Machine Vision and Applications (1988) 1:223–240

Machine Vision and Applications

Dynamic Monocular Machine Vision

Ernst Dieter Dickmanns and Volker Graefe

Fakultät für Luft- und Raumfahrttechnik (LRT), Universität der Bundeswehr München, W-Heisenberg-Weg 39, 8014 Neubiberg, West Germany

Abstract: A new approach to real-time machine vision in dynamic scenes is presented based on special hardware and methods for feature extraction and information processing. Using integral spatio-temporal models, it bypasses the nonunique inversion of the perspective projection by applying recursive least squares filtering. By prediction error feedback methods similar to those used in modern control theory, all spatial state variables including the velocity components are estimated. Only the last image of the sequence needs to be evaluated, thereby alleviating the real-time image sequence processing task.

Keywords: 4-D machine vision, real-time image sequence processing, automatic visual motion control, vehicle guidance

1. Introduction

Dynamic vision is more than fast processing of static image sequences. The dynamics aspect rests primarily in the scene observed or in the motion of the sensor and is independent of the image frequency; as in any sampled measurement process, high sampling rates are necessary for recovering highly dynamical changes. In vision, however, in addition to this, high sampling rates reduce the so-called correspondence problem, that is, keeping track of special image features or objects in space from one frame to the next.

Address reprint requests to: Prof. Ernst D. Dickmanns, Steuer-und Regelungstechnik, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, Institut Für Systemdynamik und Flugmechanik, Werner-Heisenberg-Weg 39, D-8014, Neubiberg, West Germany.

This research has been partially supported by the German Federal Ministry of Research and Technology (BMFT), the Deutsche Forschungsgemeinschaft (DFG), the Daimler-Benz AG, and by Messerschmitt-Bölkow-Blohm GmbH (MBB).

Note that humans, when talking about dynamic scenes, do not converse in image terms but do prefer spatial interpretations, both in position and velocity, whenever possible. They try to see motion of objects in space. Motion properties of objects are an integral part of a person's knowledge base like possible shapes and colors. Similarly in the approach described below, a direct spatial interpretation of image sequences is achieved by using spatial and temporal models in conjunction. This unified approach in space and time is the core of the 4-D method developed and tested for machine vision. Applications are discussed in a companion paper (Dickmanns and Graefe 1988, this issue; p. 241).

The immediate inclusion of temporal aspects is very essential since it allows a proper definition of state variables and the introduction of temporal continuity conditions for image sequence interpretation by exploiting differential equations. Geometric shape descriptions and generic models for motion *together* constitute the basis for an integrated spatio-temporal approach, which may be termed "4-D vision" or "dynamic vision."

This means that not just objects are being seen but motion processes of objects in space and time. Note that unlike "static" image sequence processing, dynamic vision has no separation between spatial object recognition from one frame to the next as a first step and motion reconstruction afterwards as a second one. Instead, object and motion are treated as a unit and the least squares fit for determining the best estimate for the object motion state, based on noise corrupted image sequences, is done in space and time simultaneously.

As a very beneficial side effect, the need for storing past images (e.g., for computation of displacement vector fields or optical flow) is reduced. The state of the scene observed is represented on a very high symbolic level by the shape descriptors and the spatio-temporal state variables including spatial ve-

locity components as an integral part (state vector components).

This approach provides an efficient framework for data fusion and active control of the viewing direction. Angular rates are state variables directly and translational velocity components of the egomotion are time integrals of the corresponding accelerations; both may easily be sensed by inertial sensors. Vision and inertial sensors have complementary properties when used for state recognition under egomotion: High angular rates, causing motion blur in the imaging process, are easily measured inertially; slow drift rates, hard to detect inertially, are easily discovered optically. For this reason, many organic species have developed this sensor combination and corresponding control facilities, for example, in vertebrates, the vestibular/ocular measurement and control system (Dichgans et al. 1973; Bizzi 1974). Active gaze control, in addition, allows the anchoring of the viewing direction on relatively fast moving prominent features of an object and thereby reduces motion blur for this object. If this object happens to be of special interest, the deterioration induced for the viewing conditions of most other objects may be acceptable. This fixation mode of vision also is very common in biological systems. In machine vision, as of course in biological vision too, a precisely servoed gaze anchoring allows the reading of object angular position from the measurement of mechanical angles while object shape may be determined from a quasistationary image.

