variables entering the homogeneous transformation matrices are the unknowns of the problem. A direct application of the methods from computer graphics is thus impossible, since the inversion of perspective projection is a strongly nonlinear problem with the need to recover one space component completely lost in mapping (range).

Introducing strong constraints to the temporal evolution of (3-D) spatial trajectories, however, allows recovering part of the information lost by exploiting first-order derivatives. This is the big advantage of spatiotemporal models and recursive least-squares estimation over direct perspective inversion (computational vision). The Jacobian matrix of this approach to be discussed throughout the text plays a vital role in the 4-D approach to image sequence understanding.

Before this can be fully appreciated, the chain of coordinate transformations from an object-centered feature distribution for each object in 3-D space to the storage of the 2-D image in computer memory has to be understood.

2.1 Three-dimensional (3-D) Space and Time

Each point in space may be specified fully by giving three coordinates in a well-defined frame of reference. This reference frame may be a "Cartesian" system with three orthonormal directions (Figure 2.1a), a spherical (polar) system with one (radial) distance and two angles (Figure 2.1b), or a cylindrical system as a mixture of both, with two orthonormal axes and one angle (Figure 2.1c).

The basic plane of reference is usually chosen to yield the most simple description of the problem: In orbital mechanics, the plane of revolution is selected for reference. To describe the shape of objects, planes of symmetry are preferred; for example, Figure 2.2 shows a rectangular box with length L, width B and height H. The total center of gravity S_t is given by the intersection of two space diagonals. It may be considered the box encasing a road vehicle; then, typically, L is largest and its direction determines the standard direction of travel. Therefore, the centerline of the lower surface is selected

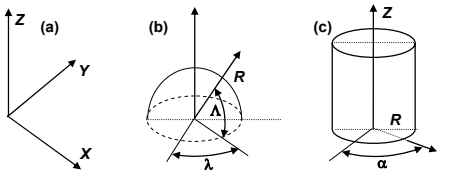

Figure 2.1. Basic coordinate systems (CS): (a) Cartesian CS, (b) spherical CS, (c) cylindrical CS

as the x-direction of a body-fixed coordinate system (x_b, y_b, z_b) with its origin 0_b at the projection S_b of S_t onto the ground plane.

To describe motion in an all-dominating field of gravity, the plane of reference may contain both the gravity and the velocity vector with the origin at the center of gravity of the moving object. The "horizontal" plane normal to the gravity vector also has some advantages, especially for vehicle dynamics since no gravity component affects motion in it.

If a rigid object moves in 3-D space, it is most convenient to describe the shape of the object in an object-oriented frame of reference with its origin at the center (possibly even the center of gravity) or some other convenient, easily definable point (probably at its surface). In Figure 2.2, the shape of the rectangular box is defined by the lengths of its sides L, B, and H. The origin is selected at the center of the ground plane S_b. If the position and orientation of this box has to be described relative to another object, the frame of reference given in the figure has to be related to the independently defined one of the other object by three translations and three rotations, in general.

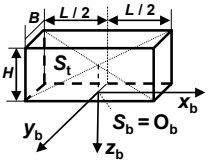

Figure 2.2. Object-oriented coordinate system for a rectangular box

To describe the object (box) shape in the new frame of reference, a coordinate transformation for all significant points defining the shape has to be performed. For the rectangular box, these are its eight corner points located at $\pm L/2$ and $\pm B/2$ for $z_b = 0$ and $-H$. The straight edges of the box remain linear connections between these points. [The selection of the coordinate axes has been performed according to the international standard for aero-space vehicles. X is in the standard direction of motion, x and z are in the plane of vehicle symmetry, and y completes a right-handed set of coordinates. The origin at the lower outside of the body alleviates measurements and is especially suited for ground vehicles, where the encasing box touches the ground due to gravity, in the normal case. Measuring altitude (elevation) positively upward requires a sign change from the positive z-direction (direction of the gravity vector in normal level flight). For this reason, some national standards for ground vehicles rotate the coordinate system by 180° around the x-axis (z upward and y to the left).]

In general, the coordinate transformations between two systems in 3-D space have three translational and three rotational components. In the 1970s, when these types of operations became commonplace in computer graphics, together with perspective mapping as the final stage of visualization for human observers, so-called *"homogeneous coordinates"* were introduced [Roberts 1965, Blinn 1977]. They allow the representation of all transformations required by transformation matrices of size 4 by 4 with different entries. Special microprocessors have been developed in the 1970s allowing us to handle these operations efficiently. Extended concatenations of several sequential transformations turn out to be products of these matrices; to achieve real-time performance for realistic simulations with visual feedback and human operators in the loop, these operations have shaped computer graphics hardware design (computer generated images, CGI [Foley et al. 1990]).

2.1.1 Homogeneous Coordinate Transformations in 3-D Space

Instead of the Cartesian vector $\underline{\mathbf{r}}_C = (x, y, z)$, the homogeneous vector

$$r_h = (p \cdot x,\ p \cdot y,\ p \cdot z,\ p) \tag{2.1}$$

is used with p as a scaling parameter. The specification of a point in one coordinate system can be "transformed" into a description in a second coordinate system by three translations along the axes and three rotations around reference axes, some of which may not belong to any of the two (initial and final) coordinate systems.

2.1.1.1 Translations

This allows writing translations along all three axes by the amount $\Delta\underline{r} = (\Delta x, \Delta y, \Delta z)$ in the form of a matrix · vector multiplication with the *homogeneous transformation matrix (HTM)* for translation:

$$r_1 = \begin{pmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot r_0. \tag{2.2}$$

The three translation components shift the reference point for the rotated original coordinate system.

2.1.1.2 Rotations

Rotations around all axes may be described with the shorthand notation c = cos(angle) and s = sin(angle) by the corresponding HTMs:

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & s & 0 \\ 0 & -s & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix};\ R_y = \begin{pmatrix} c & 0 & -s & 0 \\ 0 & 1 & 0 & 0 \\ s & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix};\ R_z = \begin{pmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.3}$$

The position of the 1 on the main diagonal indicates the axis around which the rotation takes place.

The sequence of the rotations is of importance in 3-D space because the final result depends on it. Because of the dominant importance of gravity on Earth, the usual nomenclature for Euler angles (internationally standardized in mechanical engineering disciplines) requires the first rotation be around the gravity vector, defined as "heading angle" ψ (or pan angle for cameras). This reference system is dubbed the "geodetic coordinate system"; the x- and y-axes then are in the horizontal plane. The x-direction of this coordinate system (CS) may be selected as the main direction of motion or as the reference direction on a global scale (*e.g.*, magnetic North). The magnitude of rotation ψ is selected such that the x-axis of the rotated system comes to lie vertically underneath the x-axis of the new CS [*e.g.*, the body-fixed x-axis of the vehicle (x_O in Figure 2.3, upper right corner)]. As the second rotation, the turn angle of the vehicle's x-axis perpendicular to the horizontal plane has proven to be convenient. It is called "pitch angle" θ for vehicles (or tilt

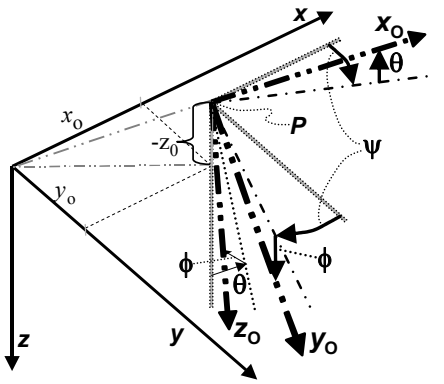

Figure 2.3. Transformation of a coordinate system

angle for cameras); the rotation takes place around an intermediate y-axis, called node axis k_y. This already yields the new x-direction x_O, around which the final rotation takes place: The roll– or bank angle ϕ indicates the angle around this axis between the plane of symmetry of the vehicle and the vertical plane. All of these angles are shown twice in the figure for easier identification of the individual axis of rotation.

2.1.1.3 Scaling

Due to Equation 2.1 scaling can be achieved simply by setting the last element in the HTM [lower right element p (4, 4)] different from 1. All components are then interpreted as scaled by the same factor p. This scaling is conveniently exploited by application to perspective mapping.

2.1.1.4 Perspective Mapping

Figure 2.4 shows some properties of perspective projection by a pinhole model. All points on a ray through the projection center P_p are mapped into a single point in the image plane at a distance f (the focal length) behind the plane $x_p = 0$. For example, the points Q_1, Q_2, and Q_3 are all mapped into the single point Q_i. This is to say that the 3-D depth to the point in the real world mapped is lost in the image. This is the major challenge for monocular vision. Therefore, the rectangle in the image plane Re_i may correspond both to the two rectangles Re_1 and Re_2 and to the trapezoids $Trap_1$ and $Trap_2$ at different ranges and with different orientations in the real world. Any four-sided polygon in space (also nonplanar ones) with the corner points on the four rays through the corners given will show up as the same (planar, rectangular) shape in the image.

To get rid of the sign changes in the image plane incurred by the projection center P_p (pinhole), the position of this plane is mirrored

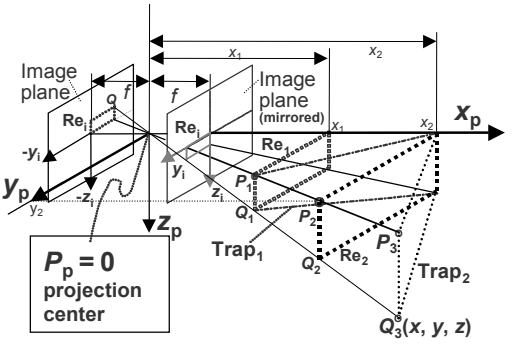

Figure 2.4. Perspective projection by a pinhole camera model

at $x_p = 0$ (as shown). This allows us to easily write down the mathematical relationship of perspective mapping when the x_p-axis is selected perpendicularly to the image plane and the origin is at the pin hole (projection center):

$$\begin{aligned} y_i \;/\; f &= y \;/\; x \\ z_i \;/\; f &= z \;/\; x, \end{aligned} \qquad \text{or} \qquad \begin{aligned} y_i &= y \;/\; (x \cdot (1 \;/\; f)) \\ z_i &= z \;/\; (x \cdot (1 \;/\; f)). \end{aligned} \tag{2.4}$$

This perspective projection equation may be included in the HTM–scheme by the projection matrix P (P_p for pixel coordinates). It has to be applied as the last matrix multiplication and yields (for each point-vector $\mathbf{x}_{Fk}$ in the real world) the "homogeneous" feature vector "e". The image coordinates y_i and z_i of a feature point are then obtained from the "homogeneous" feature vector e by dividing the second and third component resulting from Equation 2.4a by the fourth one (see Figure 2.5 for the coordinates):

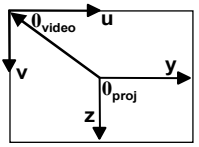

Figure 2.5. Image coordinates

$$P = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1/f & 0 & 0 & 0 \end{pmatrix}; \quad P_p = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & k_y & 0 & 0 \\ 0 & 0 & k_z & 0 \\ 1/f & 0 & 0 & 0 \end{pmatrix}. \tag{2.4a}$$

The image coordinates y_i and z_i of a feature point are then obtained from the "homogeneous" feature vector e by dividing the second and third component resulting from Equation 2.4a by the fourth one (see Figure 2.5 for the coordinates):

$$\begin{aligned} y_i &= (\text{second component } e_2)/(\text{fourth component } e_4) \\ z_i &= (\text{third component } e_3)/(\text{fourth component } e_4). \end{aligned} \tag{2.5}$$

The left matrix in Equation 2.4a leaves the y-component (within each image line) and the z-component (for each image line) as metric pixel coordinates.

The x-component has lost its meaning during projection and is used for scaling as indicated by Equation 2.4 and the last row of Equation 2.4a. If coordinates are to be given in pixel values in the image plane, the "1"s on the diagonal are replaced by k_y (element 2, 2) and k_z (element 3, 3) representing the number of pixels per unit length (Equation 2.4a right). Typical pixel sizes at present are 5 to 20 micrometer (approximately square), or k_y, $k_z \sim 200$ to 50 pixels per mm length.