For these reasons, active fast control of the viewing direction by the interpretation process is considered essential for dynamic vision. Therefore, it has been included in the systems design from the beginning.

The observation that in biological systems the sense of vision seems to be intimately linked to active motion control has lead us to consider motion *control* as the proper entry point for developing machine vision.

In hindsight, this turned out to be the right decision since the dynamical models of modern control theory proved to become the cornerstone of the new method for dynamic vision.

Motion control in space requires spatial or stereo vision. How many cameras are most appropriate for stereo vision? This question is still unresolved. Putting emphasis on motion and temporal integration, we decided to use just one camera. Spatial ambiguities may be resolved through motion stereo over time. In the case of active motion control, movements may be planned and executed in a way allowing to disambiguate a situation. For vehicle control as discussed in the companion paper (this volume) it seems more favorable to devote a second camera to high resolution imaging for better farsight than to direct stereo.

The considerations described above have lead to an approach to machine vision different from the mainstream of vision research originating from digital image processing and artificial intelligence. Gennery (1981, 1982) has taken a similar approach. In recent years Broida and Chellappa (1986) and Rives et al. (1986) seem to be heading in the same direction. For a literature survey on image sequence processing see Nagel (1983).

The next section summarizes some basic considerations on computer architectures for dynamic vision. In section 3 the nonuniform low level image processing schemes are described, upon which the approach is based. The general method for the higher levels of dynamic vision is developed and explained in section 4. Section 5 very briefly presents application results. All four application examples treated so far have been performed with real image sequence processing hardware in the real-time loop. More details on the implementation of the 4-D method, together with a discussion of the hardware developed, are given in the companion paper.

Finally, in section 6, development perspectives for the future are discussed.

2. Computer Architecture for Dynamic Vision

There are basically two approaches to the design of a real-time vision system. One is what might be called the brute force approach, using extremely fast hardware elements and possibly a massively parallel structure, yielding a supercomputer with an impressive power in terms of the notorious MIPS (million instructions per second). The other one is to look for the inherent structure and, possibly, simplicity of the problem of dynamic vision, and to find a computer architecture which is well matched to the task of visual motion control.

The second approach is, indeed, feasible and has led to the construction of a family of multiprocessor systems specialized for dynamic vision. The architecture of these real-time systems is very different from that of a typical image processing system. In spite of their relative simplicity, they have proven to be a very powerful hardware basis for various real-world experiments where mechanical systems or vehicles were controlled by dynamic vision.

An important concept upon which to base the design of a dynamic vision system is temporal continuity. Usually, natural scenes change only gradu-

ally, and if two pictures of such a scene are taken within a few milliseconds they will normally be very similar to each other.

In order to understand how the temporal continuity of natural scenes can facilitate dynamic vision, assume that a first TV image of such a scene has just been interpreted. It is then rather easy to interpret the immediately following image, as the differences between the two are very small. This observation has important consequences for the design of a real-time vision system. It means that the task of dynamic scene interpretation becomes easier if the time spent on each image is reduced, and that the task becomes more difficult if the system is slower. Therefore, the cycle time of the low level vision subsystem should ideally be less than one frame period of the TV signal used, making it possible to evaluate every single image as it is delivered by the camera. (The higher levels of the vision system which operate on symbolic descriptions of the scene may use longer cycle times, depending on the dynamics of the machine to be controlled and of the objects in the scene.)

Another important aspect on which to base the architecture of hardware for dynamic vision for motion control is the desired output of the system: it is the behavior of a visually controlled machine, and not, as often in traditional static image processing, either another image or a fairly complete, perhaps even verbal, description of the image.

The appropriate behavior of a vision controlled machine typically depends on the presence and location, or absence of certain objects in its environment. The vision task is then clearly goal directed, the first subtask being to locate features in the image which are indicative of the presence and location of important objects. It seems obvious that such features in many typical situations occupy only a small fraction of the total area of each image (Figure 1). It suffices then to process only those areas of each image which actually contain relevant features.