After a shift of the origin from the center to the upper left corner (see Figure 2.5) this y-z sequence (now dubbed u-v) is convenient for the way pixels are digitized line by line by frame grabbers from the video signal. (For real-world applications, it has to be taken into account that frame-grabbing may introduce offsets in y- and z-directions, which lead to additive terms in the corresponding fourth column.)

2.1.1.5 Transformation of a Planar Road Scene into an Image

The set of HCTs needed for linking a simple road scene with corresponding features in an image is shown in Figure 2.6. The road is assumed to be planar, level and straight. The camera is somewhere above the road at an elevation H_c (usually

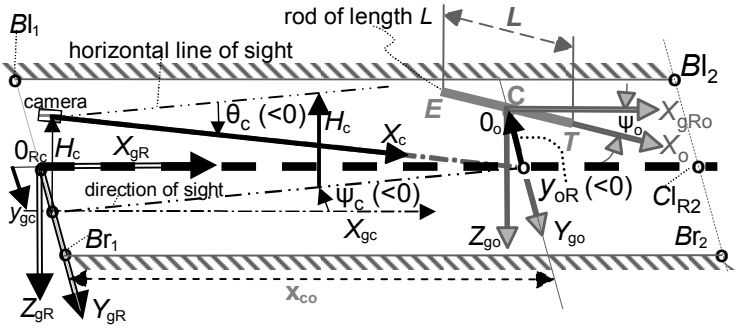

Figure 2.6. Coordinate systems for transforming a simple road scene into an image

known in vision tasks) and at a lateral offset y_{gc} (usually unknown) from the road centerline. The width B of the road is assumed to be constant in the look-ahead range; the marked centerline of the road partitions it into two equal parts. Some distance down the road there is a rod of length L as an obstacle to be detected. To simplify the task, it is assumed to be a one-dimensional object, easily describable in an object centered coordinate system to extend from $x_o = -L/2$ to $+L/2$. (The real-world object does have a cross section of some extension and shape that warrants treating it as an obstacle not to be driven over.) Relative to the road the rod does have a lateral position y_{oR} of its center point C from the centerline of the road and an orientation ψ_o between the object-fixed x-axis X_o and the tangent direction of the road X_{gRo}. Figure 2.6 shows the situation with the following CS:

1. X_o object-oriented and body-fixed in rod-direction (only one component).
2. Geodetic CS of the object (rod); geodetic CSs are defined with their X-Y-plane in the horizontal plane and their origin at the center of gravity of the object. The orientation of the x-axis in the horizontal plane is left open for a convenient choice in connection with the actual task. In this case, there is only one road direction, and therefore, X_{gRo} is selected as the reference for object orientation. There is only one rotation-angle ψ_o between the two CS 1 and 2 because gravity keeps the rod on the road surface (X_g - Y_g plane). The corresponding HTM is $R_{\psi o}$ [with a 1 in (3, 3) for rotation around the z-axis].
3. The road-centered geodetic CS (indexed "gR") at the longitudinal location of the camera has its origin at 0_{Rc} , and X_{gR} is directed along the road. Between the CS 2 and 3 there are (in the special case given here) the two translations x_{co} and y_{oR}. The corresponding HTM is T_{Ro} with entries x_{co} and y_{oR} in the last column of the first two rows.
4. The geodetic CS is at the projection center of the camera (not shown in Figure 2.6 for clarity); between the CS 3 and 4 there are again (in the special case given here) the two translations y_{gc} and H_c . The latter one is negative since z_g is posi-

tive downward. The corresponding HTM is T_{Rc} with entries y_{gc} and H_{c} in the last column of the second and third rows.

5. The camera-oriented CS (indexed "c"): The gaze direction of the camera is assumed to be fixed on the center of the road at the look-ahead distance of the object center. The elevation H_{c} of the camera above the ground and its lateral offset y_{gc} yield the camera pan (yaw) and tilt (pitch) angles ψ_{c} and Θ_{c}. The camera CS is obtained from CS 4 by two rotations: First, rotating around the Z_{gc} -axis (vertical line through camera projection center, not directly shown but indicated by the vector H_{c} in the opposite direction) by the amount ψ_{c} yields the horizontal direction of sight. The corresponding HCT–matrix is $R_{\psi\mathrm{c}}$. Now the X-Z-plane cuts the axis X_{gR} at distance x_{co}. Within this X-Z-plane now the intermediate x-axis has to be rotated until it also cuts the axis X_{gR} at distance x_{co} (pitch angle $-\Theta_{\mathrm{c}}$) and becomes X_{c}. The corresponding HTM is $R_{\Theta\mathrm{c}}$ [with a 1 in position (2, 2) for rotation around the intermediate y-axis].
6. The image CS into which the scene is now mapped by perspective projection.
7. The CS for the image matrix of pixel points in computer memory. During data acquisition and transfer, shifts may occur: By misalignments of a frame grabber, unintentional shifts of the image center may occur. Intentionally, the origin of the image CS may be shifted to the upper left corner of the image (see coordinates u, v in Figure 2.5).

Since all these transformations can be applied to only one point at a time, objects consisting of sequences of straight lines (so-called "polygons") have to be described by the ensemble of their corner points. This will be treated in Section 2.2. Note here that each object described in a certain CS has to be given by the ensemble of its corner points. For example, the rod is given by the two endpoints E and T on the X_{o} -axis at $-L/2$ and $+L/2$. The straight road is given by its left and right boundary lines Bl and Br, and by the centerline $C\mathrm{l}_{\mathrm{R}}$ in the road-CS. All three lines are realized by a straight line connection between two end points with indices 1 (left side of Figure 2.6) and 2 (right); all end points of lines defining the road lie at $z = 0$. The points on the left-hand side of the road are at $y = -B/2$, and those on the right-hand side are at $+ B/2$. The centerline $C\mathrm{l}_{\mathrm{R}}$ is at y = 0.

Let us consider the transformation of the endpoint T of the rod into the image taken by the camera according to Figure 2.6. In the 3-D homogeneous object CS of the rod, this point has the coordinate description $\mathbf{x_o}^{\mathbf{T}} = (\,\mathrm{L}/2,\, 0,\, 0,\, 1)$. After transition to the geodetic CS X_{go} , the state vector according to point 2 in the list above changes to

$$x_{go} = R_{\psi o} \cdot x_o \,. \tag{2.6}$$

To describe the point T in the road-oriented CS, the second HTM T_{Ro} for translation from C to 0_{Rc} according to point 3 above has to be applied:

$$x_{gR} = T_{Ro} \cdot R_{\psi o} \cdot x_o \,. \tag{2.7}$$

For the transition to the geodetic CS of the camera, multiplication by the HTM T_{Rc} with entries y_{gc} and H_{c} has to be performed according to point 4 above:

$$x_{gc} = T_{Rc} \cdot T_{Ro} \cdot R_{\psi o} \cdot x_o \,. \tag{2.8}$$

The two translations may be combined into a single matrix containing in the upper three rows of the fourth column the sum of the elements in the corresponding

rows of the HTMs in Equation 2.8. It has not been done here because the direction of the local road tangents would not be the same for a curved road. Therefore, before performing the second translation to the origin of the camera CS, a rotation around the vertical axis by the difference χ in local road direction would have to be inserted (yielding another rotation matrix $R_{\chi co}$ between T_{Rc} and T_{Ro}.

Now the two rotation-angles ψ_c and Θ_c for the optical axis of the camera have to be applied. Since ψ_c according to the definition has to be applied first, $R_{\psi c}$ has to stand to the right because this matrix will be encountered first when the column-vector $\mathbf{x_o}$ is multiplied from the right. This finally yields the state vector for the point T in camera coordinates:

$$\mathrm{x}_{co} = R_{\psi c} \cdot R_{\theta c} \cdot T_{Rc} \cdot T_{Ro} \cdot R_{\psi o} \cdot \mathrm{x}_o . \tag{2.9}$$

Applying perspective projection (Equation (2.4) to this 3-D-point $\mathbf{x_{co}}$ yields the "homogeneous" feature data for the image coordinates $\mathbf{e}$ according to Equation 2.5; note that the five HTMs contain the unknown variables of the vision task written below each matrix:

$$\begin{aligned} \mathrm{e} &= [\, P_p \cdot R_{\theta c} \cdot R_{\psi c} \cdot T_{Rc} \cdot T_{Ro} \cdot R_{\psi o}] \cdot \mathrm{x}_o = T_{tot} \cdot \mathrm{x}_o . \\ &\text{unknowns: } \theta_c, \psi_c, y_{gc}(x_{co}, y_{oR}) \, \psi_o \, ! \end{aligned} \tag{2.10}$$

The explicitly written down (transposed) form $(\mathbf{e})^T = (e_1, e_2, e_3, e_4)$ then yields the image coordinates

$$y_i = e_2 / e_4 \; ; \quad z_i = e_3 / e_4 . \tag{2.11}$$

The expression in square brackets is the same for any point to be transformed from object- into camera- coordinates. Therefore, in computer graphics, where all elements entering the HTMs are known beforehand, the so-called concatenated transformation matrix T_{tot} is computed as the product of all single HTMs once for each object and aspect condition. A single matrix-vector multiplication $T_{tot} \cdot \mathbf{x}_o$ then yields the position in homogeneous feature coordinates for the image of a point on the object at $\underline{\mathbf{x}}_o$ in object-centered coordinates. Equation 2.11 finally gives the image coordinates.

Note that these coordinates are real numbers, which means that the positions of the points mapped into the image are known to subpixel accuracy. If measurements of image features (Chapter 5) can be done to subpixel accuracy, too, the methods applied in recursive estimation (see Chapter 6) yield improved results by not rounding off feature coordinates to integer numbers (as is often done in a naive approach).

2.1.1.6 General Concatenations of HCTs; the Scene Tree

While the use of these transformation matrices in computer graphics is commonplace as a flexible tool for adaptation to new or modified tasks, in machine vision, until very recently, they have been exploited only during the formulation phase of the problem. Then, to be numerically more efficient on general-purpose processors, the resulting expressions of the matrix product T_{tot} have been hand-coded, initially. With the processing power available now, more easily adaptable codes become preferable; this is achieved by keeping each HTMs separate until numerical evaluation because in each matrix variables to be iterated may appear.

The challenge in machine vision as opposed to computer graphics is that some of the transformation parameters entering the matrices are not known beforehand but are the unknowns of the vision process, which have to be determined from image sequence analysis. Therefore, in each transformation, its sensitivity to small parameter changes has to be determined to compute the corresponding overall "Jacobian" matrices (JM, the first-order approximation for the nonlinear functional relationship describing the mapping of features on objects in the real world to those measured in the images). This rather compute-intensive operation and an efficient implementation will be discussed in Section 2.1.2.

The tendency toward separation of application-oriented aspects from those geared to the general methods of dynamic vision required a major change from the initial approach with respect to handling homogeneous coordinates. Concatenation is shifted to the evaluation of the scene model at runtime; then, both the nominal total HTM and the partial-derivative matrices for all unknown parameters and state variables are computed in conjunction (maybe numerically). This allows efficient use of intermediate results and makes the setup of new problems much easier for the user. The corresponding representation scheme for all objects and CSs in a so-called "scene tree" has been developed by D. Dickmanns (1997) and will be discussed in the following paragraphs.

Figure 2.7 without the shaded areas gives an example of a scene tree for describing the geometrical relations among several objects of relevance for the vision task shown in Figure 2.6 a single camera on a straight road. The nodes and edges in the shaded areas on the right-hand side and on top will be needed for the more general case of a camera onboard a vehicle moving on a curved road. In the straight road scene, the "object" represents the rod on the road at some look-ahead distance x_{co}; its lateral position on the road can be recovered in the image from the road boundaries nearby and from the vanishing point at the horizon (see Figure 2.8).The figure shows the resulting image, into which some labels for later image interpretation have been inserted.

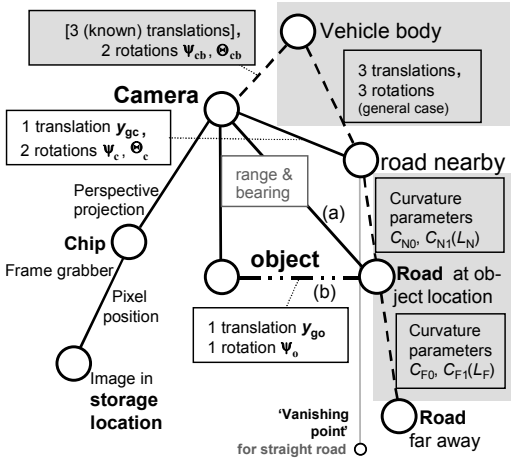

Figure 2.7. Scene tree for representing spatial relationships between objects seen and their image in perspective projection

For a horizontal straight road with parallel lines, the vanishing point, at which all parallel lines intersect, lies on the horizon line. Its distance components to the image center yield the direction of the optical axis: $-\psi_c$ to the direction of the road and $-\theta_c$ to the horizon line. The center of gravity (cg) of the rod has the averaged coordinates of the end points *E* and *T* in 3-D space.

The location of this point relative to the road centerline gives the lateral offset y_{oR} at the look-ahead distance x_{co} . The yaw angle ψ_o of the rod relative to the road can be determined by computing the difference in the positions of the end points *E* and *T*; however, the distortions from perspective mapping have to be taken into account. All these interpretations are handled automatically by the 4-D approach to dynamic vision (see Chapter 6). Since the camera looks from above almost tangentially to the plane containing the road, the distance in the real world increases with decreasing row index. If a certain geometric resolution for each pixel is required, there is a limit to the look-ahead range usable (shown on the left-hand side in Figure 2.8).

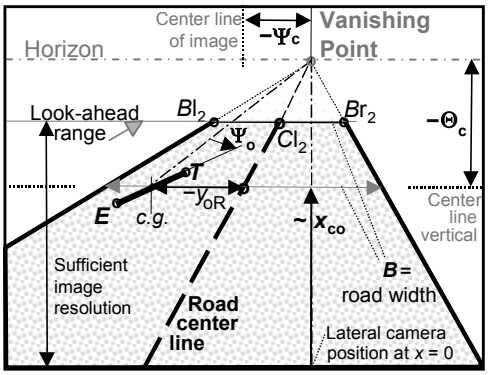

Figure 2.8. Image resulting from the scene given in Figure 2.6 after perspective mapping

For example, if each pixel is not allowed to cover more than 5 cm normal to the optical axis, the look-ahead range $L_{0.05}$ (in meters) or simply L_5 (in cm) is thus defined. This makes sense in road–scene analysis since lane markings, usually, are 10 to 50 cm wide and at least two pixels normal to a line are required for robust recognition under perturbations with edge feature extractors (Chapter 5).

Looking at sequences of images like these, the camera motion and the relative state of all objects of relevance for performing a driving mission have to be recognized sufficiently well and early with minimal time delay. The approach given in this book has proven to solve this problem reliably. Before the overall solution for precise and robust recognition can be discussed, all components needed have to be introduced first. Starting in Chapter 7, they will be applied together; the performance and complexity level will be open-ended for future growth.

Back to the scene tree: In Figure 2.7 each node represents an object in the real world (including virtual ones such as the CS at certain locations). The edges represent HCTs, *i.e.*, encodings of geometric relations. In combination with knowledge about the effects of these transformations, this allows a very compact description of all objects of relevance in the visual scene. Only the components of spatial state vectors and a few parameters of generic models for the objects are needed to represent the scene. The rest of the knowledge is coded in the object classes from which the objects hypothesized are generated.

The edges (a) and (b) at the center of Figure 2.7 (from the camera, respectively, from the "object" to the "road at the object location") are two alternative ways to determine where the object is. Edge (a) represents the case, where the bearing angles to some features of the road and to the object are interpreted separately; the road features need not necessarily be exactly at the location of the object. From these results, the location of the road and the lateral position of the object on the road can be derived indirectly in a second step. The difference in bearing angle to the road center at the range of the object yields the lateral position relative to the

road. In the case of edge (b), first only the range and bearing to the object are determined. Then at the position of the object, the features of the road are searched and measured, yielding directly the explicit lateral position of the object relative to the road. This latter procedure has yielded more stable results in recursive estimation under perturbations in vehicle pitch and yaw angles (see Chapter 6).

The sequence of edges in Figure 2.7 specifies the individual transformation steps; each node represents a coordinate system (frequently attached to a physical body) and each edge represents HCTs, generally implying several HTMs. The unknown parameters entering the HCTs are displayed in the boxes attached to the edge. At the bottom of each branch, the relevant object is represented in an object-centered coordinate system; this will be discussed in Section 2.2. A set of cameras (instead of a single one) may be included in the set of nodes making their handling schematic and rather easy. This will be discussed in connection with EMS vision later.

The additional nodes and edges in the shaded areas show how easily more detailed models may be introduced in the interpretation process. Figure 2.9 gives a sketch of the type of road scene represented by the full scene tree of Figure 2.7.

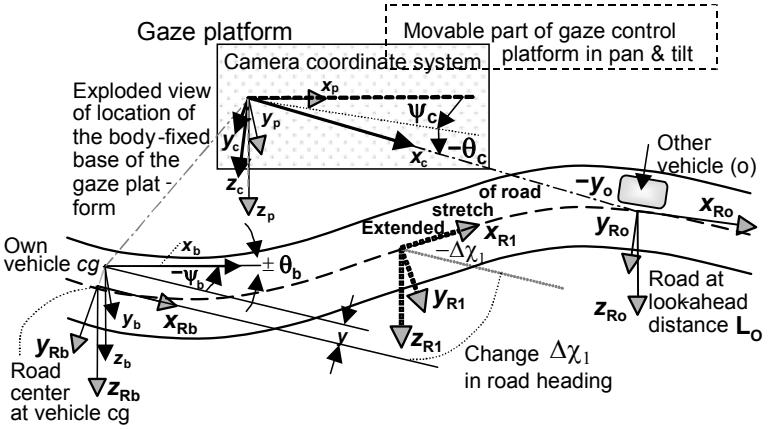

Figure 2.9. Coordinate systems for a general scene with own vehicle (index b) and one other vehicle (index o) on a curved road

Now, the position of the own vehicle relative to the road has to be determined. In the general case, these are three translational and three rotational components. Neglecting movements in bank angle (they average around 0) and in heave (vertical translation) and taking the longitudinal position as the moving origin of the vehicle CS, the same components as in the previous case have to be determined.

However, now the camera is located somewhere in the vehicle. The three translational components are usually fixed and do not change; the two rotational components from gaze control can be measured conventionally on the platform and are assumed known, error-free. So, there is no new unknown variable for active gaze control; however, the transformations corresponding to the known variables from mounting the platform on the vehicle have to be applied.

For the more general case of a curved road (shaded area to the right in Figure 2.7), the road models to be discussed in later sections have to be applied. They introduce several more unknowns into the vision process. However, using differential-geometry models minimizes the number of these terms; for planar roads, two sets of additional CSs allow large look-ahead ranges even with up to two inflection points of the road (changes of the sign of curvature; Figure 2.9 has just one).

General scheme of the scene tree: The example of a scene tree given above can be generalized for perspective mapping of many objects in the real world into images by several cameras. For practical reasons, one CS will be selected as the main reference; in vehicle guidance, this may be the geodetic CS linked to the center of gravity of the vehicle (or some easily definable one with similar advantages). This is called the "root node" and is drawn as the topmost node in standard notation.

The letter T shall designate all transformations for uniformity (both translations and rotations). The standard way of describing these transformations is from the leaves (bottom) to the root node. Therefore, when forming the total chain of transformations T_{tot} from features on objects in the real world into features in an image, denoted by K in Figure 2.10, the inverse transformation matrices T_{kj}^{-1} have to be used from the root to the leaves (left-hand side). A total transformation T_{tot} exists for each object-sensor pair, of which the object can be visually observed from the sensor. Once the scene tree has been defined for m cameras and n objects, the evaluation of the (at most $n \cdot m$) total transformation matrices is independent of the special task and can be coded as part of the general method [D. Dickmanns 1997].

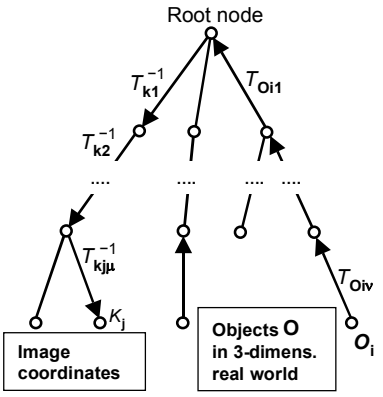

Figure 2.10. General scheme for object mapping in the scene graph

Since objects may appear and disappear during a mission, the perception system has to have the capability of autonomously inserting and deleting object branches in the scene tree. This object hypothesis generation and deletion capability is a crucial part of intelligent visual perception. Detailed discussions of various task domains will be given in later sections after the elements necessary for a flexible overall system have been introduced. Let the computation of T_{tot} be called the "traverse" of the scene graph. The recursive estimation method presented in Chapter 6 requires that this traverse is done not just once for each object-sensor pair but $(q + 1)$ times, if there are q unknown state variables and parameters entering the HTMs in T_{tot}. This model-based approach yields a first-order approximation (so-called "Jacobian matrices" or in short "Jacobians" of perspective mapping) describing the relationship between all model parameters and state components in the mentally represented world on the one hand, and feature positions in the images, on the other hand. Note that for 3-D models, there is also spatial information available in the Jacobians, allowing depth perception even with monocular vision (motion stereo). Because of this heavy

workload in computation, efficient evaluation of all these traverses is very important. This is the subject of the next section.

Again, all of this is independent of the special application, once the scene tree for the set of problems has been defined.

2.1.2 Jacobian Matrices for Concatenations of HTMs

To be flexible in coding and to reduce setup time for new task domains, the elements of Jacobian matrices may be determined by numerical differencing instead of fully analytical derivations; this is rather computer-intensive. An analytical derivation based on the factored matrices may be most computer-efficient. Figure 2.11 shows in the top row the total HTM for the homogeneous feature vector e of the example Equation 2.10 from road vehicle guidance with scene perception through an active camera. The two translations T have two and one unknown components here (maximally six are possible in total); all three rotation-angles have to be determined from vision as well. So there are six unknowns in the problem, for which the entries in the Jacobian matrix have to be computed.

To obtain these elements, nominal value and the systematically perturbed values due to changes in each unknown state variable or parameter $d\underline{\mathbf{x}}_i$ have to be computed for each feature point, just one at a time to obtain partial derivatives. If there are n state variables and q parameters to be iterated during recursive estimation, $(n + q + 1)$ total transformations have to be computed for each feature point. To be efficient, the nominal values should be exploited as much as possible also for computing the $n + q$ perturbed values. This procedure for the unknown state components is sketched in the sequel; adaptable parameters will be discussed in Section 2.2 and Chapter 6.