In dynamic scene interpretation the location of all important features is usually known in advance and with fairly good precision from the interpretation of previous images. This means that, when interpreting the next image in the sequence, the search space in which the feature of interest should be looked for is small, and the feature can be rediscovered rather quickly if the search is indeed focused on this small search space. This leads to the probably most important point in the design of hardware for real-time vision: since nearly all the relevant information in the image is contained in a limited number of small regions the combined size of

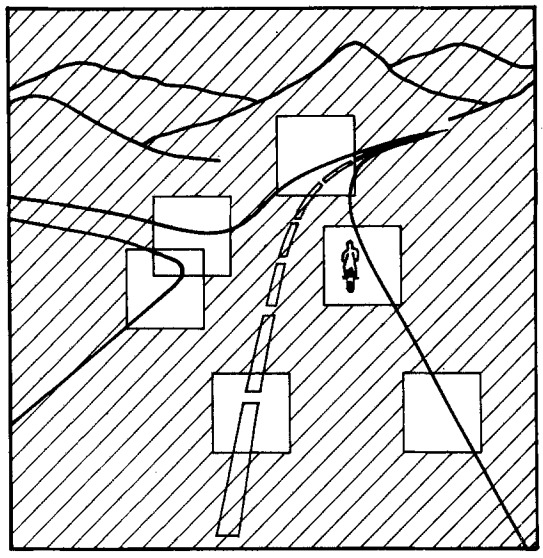

Figure 1. Small regions of an image contain almost all information relevant for motion control.

which is only a small fraction (often less than 10%) of the whole image, much will be gained if all the available computing power can be concentrated on those regions. Moreover, since each region may contain a different type of feature, it is important to be able to use different algorithms in each region.

This shows that a conventional image processing system which is designed to treat all pixels in an image in the same way does not have the proper structure for dynamic vision. The same is true for some massively parallel computers of the single instruction, multiple data (SIMD) type. Because these machines, too, must treat all pixels of an image in the same manner, they may waste 90% or more of their computing power on processing parts of the image which are known in advance to contain no relevant information. In the worst case, additional computing power is needed to delete all the irrelevant data which are produced in the process.

The concept of processing only a limited number of well defined regions within an image is also the key to a natural division of the problem into subtasks which can be executed in parallel on a coarsely grained multiprocessor system. Each parallel processor in such a system can be assigned one relevant region, and it can locate—independently of all other processors—the associated features in that region. Such a system not only has a very clear structure (one region—one group of features—one subtask—one processor), but it can also be very efficient, since the parallel processors do not have to spend time synchronizing or coordinating each other.

An important key to this concept is that the size, shape, and location of each region may be varied

during the interpretation process in a data dependent way. Each region will normally be continuously adjusted in such a way as to completely contain a relevant feature or object. If the regions were fixed, the system would be much less efficient for two reasons. First, some regions would contain no relevant information but would nevertheless absorb computing power; secondly, some features or objects would be dissected by the borders between regions, creating the difficulty of detecting and interpreting arbitrarily dissected parts, representing them internally, and finally recombining them into objects.

Architectural details of a family of vision systems designed according to these concepts are given in the companion paper (this volume).

3. Feature Detection and Tracking Algorithms

In a dynamic vision system as discussed here, the time available for feature extraction is limited to about 50 ms per image. In order to evaluate every available image, the time should in fact be limited to less than one frame period of the TV camera (17 ms). An image contains (roughly) 10^5 pixels, and standard image processing methods require many operations per pixel [Reddy (1978) has estimated that 1000 operations per pixel are required for segmentation]. It is therefore obvious that speed is a most critical characteristic of any feature extraction algorithm for dynamic vision.

Two powerful approaches are available to maximize the speed of feature extraction in dynamic vision: application of advance knowledge, and a strict concentration on obtaining that, and *only that*, information which is necessary to accomplish the given task. In other words, only a relatively small number of carefully selected relevant features should be extracted depending on the situation and on the requirements of the task; knowledge should be applied to maximize the efficiency in processing the selected features.

Both approaches emphasize the difference between static image processing and dynamic vision. In static image processing very little is often known in advance of the image presented to the system. The task then is to extract as much information from the image as possible. This is very different from dynamic vision, where each new image is known to be a natural continuation of a sequence of images which the system has interpreted already; differences relative to the previous image are to be expected in small details only. Most of the time the feature extraction has to answer only a small number of precise questions relating to one or another of the small differences, such as how much and in which direction did a certain feature move in a small fraction of a second.