2.1.2.1 Partial Derivatives of Homogeneous Transformation Matrices

The overall transformation matrix for each feature point $\underline{\mathbf{x}}_{Fk}$ on the object (the rod in our case) in 3-D space was given by Equation 2.10:

$$\begin{aligned} \mathrm{e} &= [\, P_\mathrm{p} \cdot R_{\theta\mathrm{c}} \cdot R_{\psi\mathrm{c}} \cdot T_{\mathrm{Rc}} \cdot T_{\mathrm{Ro}} \cdot R_{\psi\mathrm{o}} \,] \cdot \mathrm{x}_\mathrm{o} = T_{\mathrm{tot}} \cdot \mathrm{x}_\mathrm{o}. \\ &\text{unknowns: } \theta_\mathrm{c}, \psi_\mathrm{c}, y_{\mathrm{gc}}(x_{\mathrm{co}}, y_{\mathrm{oR}})\ \psi_\mathrm{o}\ ! \end{aligned} \tag{2.10}$$

To describe the state of the rod relative to the camera, first its state relative to the road CS has to be specified by $(\psi_\mathrm{o}, x_{\mathrm{co}}, y_{\mathrm{oR}})$. Then the state of the camera relative to the road is given by $[y_{\mathrm{gc}}$, (the known elevation H_c), ψ_c, $\Theta_\mathrm{c}]$. Under the assumption of a planar straight road, the geometric arrangement of the objects "camera" and "rod" is fully described by specifying these "state components". The unknown state vector $\mathrm{x_S}$ of this problem with six components thus is

$$\mathrm{x_S^T} = (\psi_\mathrm{o},\ x_{\mathrm{co}},\ y_{\mathrm{oR}}, y_{\mathrm{gc}},\ \psi_\mathrm{c},\ \theta_\mathrm{c}). \tag{2.12}$$

These components have to be iterated during the vision process so that the underlying models yield a good fit to the visual features observed. In Equation 2.10 each R represents a single rotational and each T up to three translational compo-

nents, the unknowns of which are written below them. The total HTM including perspective mapping, designated by T_t is written

$$T_{tot} = P \cdot R_{\theta c} \cdot R_{\psi c} \cdot T_{Rc} \cdot T_{Ro} \cdot R_{\psi o}. \tag{2.13}$$

The partial derivative of T_t with respect to any component of the state vector $\underline{\mathbf{x}}_S$ (Equation 2.12) is characterized by the fact that each component enters just one of the HTMs and not the other ones. Applying the chain rule for derivatives of products such as $\partial T_t / \partial(x)$ yields zeros for all the other matrices, so that the overall derivative matrix, for example, with respect to y_{gc} (abbreviated here for simplicity by just x) entering the central HTM T_{Rc} is given by

$$\begin{aligned} \partial T_t / \partial(x) = T'_{tx} &= R_{\theta c} \cdot R_{\psi c} \cdot \underbrace{\partial T_{Rc} / \partial(x)}_{T'_{Rcx}} \cdot T_{Ro} \cdot R_{\psi o} \\ &= R_{\theta c} \cdot R_{\psi c} \cdot \quad T'_{Rcx} \quad \cdot T_{Ro} \cdot R_{\psi o}. \end{aligned} \tag{2.14}$$

Two cases have to be distinguished for the general partial derivatives of the HTMs containing the variables to be iterated: Translations and rotations.

Translations: These components are represented by the first three values in column 4 of HTMs. In addition to the nominal value for the transformation matrix T_N as given in Equation 2.2, also the partial derivative matrices for the unknown variables are computed in parallel. The full set of these matrices is given by

$$r_1 = \begin{pmatrix} 1 & 0 & 0 & \Delta r_x \\ 0 & 1 & 0 & \Delta r_y \\ 0 & 0 & 1 & \Delta r_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot r_0 = T_{rN} \cdot r_0; \quad \frac{\partial T_{Rc}}{\partial(\Delta r_x)} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = T'_{rx}, \tag{2.15 (a)}$$

$$\frac{\partial T_{Rc}}{\partial(\Delta r_y)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = T'_{ry} \quad ; \quad \frac{\partial T_{Rc}}{\partial(\Delta r_z)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} = T'_{rz}. \tag{2.15 (b)}$$

Rotations: According to Equation 2.3 the nominal HTMs for rotation (around the x-, y- and z- axis respectively) are repeated below: s stands for the sine and c for the cosine of the corresponding angle of rotation, say α.

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & s & 0 \\ 0 & -s & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; R_y = \begin{pmatrix} c & 0 & -s & 0 \\ 0 & 1 & 0 & 0 \\ s & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; R_z = \begin{pmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.3}$$

The partial derivatives of the transformation matrices $\partial R / \partial\alpha$, may be obtained from

$$\mathrm{d}(\sin\alpha)/d\alpha = \cos\alpha \; = c; \quad d(\cos\alpha)/d\alpha = -\sin\alpha \; = -s. \tag{2.16}$$

This leads to the derivative matrices for rotation

$$R'_{xx} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -s & c & 0 \\ 0 & -c & -s & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; R'_{yy} = \begin{pmatrix} -s & 0 & -c & 0 \\ 0 & 0 & 0 & 0 \\ c & 0 & -s & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; R'_{zz} = \begin{pmatrix} -s & c & 0 & 0 \\ -c & -s & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.17}$$

It is seen that exactly the same entries s and c as in the nominal case are required, but at different locations and some with different signs. The constant values 1 have disappeared, of course.

2.1.2.2 Concatenations and Efficient Computation Schemes

The overall transformation matrix for each point (see Equation 2.10) together with the concatenated derivative matrices for computing the Jacobian matrices are shown in Figure 2.11. It can be seen that many multiplications of matrices are the

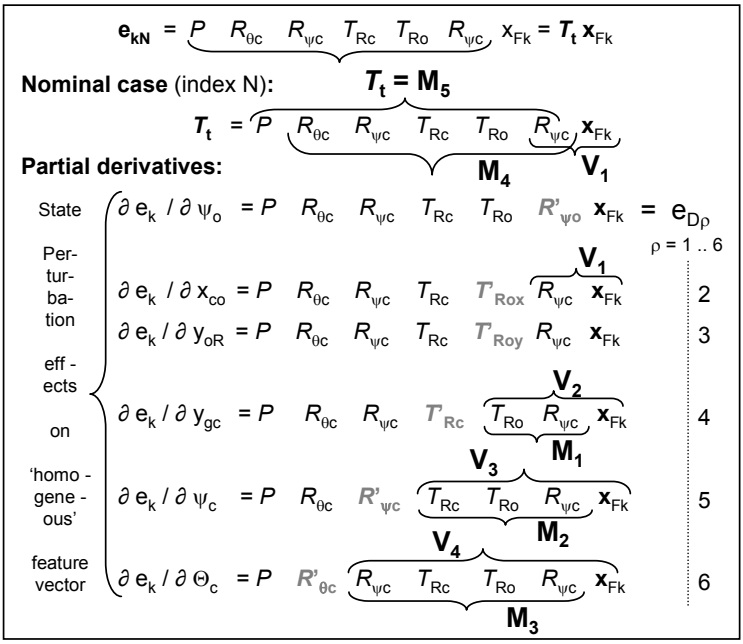

Figure 2.11. Scheme of matrix multiplication for efficient computation of concatenated homogeneous coordinate transformations (top) and elements of the Jacobian matrices

same for the nominal case and for the concatenated derivative matrices. Since the elements of the overall derivative matrix (Equation 2.14) are sparsely filled (Equation 2.15 and 2.17), let us first have a look at their matrix products for efficient coding.

The derivatives of the translation matrices all have a single "1" in the upper three rows of the last columns, the positions depend on the variable for partial derivation; the rest of the elements are zero, allowing efficient computation of the products. If such a matrix is multiplied from the right by another matrix, the multiplication just copies the last row of this matrix into the corresponding row of the product matrix where the 1 is in the derivative matrix (row 1 for x, 2 for y, and 3 for z). If such a matrix is multiplied to the left by another matrix, the multiplication just copies the ith column of this matrix into the last column of the product matrix. The index i designates the row, in which the 1 is in the derivative matrix (row 1 for x, 2 for y, and 3 for z). Note that the zeros in the first three columns lead to the effect that in all further matrix multiplications to the left, these three columns remain zero and need not be computed any longer. The significant column of the matrix product is the last one, filled in each row by the inner product of the row-vector

with the last column of the old (intermediate) product matrix. This analytical insight can save 75% of the computational steps for all matrices to the left if column vectors are used and matrix multiplication starts from the right-hand side.

The derivative matrices for rotational variables have four nonzero elements. These trigonometric functions have already been evaluated for the nominal case. The nonzero elements appear in such a pattern that working with sets of row– or column–vectors for matrices cuts in half the number of multiplications necessary for the elements of the product matrix.

As a final step toward image coordinates, the leftmost matrix P for perspective projection has to be applied. Note that this matrix product and the following scaling operations with element e_4 (Equation 2.11) yield two feature positions y and z in the image for each real-world feature. Thus, two Jacobian elements are also obtained for each image feature.

To get from the partial derivative of the total homogeneous transformation matrix $T'_{t\rho}$ to the correspondingly varied feature position in the image, this matrix has to be multiplied from the right-hand side by the 3-D feature vector $\mathbf{x}_{Fk}$ for the feature point (see Figure 2.11). This yields n vectors $e_{Dk\rho}$, $\rho = 1$ to n (6 in our case), each of which has four components. This is shown in the lower part of Figure 2.11. Multiplying these expressions by a finite variation in the state component $\delta x_{s\rho}$ results in the corresponding changes in the homogeneous feature vector:

$$\delta e_{\rho} = e_{D\rho} \cdot \delta x_{S\rho}. \tag{2.18}$$

The virtually displaced "homogeneous" feature position vector e (index p) around the nominal point (designated by index N) is computed from the "homogeneous" feature vectors e_N for the nominal case and δe_ρ from Equation 2.18.

$$\begin{aligned} e_{p\rho 2} &= e_{N2} + e_{D\rho 2} \cdot \delta x_{S\rho}; \quad e_{p\rho 3} = e_{N3} + e_{D\rho 3} \cdot \delta x_{S\rho}; \\ e_{p\rho 4} &= e_{N4} + e_{D\rho 4} \cdot \delta x_{S\rho}. \end{aligned} \tag{2.19}$$

Now the perturbed image feature positions after Equation 2.5 are

$$y_{p\rho} = e_{p\rho 2} / e_{p\rho 4} \; ; \quad z_{p\rho} = e_{p\rho 3} / e_{p\rho 4}. \tag{2.20}$$

Inserting the proper expressions from Equation 2.19 yields, with $e_{N2} / e_{N4} = y_{pN}$

$$\begin{aligned} y_{p\rho} &= (e_{N2} + e_{D\rho 2} \cdot \delta x_{S\rho}) / (e_{N4} + e_{D\rho 4} \cdot \delta x_{S\rho}) \\ &= y_{pN} \cdot [1 + e_{D\rho 2} \cdot \delta x_{S\rho} / e_{N2}] / [1 + e_{D\rho 4} \cdot \delta x_{S\rho} / e_{N4}]. \end{aligned} \tag{2.21}$$

Since the components of e_D contain unknown variations δx_ρ, a linear relationship between these unknowns and small variations in feature positions are sought. If $e_{D4} / e_{N4} << 1$, the ratio in Equation 2.21 can be approximated by

$$\begin{aligned} y_{p\rho} &= y_{pN} \cdot (1 + e_{D\rho 2} / e_{N2} \cdot \delta x_{S\rho}) \cdot (1 - e_{D\rho 4} / e_{N4} \cdot \delta x_{S\rho}) \\ &= y_{pN} \cdot [1 + (e_{D\rho 2} / e_{N2} - e_{D\rho 4} / e_{N4}) \cdot \delta x_{S\rho} \; (-) \\ &\qquad - (e_{D\rho 2} \cdot e_{D\rho 4}) / (e_{N2} \cdot e_{N4}) \cdot \delta x_{S\rho}^2]. \end{aligned} \tag{2.22}$$

Neglecting the last term with $\delta x^2 s_\rho$ as being at least one order of magnitude smaller than the linear term with $\delta x s_\rho$, a linear relationship between changes in y due to $\delta x s_\rho$ has been found

$$\delta y_{p\rho} = y_{p\rho} - y_{pN} \approx y_{pN} \cdot \; (e_{D\rho 2} / e_{N2} - e_{D\rho 4} / e_{N4}) \cdot \delta x_{S\rho}. \tag{2.23}$$

The element of the Jacobian matrix linked to the horizontal (y_i) feature at point x_{Fk} in the real world and to the unknown state variable xs_ρ now becomes

$$J_{k\rho y} = \partial y_k / \partial x_{S\rho} = \delta y_{p\rho} / \delta x_{S\rho} = y_{pN} \cdot (e_{D\rho 2} / e_{N2} - e_{D\rho 4} / e_{N4}). \quad (2.24)$$

The corresponding relation for the vertical feature position in the image is obtained in a similar way as

$$J_{k\rho z} = \partial z_k / \partial x_{S\rho} = \delta z_{p\rho} / \delta x_{S\rho} = z_{pN} \cdot (e_{D\rho 3} / e_{N3} - e_{D\rho 4} / e_{N4}). \quad (2.25)$$

This approach is a very flexible scheme for obtaining the entries into the Jacobian matrix efficiently. Adaptations to changing scene trees, due to new objects appearing with knew unknown states to be determined visually, can thus be made in an easy way.

The general approach discussed leaves two variants open to be selected for the actual case at hand:

1. *Very few feature points for an object:* In this case, it may be more economic with respect to computational load to multiply the sequence of transformations in Figure 2.11 from the left by the homogeneous 3-D feature point x_{Fk} (four components). This always requires only four inner vector products (= 25% of a matrix product). So, in total, for 6 matrix vector products, 24 inner products are needed; for the 7 expressions in Figure 2.11, a total of 168 such products result.
2. *Many feature points on an object:* Multiplying (concatenating) the elemental transformation matrices for the seven expressions in Figure 2.11 from right to left, in a naive approach requires at most $16 \cdot 5 \cdot 7 = 560$ inner vector products. For each feature point in the real world on a single object, $7 \cdot 4 = 28$ inner vector products have to be added to obtain the e-vector and its six partial derivatives. Asking for the number of features m on an object for which this approach is more economic as the one above, the relation $m \cdot 168 = 560 + m \cdot 28$ has to be solved for m as the break-even point, yielding $m = 560/140 = 4$.

So for more than four features on a single object, in our case with six unknowns in five transformation matrices plus perspective projection, the concatenation of transformation matrices first, and the multiplication with the coordinates of the feature points x_{Fk} afterward, is more computer-efficient.

Considering the fact that the derivative matrices are sparsely filled, as discussed above, and that many matrix products can be reused, frequently more than once, concatenation, performed as standard method in computer graphics, also becomes of interest in computer vision. However, as Figure 2.11 shows, much larger memory space has to be allotted for the iteration of transformation variables (the partial derivative matrices and their products). Note that to the left of derivative matrices of translations, also just a vector results for all further products, as in method 1 above. Taking advantage of all these points, method 2 is usually more efficient for more that two to three feature points on an object.

2.1.3 Time Representation

Time is considered an independent variable, monotonically increasing at a constant rate (as a good approximation to experience in the spatiotemporal domain of interest here). The temporal resolution required of measurement and control processes

depends on the application area; with humans as the main partner in dealing with the real world, their characteristic timescale will also predominate for the technical systems under investigation here.

Due to the fact that humans need at least 30 ms between two signals sensed, to be able to tell their correct sequence (independent of the sensory modality: tactile, auditory, or visual) [Pöppel et al. 1991; Pöppel, Schill 1995], this time-window of 30 ms is considered the "window of simultaneity". It is the basic temporal unit within which all signals are treated as simultaneous [Ruhnau 1994a, b]. This fact has also been the decisive factor in fixing the video frame rate. (More precisely, the subdivision into fields of interleaved odd and even lines and the reduced field rate by a factor of 2 was introduced to cheat human perception because of missing technological performance levels at the time of definition in the 1930s). This was done to achieve the impression of smooth analog motion for the observer, even though the fields are discrete and do represent jumps. When looking at field sequences of video signals from a static scene, taken at a large angular rate of the camera in the direction of image lines, a noticeable shift between frames can be observed. For precise interpretation and early detection of an onset of motion, therefore, the alternating fields at twice the frame rate (frequency of 50, respectively, 60 Hz) should be analyzed.

Since today's machine vision very often relies on the old standard video equipment, the basic cycle time for full images is adopted for dynamic machine vision and for control output. The sampling periods are 16 2/3 ms in the US (33 1/3 ms for full images or for each odd or even field) and 20 ms (40 ms) in Europe. The decision is justified by the fact that the corner frequency of human extremities for control actuation is about 2 Hz (arms and legs). In sampled control theory, a dozen samplings per period are considered sufficient to achieve analogue-like overall behavior. Therefore, constant control outputs over one video period are acceptable from this point of view. Note that the transition to fully digital image sensors in the near future will allow more freedom in the choice of frame rates.

Processes in the real world are described most compactly by relating temporal change rates of state variables to the values of the state variables, to the control variables involved, and to additional perturbations, which can hardly be modeled; these relations are called differential equations.

They can be transformed into difference equations according to sampled data theory with constant control output over the sampling period by numerical (or analytical) integration; perturbations will show up as added accumulated values with similar statistical properties, as in the analog case. The standard forms for (linearized) state transitions over a time period T are the state transition matrix $A(T)$ and the control effect matrix $B(T)$. $A(T)$ multiplied by the old state vector yields the homogeneous part of the new state vector; $B(T)$ describes the effect of constant unit control inputs onto the new state; multiplying $B(T)$ with the actual control output and adding this to the homogeneous part yields the new state.

Using this knowledge about motion processes of 3-D objects in 3-D space for image sequence interpretation is the core of the 4-D approach to dynamic vision developed by [Dickmanns, Wuensche 1987, 1999]. Combining temporal prediction with the first-order derivative matrix of perspective projection (the "Jacobian matrix" of spatial vision discussed in previous sections) allows bypassing perspective

inversion. Since each row of the Jacobian matrix contains the first-order sensitivity elements of the relation, how the feature measured depends on each state variable, spatial interpretation may become (at least partially) possible even with monocular vision, if either the object or the observer is moving. This will be discussed further down. Temporal embedding thus alleviates image interpretation despite the higher data rates. Temporal continuity conditions and attention control can counteract these higher data rates.

In addition, the eigenvalues of the transition matrix A represent characteristic time scales of the process. These and the frequency content of control inputs and of perturbations determine the temporal characteristics of the motion process.

In the framework of mission performance, other timescales may have special importance. The time needed for stabilizing image sequence interpretation is crucial for arriving at meaningful decisions based on this perception process. About a half second to one second are typical values for generating object hypotheses and having the transients settle down from poor initialization. Taking limited rates of change of state variables into account, preview (and thus prediction) times of several seconds seem to be reasonable in many cases. Total missions may last for several hours.

With respect to flawless functioning and maintenance of the vehicle's body, special timescales have to be observed, which an autonomous system should be aware of. All these aspects will be briefly discussed in the next section together with similar multiple scale problems in the spatial domain. To be flexible, an autonomous visual perception system should be capable of easy adjustment to temporal and spatial scales according to the task at hand.

2.1.4 Multiple Scales

The range of scales in the temporal and spatial domains needed to understand processes in the real world is very large. They are defined by typical sensor and mission dimensions as well as by the environmental conditions affecting both the sensors and the mission to be performed.

2.1.4.1 Multiple Space Scales

In the spatial domain, the size of the light sensitive elements in the sensor array may be considered the lower limit of immediate interest here. Typically, 5 to 20 micrometer (μm) is common today. Alternatively, as an underlying characteristic dimension, the typical width of the electronic circuitry may be chosen. This is about 0.1 to 2 μm, and this dimension characterizes the state of the art of microprocessors. Taking the 1-meter (m) scale as the standard, since this is the order of magnitude of typical body dimension of interest, the lower bound for spatial scales then is 10^{-7} m.

As the upper limit, the distance of the main light source on Earth, the orbital radius of the planet Earth circling the Sun (about 150 million km, that is $1.5 \cdot 10^{11}$ m), may be chosen with all other stars at infinity. The scale range of practical interest

thus is about 18 orders of magnitude. However, the different scales are not of equal and simultaneous interest.

Looking at the mapping conditions for perspective imaging, the 10^{-5} m range has immediate importance as the basic grid size of the sensor. For remote sensing of the environment, when the characteristic speed is in the order of magnitude of tens of m/s, several hundred meters may be considered a reasonable viewing range yielding about 3 to 10 seconds reaction time until the vehicle may reach the location inspected. If objects with a characteristic dimension of about 5 cm = 0.05 m should just fill a single pixel in the image when seen at the maximum distance of, say 200 m, the focal length f required is $f/10^{-5} = 200/0.05$ or $f = 0.04$ m or 40 mm. If one would like to have a 1 cm wide line mapped onto 2 pixel at 10 m distance (*e.g.*, to be able to read the letters on a license plate), the focal length needed is $f =$ 20 mm. To recognize lane markings 12 cm wide at 6m distance with 6 pixels on this width, a focal length of 3 mm would be sufficient. This shows that for practical purposes, focal lengths in the millimeter to decimeter range are adequate. This also happens to be the physical dimension of modern CCD-TV cameras.

Because the wheel diameters of typical vehicles are of the order of magnitude of about 1 m, objects become serious obstacles if their height exceeds about 0.1 m. Therefore, the 0.1 to 100 m range (typical look-ahead distance) is the most important and most frequently used one for ground vehicles. Entire missions, usually, measure 1 to 100 km in range. For air vehicles, several thousand km are typical travel distances since the Earth radius is about 6 371 km.

It may be interesting to note that the basic scale "1 m" was defined initially as 10^{-7} of one quarter of the circumference around the globe via both poles about 2 centuries ago.

Inverse use of multiple space scales in vision: In a visual scene, the same object may be a few meters or a few hundred meters away; the system should be able to recognize the object as the same unit independent of the distance viewed. To achieve this more easily, multiple focal lengths for a set of cameras will help since a larger focal length directly counteracts the downscaling of the image size due to increased range. This is the main reason for using multifocal camera arrangements in EMS vision. With a spacing of focal lengths by a factor of 4 (corresponding to the second pyramid stage each time), a numerical range of 16 may be bridged with three cameras.

In practical applications, a new object is most likely picked up in the wide field of view having least resolution. As few as four pixels (2 × 2) may be sufficient for detecting a new object reliably without being able to classify it. Performing a saccade to bring the object into the field of view of a camera with higher resolution, would result in suddenly having many pixels available, additionally. In a bifocal system with a focal length ratio of 4, the resolution would increase to 8 × 8 (64 pixels); in a trifocal system it would even go up to 32 × 32 (*i.e.*, 1K pixels) on the same area in the real world. Now the object may be analyzed on these scales in parallel. The coarse space scale may be sufficient for tracking the object with high temporal resolution up to video rate. On the high-resolution space scale, the object may then be analyzed with respect to its detailed shape, possibly on a lower time-

scale if computing power is limited. This approach has proven to be efficient and robust.

Even with this approach, to cover a range of possible object distances of two to three orders of magnitude, the size of objects in the images still varies over more than one order of magnitude; this fact has to be dealt with. Pyramid techniques [Burt *et al.* 1981] and multiple scales of feature extraction operators are used to achieve this. This requires that the same object be represented on different scales (with different spatial resolution and corresponding shape descriptors). Homogeneous coordinates allow representing different scales by just one parameter, the scaling factor. In the 4 × 4 transformation matrices, it enters at position (4, 4).

2.1.4.2 Multiple Timescales

On the time axis, the lower limit of resolution is considered the cycle time of electronic devices such as sensors and processors; it is presently in the 10^{-8} to 10^{-10} second (s) range. Typical process elements in digital computing such as message overheads for communication with other processing elements last of the order of magnitude of 0.1 to 1 ms; this also is a typical range of cycle times for conventional sensing as with inertial sensors.

The video cycle times mentioned above are the next characteristic timescale. Human reaction times are characterized by and may well be the reason for the 1−second basic timescale. Therefore, the 0.1 to 10 s scale ranges are the most important and most frequently used. "Maneuvers" as typical time histories of control outputs for achieving desired transitions from one regime of steady behavior to another last up to several minutes. Quasi-steady behaviors such as road-running on a highway or flying across an ocean may last several hours. Beyond this, the astronomically based scales of *days* and *years* predominate. One day means one revolution with respect to the sun around the Earth's axis; it has subdivisions into 24 hours of 60 minutes of 60 seconds each that is 86 400 seconds in total. One "year" means one revolution of Earth around the Sun and includes about 365 days. The corresponding lighting and climatic conditions (seasonal effects) affect the operation of vehicles in natural and man-made environments to a large degree.