These two approaches will be discussed in more detail in the sequel.

3.1 Task Specific Feature Extraction

Limiting the number of features to be processed is not meant to exclude useful redundancy, which is absolutely necessary for any robust system, but rather to avoid wasting time or computing resources on processing irrelevant parts of a scene. All the available resources in a dynamic vision system should be concentrated on obtaining that information which is necessary, or at least helpful, to accomplish a certain task, usually the control of a moving system. The point is to extract only *task relevant* features and not every conceivably extractable feature. In the applications described in the companion paper, never more than about 10 features were needed, and 1000 features will probably be sufficient to handle rather complex scenes.

One of the problems which must be solved in the design of a dynamic vision system for a specific task is defining and selecting the relevant features. This is best done in a top-down approach, starting from the task the vision system is supposed to execute.

For balancing an inverted pendulum on an electric cart, which is a simple but typical example, it is sufficient to know the coordinates of two points on the rod as a function of time. If the coordinates of more points are available, this provides valuable redundancy which can be used to make the system more robust. On the other hand, nothing can be gained for the performance of the system by analyzing the background or the floor. All the available computing power should, therefore, be concentrated on locating various points of the rod.

Similarly, in the docking experiment discussed in the companion paper, the docking partner can be recognized and its relative position can be estimated by localizing corners of its contour in the image. Analyzing the entire contour might provide useful redundancy, but certainly all background features are irrelevant and should be ignored in the interest of efficiency.

Defining all features relevant for an autonomous vehicle in a natural environment is more difficult because of the great variety of situations it may encounter. It is, however, easy to see that large parts of a typical scene will never contain any relevant information, such as the mountains and the sky in Figure 1. Certain features will always be relevant, for instance, grey level edges which are char-

acteristic of the borders of the road or lane, while it is not clear yet what kinds of other features may be relevant for the detection and classification of obstacles in certain situations. A pragmatic approach is to start with relatively simple scenes, such as an empty freeway, where the number of relevant features is small and their nature is obvious (borders of the road or lane), and then, as experience is accumulated, admit more complexity, like obstacles, other vehicles or intersections.

In any case, the key point is that the low level part of a dynamic vision system should process only those features which yield information actually required by the higher levels. The requirements of the higher levels are derived from the desired performance of the machine to be controlled.

3.2 Knowledge Based Feature Extraction

Typically, although there are a few exceptions, the appearance of a dynamic scene changes only gradually; this is due to the inertia and limited energy of all massive objects. In a sequence of TV images of such a scene it is, therefore, possible to predict the appearance of each new image from the previous images. Predicting the entire image would be expensive, and usually it suffices to predict the location, and possibly the appearance, of a limited number of features, like edges, corners, etc. The prediction will not be perfectly correct, but that is not necessary. All that is required is that the remaining search space, which corresponds to the uncertainty area of the prediction, be sufficiently small to complete the actual search for the feature within one video cycle. As most features in typical scenes move by at most one or two pixels between two successive images, a "zeroth order prediction," where it is assumed that the feature will reappear at the same location as in the last image, will often be sufficient. In exceptional cases with very fast moving objects (e.g., the inverted pendulum in the start-up phase or after an extreme external disturbance) a "first order prediction," which also takes the estimated velocity of the feature into account, may sometimes be more appropriate. It should be noted, however, that the motion blur caused by all normal TV cameras places a natural limit on the admissible velocity of features in an image.

Such a prediction-and-correction method has been the key to the success in balancing the inverted pendulum. It was used there in combination with an ad hoc method for locating the pendulum in the image, based on an anisotropic, nonlinear filter. The filter took into account the effects of motion blur, caused by the sometimes rapid motions of the system (Haas and Graefe 1983). Such a fairly sophisticated method was necessary to cope with such problems as camera shading, irregular lighting, camera and electronic noise, and the high speed of the mechanical system. Applying the filter in the entire image in real time would have exceeded the capabilities of even very powerful computers. By predicting the location of the pendulum in the image using either a zeroth or a first order prediction, depending on momentary velocity, the search space was reduced to less than 1% of the image. Two eight-bit microprocessors were then sufficient for the task.