The lifetimes of typical man-made objects are the order of 10 years (vehicles, sensors); human life expectancy is 5 to 10 decades. Objects encountered in the environment may be hundreds (trees, buildings, *etc.*) or many thousands of years of age (geological formations). Therefore, also in the temporal domain the range of scales of interest spans from 10^{-9} to about 10^{10} seconds or 19 orders of magnitude. Autonomous systems to be developed should have the capability of handling this range of scales as educated humans are able to do. In a single practical application, the actual range of interest is much lower, usually.

2.2 Objects

Beside background knowledge of the environmental conditions at some point or region on the globe and the variations over the seasons, most of our knowledge

about "the world" is affixed to object and subject classes. Of course, the ones of most importance are those one has to deal with most frequently in everyday life. Therefore, developing the sense of vision for road or air vehicles requires knowledge about objects and subjects encountered in these contexts most frequently; but also critical events which may put the achievement of mission goals at risk have to be known, even when their appearance is rather rare.

2.2.1 Generic 4-D Object Classes

The efficiency of the 4-D approach to dynamic vision is achieved by associating background knowledge about classes of objects and their behavioral capabilities with measurement data input. This knowledge is available in generic form, that is, structural information typical for object classes is fixed while specific parameters in the models have to be adapted to the special case at hand. Motion descriptions for the *c*enter of *g*ravity (the translational trajectory of the cg in space) forming the so-called "where"-problem, are separated from shape descriptions, called the "what"-problem. Typically, summing and averaging of feature positions is needed to solve the "where"-problem while differencing of feature positions contributes to solving the "what"-problem. In the approach chosen, the *"where"*-problem consists of finding the translational transformation parameters in the homogeneous transformations involved. The *"what"*-problem consists of finding an appropriate generic shape model and the best fitting shape and photometric parameters for this model after perspective projection (possibly including rotational degrees of freedom to account for the effects of aspect conditions).

As in computer graphics, all shape description is done in object-centered coordinates in 3-D, if possible, to take full advantage of a decoupled motion description relative to other objects.

2.2.2 Stationary Objects, Buildings

In road traffic, the road network and vegetation as well as buildings near the road are the stationary objects of most interest; roads will be discussed in Chapters 7 to 10. Highly visible large structures may be used as landmarks for orientation. The methods for shape representation are the same as for mobile objects; they do not need a motion description, however. These techniques are well known from computer graphics and are not treated here.

2.2.3 Mobile Objects in General

In this introductory chapter, only the basic ideas for object representation in the 4-D approach will be discussed. Detailed treatment of several object classes is done in connection with the application domain (Chapter 14). As far as possible, the methods used are taken from computer graphics to tap the large experience accumulated in that area. The major difference is that in computer vision, the actual

model both with respect to shape and to motion is not given but has to be inferred from the visual appearance in the image sequence. This makes the use of complex shape models with a large number of tesselated surface elements (*e.g.*, triangles) obsolete; instead, simple encasing shapes like rectangular boxes, cylinders, polyhedra, or convex hulls are preferred. Deviations from these idealized shapes such as rounded edges or corners are summarized in fuzzy symbolic statements (like "rounded") and are taken into account by avoiding measurement of features in these regions.

2.2.4 Shape and Feature Description

With respect to shape, objects and subjects are treated in the same fashion. Only rigid objects and objects consisting of several rigid parts linked by joints are treated here; for elastic and plastic modeling see, *e.g.*, [Metaxas, Terzepoulos 1993]. Since objects may be seen at different distances, the appearance in the image may vary considerably in size. At large distances, the 3-D shape of the object, usually, is of no importance to the observer, and the cross section seen contains most of the information for tracking. However, this cross section may depend on the angular aspect conditions; therefore, both coarse-to-fine and aspect-dependent modeling of shape is necessary for efficient dynamic vision. This will be discussed for simple rods and for the task of perceiving road vehicles as they appear in normal road traffic.

2.2.4.1 Rods

An idealized rod (like a geometric line) is an object with an extension in just one direction; the cross section is small compared to its length, ideally zero. To exist in the real 3-D world, there has to be matter in the second and third dimensions. The simplest shapes for the cross section in these dimensions are circles (yielding a thin cylinder for a constant radius along the main axis) and rectangles, with the square as a special case. Arbitrary cross sections and arbitrary changes along the main axis yield generalized cylinders, discussed in [Nevatia, Binford 1977] as a flexible generic 3-D-shape (sections of branches or twigs from trees may be modeled this way). In many parts of the world, these "sticks" are used for marking the road in winter when snow may eliminate the ordinary painted markings. With constant cross−sections as circles and triangles, they are often encountered in road traffic also: Poles carrying traffic signs (at about 2 m elevation above the ground) very often have circular cross sections. Special poles with cross sections as rounded triangles (often with reflecting glass inserts of different shapes and colors near the top at about 1 m) are in use for alleviating driving at night and under foggy conditions. Figure 2.12 shows some shapes of rods as used in road traffic. No matter what the shape, the rod will appear in an image as a line with intensity edges, in general. Depending on the shape of the cross section, different shading patterns may occur. Moving around a pole with cross section (b) or (c) at constant distance R, the width of the line will change; in case (c), the diagonals will yield maximum line width when looked at orthogonally.

Under certain lighting conditions, due to different reflection angles, the two sides potentially visible may appear at different intensity values; this allows recognizing the inner edge. However, this is not a stable feature for object recognition in the general case.

The length of the rod can be recognized only in the image directly when the angle between the optical axis and the main axis of the rod is known. In the special case where both axes are aligned, only the cross section as shown in (a) to (c) can be seen and rod length is not at all observable. When a rod is thrown by a human, usually, it has both translational and rotational velocity components. The rotation occurs around the center of gravity (marked in Figure 2.12), and rod length in the image will oscillate depending on the plane of rotation. In the special case where the plane of rotation contains the optical axis, just a growing and shrinking line appears. In all other cases, the tips of the rod describe an ellipse in the image plane (with different excentricities depending on the aspect conditions on the plane of rotation).

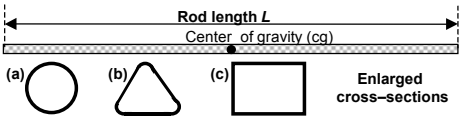

Figure 2.12. Rods with special applications in road traffic

2.2.4.2 Coarse-to-fine 2-D Shape Models

Seen from behind or from the front at a large distance, any road vehicle may be adequately described by its encasing rectangle. This is convenient since this shape has just two parameters, width B and height H. Precise absolute values of these parameters are of no importance at large distances; the proper scale may be inferred from other objects seen such as the road or lane width at that distance. Trucks (or buses) and cars can easily be distinguished. Experience in real-world traffic scenes tells us that even the upper boundary and thus the height of the object may be omitted without loss of functionality. Reflections in this spatially curved region of the car body together with varying environmental conditions may make reliable tracking of the upper boundary of the body very difficult. Thus, a simple U-shape of unit height (corresponding to about 1 m turned out to be practically viable) seems to be sufficient until 1 to 2 dozen pixels on a line cover the object in the image. Depending on the focal length used, this corresponds to different absolute distances.

Figure 2.13a shows this very simple shape model from straight ahead or exactly from the rear (no internal details). If the object in the image is large enough so that details may be distinguished reliably by feature extraction, a polygonal shape approximation of the contour as shown in Figure 2.13b or even with internal details (Figure 2.13c) may be chosen. In the latter case, area-based features such as the license plate, the dark

Figure 2.13. Coarse-to-fine shape model of a car in rear view: (a) encasing rectangle of width B (U-shape); (b) polygonal silhouette; (c) silhouette with internal structure

tires, or the groups of signal lights (usually in orange or reddish color) may allow more robust recognition and tracking.

2.2.4.3 Coarse-to-fine 3-D Shape Models

If multifocal vision allows tracking the silhouette of the entire object (*e.g.*, a vehicle) and of certain parts, a detailed measurement of tangent directions and curves may allow determining the curved contour. Modeling with Ferguson curves [Shirai 1987], "snakes" [Blake 1992], or linear curvature models easily derived from tangent directions at two points relative to the chord direction between those points [Dickmanns 1985] allows efficient piecewise representation. For vehicle guidance tasks, however, this will not add new functionality.

If the view onto the other car is from an oblique direction, the depth dimension (length of the vehicle) comes into play. Even with viewing conditions slightly off the axis of symmetry of the vehicle observed, the width of the car in the image will start increasing rapidly because of the larger length L of the body and due to the sine-effect in mapping.

Usually, it is very hard to determine the lateral aspect angle, body width B and length L simultaneously from visual measurements. Therefore, switching to the body diagonal D as a shape representation parameter has proven to be much more robust and reliable in real-world scenes [Schmid 1993]. Figure 2.14 shows the generic description for all types of rectangular boxes. For real objects with rounded shapes such as road vehicles, the encasing rectangle often is a sufficiently precise description for many purposes. More detailed shape descriptions with sub–objects (such as wheels, bumper, light groups, and license plate) and their appearance in the image due to specific aspect conditions will be discussed in connection with applications.

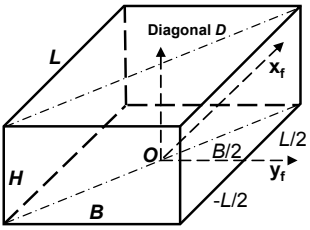

Figure 2.14. Object-centered representation of a generic box with dimension L, B, H; origin in center of ground plane

3-D models with different degrees of detail: Just for tracking and relative state estimation of cars, taking one of the vertical edges of the lower body and the lower bound of the object into account has proven sufficient in many cases [Thomanek 1992, 1994, 1996]. This, of course, is domain specific knowledge, which has to be introduced when specifying the features for measurement in the shape model. In general, modeling of highly measurable features for object recognition has to depend on aspect conditions.

Similar to the 2-D rear silhouette, different models may also be used for 3-D shape. Figure 2.13a corresponds directly to Figure 2.14 when seen from behind. The encasing box is a coarse generic model for objects with mainly perpendicular surfaces. If these surfaces can be easily distinguished in the image and their separation line may be measured precisely, good estimates of the overall body dimen-

sions can be obtained for oblique aspect conditions even from relatively small image sizes. The top part of a truck and trailer frequently satisfies these conditions.

Polyhedral 3-D shape models with 12 independent shape parameters (see Figure 2.15 for four orthonormal projections as frequently used in engineering) have been investigated for road vehicle recognition [Schick 1992]. By specializing these parameters within certain ranges, different types of road vehicles such as cars, trucks, buses, vans, pickups, coupes, and sedans may be approximated sufficiently well for recognition [Schick, Dickmanns 1991; Schick 1992; Schmid 1993]. With these models, edge measurements should be confined to vehicle regions with small curvatures, avoiding the idealized sharp 3-D edges and corners of the generic model.

Aspect graphs for simplifying models and visibility of features: In Figure 2.15, the top-down the side view and the frontal and rear views of the polygonal model are given. It is seen that the same 3-D object may look completely different in these special cases of aspect conditions. Depending on them, some features may be visible or not. In the more general case with oblique viewing directions, combined features from the views shown may be visible. All aspect conditions that allow seeing the same set of features (reliably) are collected into one class. For a rectangular box on a plane and the camera at a fixed elevation above the ground, there are eight such aspect classes (see Figures 2.15 and 2.16): Straight from the front, from each side, from the rear, and an additional four from oblique views. Each can contain features from two neighboring groups.

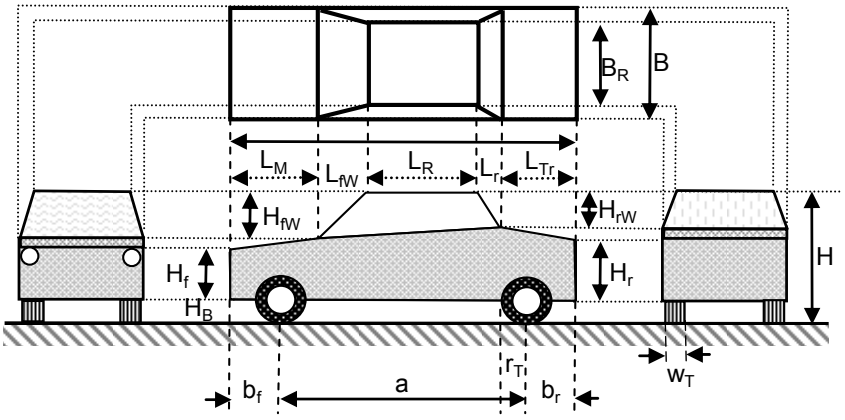

Figure 2.15. More detailed (idealized) generic shape model for road vehicles of type "car" [Schick 1992]

Due to this fact, a single 3-D model for unique (forward perspective) shape representation has to be accompanied by a set of classes of aspect conditions, each class containing the same set of highly visible features. These allow us to infer the presence of an object corresponding to this model from a collection of features in the image (inverse 3-D shape recognition including rough aspect conditions, or – in short – "hypothesis generation in 3-D").

This difficult task has to be solved in the initialization phase. Within each class of aspect conditions hypothesized, in addition, good initial estimates of the relevant state variables and parameters for recursive iteration have to be inferred from the relative distribution of features. Figure 2.16 shows the features for a typical car; for each vehicle class shown at the top, the lower part has special content.

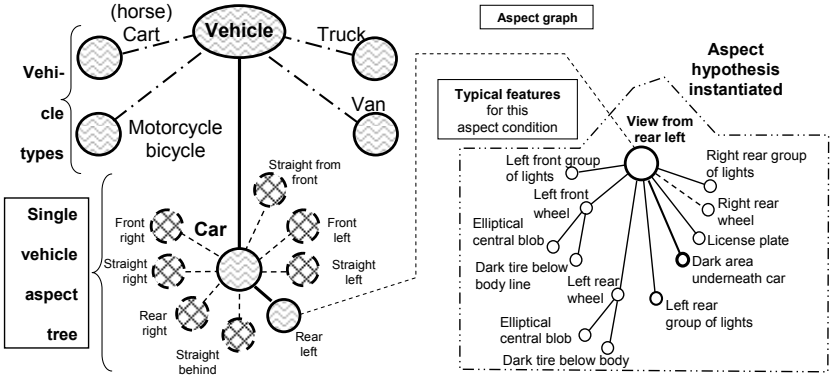

Figure 2.16. Vehicle types, aspect conditions, and feature distributions for recognition and classification of vehicles in road scenes

In Figure 2.17, a sequence of appearances of a car is shown driving in simulation on an oval course. The car is tracked from some distance by a stationary camera with gaze control that keeps the car always in the center of the image; this is called fixation-type vision and is assumed to function ideally in this simulation, *i.e.*, without any error).

The figure shows but a few snapshots of a steadily moving vehicle with sharp edges in simulation. The actual aspect conditions are computed according to a motion model and graphically displayed on a screen, in front of which a camera observes the motion process. To be able to associate the actual image interpretation with the results of previous measurements, a motion model is necessary in the analysis process also, constraining the actual motion in 3-D; in simulation, of course, the generic dynamical model is the same as in simulation. However, the actual control input is unknown and has to be reconstructed from the trajectory driven and observed (see Section 14.6.1).

2.2.5 Representation of Motion

The laws and characteristic parameters describing motion behavior of an object or a subject along the fourth dimension, time, are the equivalent to object shape representations in 3-D space. At first glance, it might seem that pixel position in the image plane does not depend on the actual speed components in space but only on the actual position. For one time this is true; however, since one wants to understand 3-D motion in a temporally deeper fashion, there are at least two points requiring modeling of temporal aspects:

1. Recursive estimation as used in this approach starts from the values of the state variables predicted for the next time of measurement taking.
2. Deeper understanding of temporal processes results from having representational terms available describing these processes or typical parts thereof in symbolic form, together with expectations of motion behavior over certain timescales.

A typical example is the maneuver of lane changing. Being able to recognize these types of maneuvers provides more certainty about the correctness of the perception process. Since everything in vision has to be hypothesized from scratch, recognition of processes on different scales simultaneously helps building trust in the hypotheses pursued. Figure 2.17 may have been the first result from hardware-in-the-loop simulation where a technical vision system has determined the input

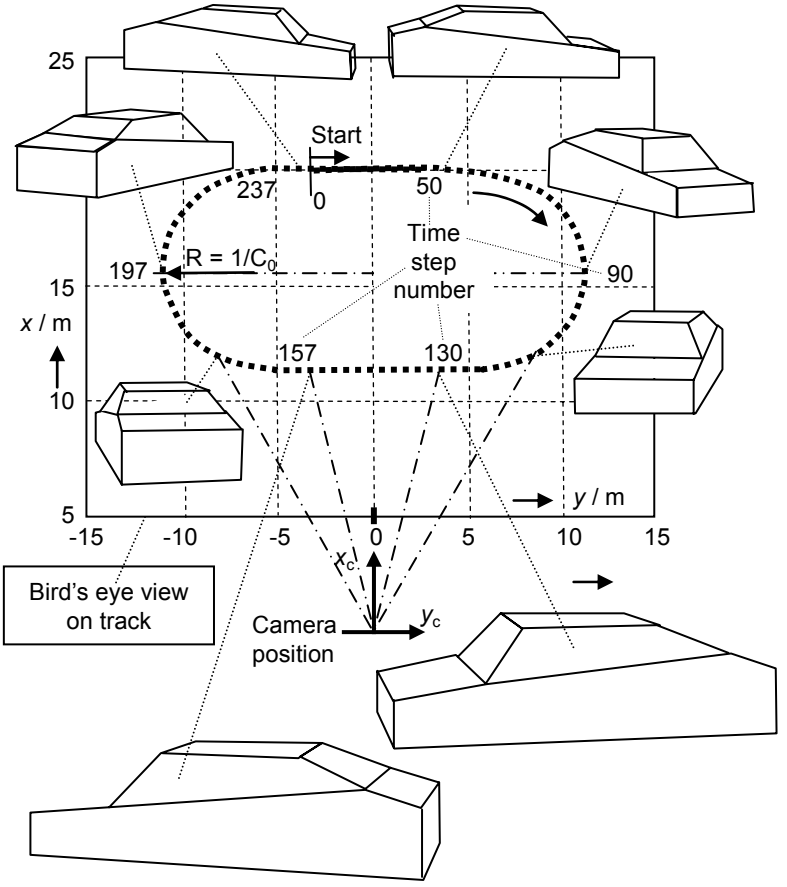

Figure 2.17. Changing aspect conditions and edge feature distributions while a simulated vehicle drives on an oval track with gaze fixation (smooth visual pursuit) by a stationary camera. Due to continuity conditions in 3-D space and time, "catastrophic events" like feature appearance/disappearance can be handled easily.

control time history for a moving car from just the trajectory observed, but, of course, with a motion model "in mind" (see Section 14.6.1).

The translations of the center of gravity (cg) and the rotations around this cg describe the motion of objects. For articulated objects also, the relative motion of the components has to be represented. Usually, the modeling step for object motion results in a (nonlinear) system of n differential equations of first order with n state components $\underline{X}$, q (constant) parameters $\underline{p}$ and r control components $\underline{U}$ (for subjects see Chapter 3).

2.2.5.1 Definition of State and Control Variables

A set of

- *State variables* is a collection of variables for describing temporal processes, which allows decoupling future developments from the past. State variables cannot be changed at one time. (This is quite different from "states" in computer science or automaton theory. Therefore, to accentuate this difference, sometimes use will be made of the terms *s-state* for systems dynamics states and *a-state* for automaton-state to clarify the exact meaning.) The same process may be described by different state variables, like Cartesian or polar coordinates for positions and their time derivatives for speeds. Mixed descriptions are possible and sometimes advantageous. The minimum number of variables required to completely decouple future developments from the past is called the order n of the system. Note that because of the second-order relationship between forces or moments and the corresponding temporal changes according to Newton's law, velocity components are state variables.
- *Control variables* are those variables in a dynamic system, that may be changed at each time "at will". There may be any kind of discontinuity; however, very frequently control time histories are smooth with a few points of discontinuity when certain events occur.

Differential equations describe constraints on temporal changes in the system. Standard forms are n equations of first order ("state equations") or an n-th order system, usually given as a transfer function of nth order for linear systems. There are an infinite variety of (usually nonlinear) differential equations for describing the same temporal process. System parameters $\underline{p}$ allow us to adapt the representation to a class of problems

$$dX / dt = f(X, p, t) . \tag{2.26}$$

Since real-time performance, usually, requires short cycle times for control, linearization of the equations of motion around a nominal set point (index N) is sufficiently representative of the process if the set point is adjusted along the trajectory. With the substitution

$$X = X_N + x , \tag{2.27}$$

one obtains

$$dX / dt = dX_N / dt + dx / dt . \tag{2.28}$$

The resulting sets of differential equations then are for the nominal trajectory:

$$dX_N / dt = f(X_N, p, t) ; \tag{2.29}$$

for the linearized perturbation system follows:

$$dx/dt = F \cdot x + v'(t)\,, \tag{2.30}$$

with

$$F = df/dX_N \tag{2.31}$$

as an (n × n)-matrix and $\underline{v}'(t)$ an additive noise term.

2.2.5.2 Transition Matrices for Single Step Predictions

Equation 2.30 with matrix F may be transformed into a difference equation with cycle time T for grid point spacing by one of the standard methods in systems dynamics or control engineering. (Precise numerical integration from 0 to T for $v = 0$ may be the most convenient one for complex right–hand sides.) The resulting general form then is

$$\begin{aligned} &x[(k+1)T] = A \cdot x[kT] + v[kT] \\ &\text{or in short-hand} \qquad x_{k+1} = A \cdot x_k + v_k\,, \end{aligned} \tag{2.32}$$

with matrix A of the same dimension as F. In the general case of local linearization, all entries of this matrix may depend on the nominal state variables. Procedures for computing the elements of matrix A from F have to be part of the 4-D knowledge base for the application at hand.

For objects, the trajectory is fixed by the initial conditions and the perturbations encountered. For subjects having additional control terms in these equations, determination of the actual control output may be a rather involved procedure. The wide variety of subjects is discussed in Chapter 3.

2.2.5.3 Basic Dynamic Model: Decoupled Newtonian Motion

The most simple and yet realistic dynamic model for the motion of a rigid body under external forces F_e is the Newtonian law

$$d^2x/dt^2 = \Sigma F_e(t)/m\,. \tag{2.33}$$

With unknown forces, colored noise $\underline{v(t)}$ is assumed, and the right–hand side is approximated by first–order linear dynamics (with time constant $T_C = 1/\alpha$ for acceleration a). This general third-order model for each degree of freedom may be written in standard state space form [BarShalom, Fortmann 1988]

$$d/dt \begin{pmatrix} x \\ V \\ a \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\alpha \end{pmatrix}}_{F} \cdot \begin{pmatrix} x \\ V \\ a \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}}_{g} \cdot v(t). \tag{2.34}$$

For the corresponding discrete formulation with sampling period T and $e^{-\alpha T} = \gamma$, the transition matrix $\underline{A}$ becomes

$$A = \begin{pmatrix} 1 & T & [T/\alpha - (1-\gamma)/\alpha] \\ 0 & 1 & (1-\gamma)/\alpha \\ 0 & 0 & \gamma \end{pmatrix}; \; for \; \alpha = 0 : A = \begin{pmatrix} 1 & T & T/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.35}$$

The perturbation input vector is modeled by

$$b_k \cdot v_k \quad \text{with} \quad b_k^T = [T^2/2,\; T,\; 1], \tag{2.36}$$

which yields the discrete model

$$x_{k+1} = A \cdot x_k + b_k \cdot v_k \,. \tag{2.37}$$

The value of the expectation is $\mathrm{E}[v_k] = 0$, and the variance is $\mathrm{E}[v_k^2] = \sigma_q^2$ (essential for filter tuning).The covariance matrix Q for process noise is given by

$$Q = b_k \cdot \sigma_q^2 \cdot b_k^T = \begin{pmatrix} T^4/4 & T^3/2 & T^2/2 \\ T^3/2 & T^2 & T \\ T^2/2 & T & 1 \end{pmatrix} \cdot \sigma_q^2. \tag{2.38}$$

This model may be used independently in all six degrees of freedom as a default model if no more specific knowledge is given.