As a more general realization of the prediction–correction concept, the method of "controlled correlation," sometimes also referred to as "intelligent correlation," has been developed (Kuhnert 1986a, 1986c, 1988). First real-time results obtained with this method (road tracking and obstacle detection from within a simulated autonomous vehicle) were reported in Kuhnert and Zapp (1985).

Correlation is the basis of many visual trackers. It is, however, not often used in computer vision, probably because it is considered computationally expensive. In its discrete non-normalized form, the 2-D correlation function C is defined as

$$C_{i,j} = \sum_k \sum_l I_{i+k,j+l} \cdot M_{k,l}$$

$$k = -K \ldots K; l = -L \ldots L$$

The essence of correlation is that a 2-D reference pattern M (the "mask"), which is usually much smaller than the image, is laid over the image I (2-D array of gray-level values), where each element of the mask is multiplied with the corresponding pixel of the image, and the products are summed, yielding a correlation value corresponding to the position (i,j) of the mask. The process is then repeated for all positions of the mask relative to the image, yielding a correlation function. If a region in the image resembles the reference pattern, the correlation function has a peak at that location. Thus, correlation can be used to find such regions.

If there are n pixels in the image and m elements in the mask ($m \ll n$) the computation of the correlation function requires almost $m \cdot n$ multiplication, which usually is a very large number. After the correlation function has been computed, it has to be searched for the relevant peaks. Because of the large number of correlation values this search is expensive, too.

On the other hand, correlation has some very desirable properties. It is very flexible in the sense

that any pattern can be used as a mask and thus be looked for in the image. Most importantly, however, correlation is robust against noise. From communication theory it is known that correlation is the best linear method to detect a signal in the presence of additive ergodic white noise.

Noise is a severe problem in feature extraction. It may exist in many forms, for example, camera noise (time varying and fixed pattern), electronic noise in the analog part of the vision system, rounding errors in the digital part, irregular lighting, shadows, dirt covering parts of a scene, or branches of trees moving in the wind. It causes many feature extraction methods, which work well in noise-free synthetic images, to break down in natural scenes. Not all the kinds of noise which impede the processing of natural scenes can be considered additive, ergodic, and white. But, nevertheless, correlation is certainly a good candidate for a noise resistant feature extraction method. This is supported by results of an investigation (Kuhnert 1984) where several edge detectors were compared with the human visual system with respect to their ability to detect edges in synthetic pseudo-noise images. The correlation-based operators reproduced the abilities of the test persons more closely than any of the other ones. This observation adds to the attractiveness of correlation, since the human visual system is one of the best vision systems known.

Several types have been taken to reduce the computational cost of feature extraction by correlation, leading to "controlled correlation" and making it a very efficient method for dynamic vision. The first step is to correlate the mask only with a small region of interest which includes the predicted location of the feature, rather than with the entire image. This alone can reduce the required effort by more than two orders of magnitude.

If only elementary features such as edge elements are searched, the correlation masks can be small and, moreover, a small set of masks is sufficient to cover all possible orientations of short edge elements. If, on the other hand, complex features or even images of entire physical objects are looked for, the masks, in general, will be larger and many different ones will be needed to cover various sizes, orientations, aspect angles, and illumination conditions of a single class of feature or object. In regard to efficiency, short edge elements are therefore excellent features to base an image analysis on. They may be used in subsequent steps to construct longer edges, while still higher levels in the system may use knowledge to combine several edges, reconstructing 2-D images of objects and finally the 3-D objects and their motions. This result perhaps bears a relationship with the findings of Hubel and Wiesel (1959) indicating that vertebrates also have receptive fields in their visual system which are tuned to short edge elements with specific orientations.

Depending on the situation, choosing an appropriate path along which to look for a correlation peak often helps to gain additional advantages. It should be remembered that one is not really interested in the entire correlation function, but only in the location of that peak which corresponds to the correct feature, or, considering the effects of noise and of possibly existing false features in the vicinity of the desired one, of a small number of candidates for the correct peak. If a good search path is chosen, all good candidates for the correct peak can be found quickly and the search can then be discontinued.