2.3 Points of Discontinuity in Time

The aspects discussed above for smooth parts of a mission with nice continuity conditions alleviate perception; however, sudden changes in behavior are possible, and sticking to the previous mode of interpretation would lead to disaster.

Efficient dynamic vision systems have to take advantage of continuity conditions as long as they prevail; however, they always have to watch out for discontinuities in object motion observed to adjust readily. For example, a ball flying on an approximately parabolic trajectory through the air can be tracked efficiently using a simple motion model. However, when the ball hits a wall or the ground, elastic reflection yields an instantaneous discontinuity of some trajectory parameters, which can nonetheless be predicted by a different model for the motion event of reflection. So the vision process for tracking the ball has two distinctive phases which should be discovered in parallel to the primary vision task.

2.3.1 Smooth Evolution of a Trajectory

Flight phases (or in the more general case, smooth phases of a dynamic process) in a homogeneous medium without special events can be tracked by continuity models and low-pass filtering components (like Section 2.2.5.3). Measurement values with oscillations of high frequency are considered to be due to noise; they have to be eliminated in the interpretation process. The natural sciences and engineering have compiled a wealth of models for different domains. The least-squares error model fit has proven very efficient both for batch processing and for recursive estimation. Gauss [1809] opened up a new era in understanding and fitting motion processes when he introduced this approach in astronomy. He first did this with the solution curves (ellipses) for the differential equations describing planetary motion.

Kalman [1960] derived a *recursive formulation* using *differential models* for the motion process when the statistical properties of error distributions are known. These algorithms have proven very efficient in space flight and many other applications. Meissner, Dickmanns [1983]; Wuensche [1987] and Dickmanns [1987] extended this approach to perspective projection of motion processes described in physical

space; this brought about a quantum leap in the performance capabilities of real-time computer vision. These methods will be discussed for road vehicle applications in later sections.

2.3.2 Sudden Changes and Discontinuities

The optimal settings of parameters for smooth pursuit lead to unsatisfactory tracking performance in case of sudden changes. The onset of a harsh braking maneuver of a car or a sudden turn may lead to loss of tracking or at least to a strong transient motion estimated. If the onsets of these discontinuities can be predicted, a switch in model or tracking parameters at the right moment will yield much better results. For a bouncing ball, the moment of discontinuity can easily be predicted by the time of impact on the ground or wall. By just switching the sign of the angle of incidence relative to the normal of the reflecting surface and probably decreasing speed by some percentage, a new section of a smooth trajectory can be started with very likely initial conditions. Iteration will settle much sooner on the new, smooth trajectory arc than by continuing with the old model disregarding the discontinuity (if this recovers at all).

In road traffic, the compulsory introduction of the braking (stop) lights serves the same purpose of indicating that there is a sudden change in the underlying behavioral mode (deceleration), which can otherwise be noticed only from integrated variables such as speed and distance. The pitching motion of a car when the brakes are applied also gives a good indication of a discontinuity in longitudinal motion; it is, however, much harder to observe than braking lights in a strong red color.

Conclusion:

> As a general scheme in vision, it can be concluded that partially smooth sections and local discontinuities have to be recognized and treated with proper methods both in the 2-D image plane (object boundaries) and on the time line (events).

2.4 Spatiotemporal Embedding and First-order Approximations

After the rather lengthy excursion to object modeling and how to embed temporal aspects of visual perception into the recursive estimation approach, the overall vision task will be reconsidered in this section. Figure 2.7 gave a schematic survey of the way features at the surface of objects in the real 3-D world are transformed into features in an image by a properly defined sequence of "homogeneous coordinate transformations" (HCTs). This is easily understood for a static scene.

To understand a dynamically changing scene from an image sequence taken by a camera on a moving platform, the temporal changes in the arrangements of objects also have to be grasped by a description of the motion processes involved.

Therefore, the general task of real-time vision is to achieve a compact internal representation of motion processes of several objects observed in parallel by evaluating feature flows in the image sequence. Since egomotion also enters the content of images, the state of the vehicle carrying the cameras has to be observed simultaneously. However, vision gives information on *relative* motion only between objects, unfortunately, in addition, with appreciable time delay (several tenths of a second) and no immediate correlation to inertial space. Therefore, conventional sensors on the body yielding relative motion to the stationary environment (like odometers) or inertial accelerations and rotational rates (from inertial sensors like accelerometers and angular rate sensors) are very valuable for perceiving egomotion and for telling this apart from the visual effects of motion of other objects. Inertial sensors have the additional advantage of picking up perturbation effects from the environment before they show up as unexpected deviations in the integrals (speed components and pose changes). All these measurements with differing delay times and trust values have to be interpreted in conjunction to arrive at a consistent interpretation of the *situation* for making decisions on appropriate behavior.

Before this can be achieved, perceptual and behavioral capabilities have to be defined and represented (Chapters 3 to 6). Road recognition as indicated in Figures 2.7 and 2.9 while driving on the road will be the application area in Chapters 7 to 10. The approach is similar to the human one: Driven by the optical input from the image sequence, an internal *animation process in 3-D space and time* is started with members of generically known object and subject classes that are to duplicate the visual appearance of "the world" by prediction-error feedback. For the next time for measurement taking (corrected for time delay effects), the expected values in each measurement modality are predicted. The prediction errors are then used to improve the internal state representation, taking the Jacobian matrices and the confidence in the models for the motion processes as well as for the measurement processes involved into account (error covariance matrices).

For vision, the concatenation process with HCTs for each object-sensor pair (Figure 2.7) as part of the physical world provides the means for achieving our goal of understanding dynamic processes in an integrated approach. Since the analysis of the next image of a sequence should take advantage of all information collected up to this time, temporal prediction is performed based on the actual best estimates available for all objects involved and based on the dynamic models as discussed. Note that no storage of image data is required in this approach, but only the parameters and state variables of those objects instantiated need be stored to represent the scene observed; usually, this reduces storage requirements by several orders of magnitude.

Figure 2.9 showed a road scene with one vehicle on a curved road (upper right) in the viewing range of the egovehicle (left); the connecting object is the curved road with several lanes, in general. The mounting conditions for the camera in the vehicle (lower left) on a platform are shown in an exploded view on top for clarity. The coordinate systems define the different locations and aspect conditions for object mapping. The trouble in vision (as opposed to computer graphics) is that the entries in most of the HCT-matrices are the unknowns of the vision problem (relative distances and angles). In a tree representation of this arrangement of objects (Figure 2.7), each edge between circles represents an HCT and each node (circle)

represents an object or sub–object as a movable or functionally separate part. Objects may be inserted or deleted from one frame to the next (dynamic scene tree).

This scene tree represents the mapping process of features on the surface of objects in the real world up to hundreds of meters away into the image of one or more camera(s). They finally have an extension of several pixels on the camera chip (a few dozen micrometers with today's technology). Their motion on the chip is to be interpreted as body motion in the real world of the object carrying these features, taking body motion affecting the mapping process properly into account. Since body motions are smooth, in general, spatiotemporal embedding and first-order approximations help making visual interpretation more efficient, especially at high image rates as in video sequences.

2.4.1 Gain by Multiple Images in Space and/or Time for Model Fitting

High–frequency temporal embedding alleviates the correspondence problem between features from one frame to the next, since they will have moved only by a small amount. This reduces the search range in a top-down feature extraction mode like the one used for tracking. Especially, if there are stronger, unpredictable perturbations, their effect on feature position is minimized by frequent measurements. Doubling the sampling rate, for example, allows detecting a perturbation onset much earlier (on average). Since tracking in the image has to be done in two dimensions, the search area may be reduced by a square effect relative to the one-dimensional (linear) reduction in time available for evaluation. As mentioned previously for reference, humans cannot tell the correct sequence of two events if they are less than 30 ms apart, even though they can perceive that there are two separate events [Pöppel, Schill 1995]. Experimental experience with technical vision systems has shown that using every frame of a 25 Hz image sequence (40 ms cycle time) allows object tracking of high quality if proper feature extraction algorithms to subpixel accuracy and well-tuned recursive estimation processes are applied. This tuning has to be adapted by knowledge components taking the situation of driving a vehicle and the lighting conditions into account.

This does not include, however, that all processing on the higher levels has to stick to this high rate. Maneuver recognition of other subjects, situation assessment, and behavior decision for locomotion can be performed on a (much) lower scale without sacrificing quality of performance, in general. This may partly be due to the biological nature of humans. It is almost impossible for humans to react in less than several hundred milliseconds response time. As mentioned before, the unit "second" may have been chosen as the basic timescale for this reason.

However, high image rates provide the opportunity both for early detection of events and for data smoothing on the timescale with regard to motion processes of interest. Human extremities like arms or legs can hardly be activated at more than 2 Hz corner frequency. Therefore, efficient vision systems should concentrate computing resources to where information can be gained best (at expected feature locations of known objects/subjects of interest) and to regions where new objects may occur. Foveal–peripheral differentiation of spatial resolution in connection with fast gaze control may be considered an optimal vision system design found in

nature, if a corresponding management system for gaze control, knowledge application and interpretation of multiple, piecewise smooth image sequences is available.

2.4.2 Role of Jacobian Matrix in the 4-D Approach to Dynamic Vision

It is in connection with 4-D spatiotemporal motion models that the sensitivity matrix of perspective feature mapping gains especial importance. The dynamic models for motion in 3-D space link feature positions from one time to the next. Contrary to perspective mapping in a single image (in which depth information is completely lost), the partial first-order derivatives of each feature with respect to all variables affecting its appearance in the image do contain spatial information. Therefore, linking the temporal motion process in 4-D with this physically meaningful Jacobian matrix has brought about a quantum leap in visual dynamic scene understanding [Dickmanns, Meissner 1983, Wünsche 1987, Dickmanns 1987, Dickmanns, Graefe 1988, Dickmanns, Wuensche 1999]. This approach is fundamentally different from applying some (arbitrary) motion model to features or objects in the image plane as has been tried many times before and after 1987. It was surprising to learn from a literature review in the late 1990s that about 80 % of so-called Kalman–filter applications in vision did not take advantage of the powerful information available in the Jacobian matrices when these are determined, *including egomotion and the perspective mapping process*.

The nonchalance of applying Kalman filtering in the image plane has led to the rumor of brittleness of this approach. It tends to break down when some of the (unspoken) assumptions are not valid. Disappearance of features by self-occlusion has been termed a *catastrophic event*. On the contrary, Wünsche [1986] was able to show that not only temporal predictions in 3-D space were able to handle this situation easily, but also that it is possible to determine a limited set of features allowing optimal estimation results. This can be achieved with relatively little additional effort exploiting information in the Jacobian matrix. It is surprising to notice that this early achievement has been ignored in the vision literature since. His system for visually perceiving its state relative to a polyhedral object (satellite model in the laboratory) selected four visible corners fully autonomously out of a much larger total number by maximizing a goal function formed by entries of the Jacobian matrix (see Section 8.4.1.2).

Since the entries into a row of the Jacobian matrix contain the partial derivatives of feature position with respect to all state variables of an object, the fact that all the entries are close to zero also carries information. It can be interpreted as an indication that this feature does not depend (locally) on the state of the object; therefore, this feature should be discarded for a state update.

If all elements of a column of the Jacobian matrix are close to zero, this is an indication that all features modeled do not depend on the state variable corresponding to this column. Therefore, it does not make sense to try to improve the estimated value of this state component, and one should not wonder that the mathematical routine denies delivering good data. Estimation of this variable is not possible under these conditions (for whatever reason), and this component should

be removed from the list of variables to be updated. It has to be taken as a standard case, in general in vision, that only a selection of parameters and variables describing another object are observable at one time with the given aspect conditions. There has to be a management process in the object recognition and tracking procedures, which takes care of these particular properties of visual mapping (see later section on system integration).

If this information in properly set up Jacobian matrices is observed during tracking, much of the deplored brittleness of Kalman filtering should be gone.

http://www.springer.com/978-1-84628-637-7

Dynamic Vision for Perception and Control of Motion

Dickmanns, E.D.

2007, XVIII, 474 p. 244 illus., Hardcover
ISBN: 978-1-84628-637-7