Correlation masks often contain elements whose absolute magnitude is much smaller than that of others. The correlation function, and in particular the locations of its peaks, do not change much if these small values are replaced by zero. It is possible to implement the correlation method in such a way that mask elements of value zero are skipped in the execution of the program and do not cause any operation of the computer at all. Setting many mask coefficients to zero will then reduce the number of operations to be executed, making the resulting algorithm faster. Masks of this type are called "sparsely populated." It is important to notice that, when sparsely populated masks are used, the computational cost of the algorithm does not depend on the extent of the mask, but only on the number of its nonzero elements.

Most correlation algorithms cause the computer to spend much time generating addresses and checking loop termination conditions. If all addresses are computed during compilation and if loops are avoided altogether, a much faster program results, consisting of only one (very long) linear piece of code. An additional step of optimization applies only to computers like the BVV 2, which require much more time for a multiplication than for an addition (the coprocessor of the BVV 3 is different, it multiplies and adds simultaneously). The fact is utilized that even very coarsely quantized correlation masks, if they meet certain symmetry conditions, are almost as effective as finely quantized masks (Kuhnert 1988). This is true even for ternary masks which contain only elements with values of -1, 0, and $+1$. Such masks can be realized by algorithms which perform subtractions and additions only.

Figure 2 shows an example of controlled correlation as it is used to track the left shoulder of a road

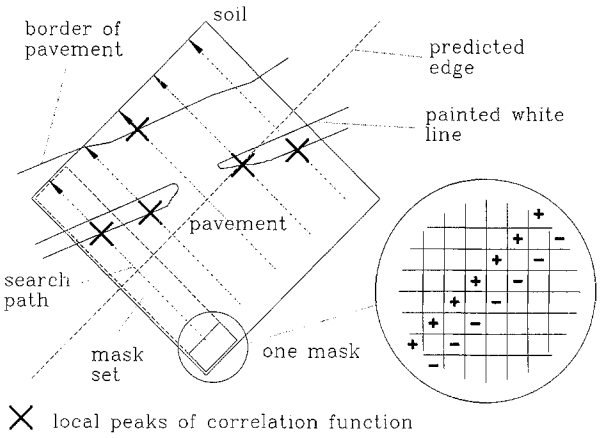

Figure 2. Using controlled correlation in order to find the left shoulder of a road. [The figure shows a very small section of the total image where the dark road surface borders the brighter soil. In the mask, "+" represents a value of +1 and "−" a value of −1. The five search paths together constitute the region of interest.]

from within a vehicle; the figure shows a small section of the image. A white line is painted on the road near the border of the pavement. The feature looked for is the right edge of the painted white line. Its *predicted* location and direction are indicated as a dashed line. The region of interest is centered around the predicted edge; its size is 2016 pixels (3% of the entire image). The mask is also indicated in Figure 2; it contains 14 nonzero elements. The correlation function is initially computed along a search path beginning near the lower corner of the region of interest and ending near its left corner. The search path is a straight line perpendicular to the predicted edge. Of the peaks found on the search path either the first peak of significant size or the "best" peak is accepted as a border point.

In natural scenes it is sometimes difficult to discriminate between a valid peak and noise effects or irregularities in the scene. Therefore, several algorithms are available to select a "best" peak of the correlation function; one of them may be chosen during runtime according to the situation.

All masks constituting the initial search path form a mask set. The search is repeated four more times on parallel search paths, using the same mask set, shifted to the upper right by six pixels each time, yielding altogether five border points.

In Figure 2 it is assumed that the white line is not clearly visible everywhere. It is, therefore, missed by the third search path, and the border of the pavement is found instead. It happens frequently in dynamic vision that a feature is missed temporarily and it is questionable whether an error-free feature extraction is at all possible. Fortunately, such errors usually do not persist and in one of the next few images the lost feature or an equivalent one will be available again. In reality any robust system will have to use redundancy and world knowledge to handle such errors as a matter of routine. In any case it is up to the higher levels of the system to handle such a situation, for example, to recognize the outlying point and to eliminate it, or to tolerate the error. A more sophisticated version of the feature extractor could be designed which would not accept the edge of the pavement instead of the white line (Kuhnert 1986c); it would, however, run more slowly. It should be realized that, because of the transient nature of such errors, in dynamic vision the total system performance may well be better with a fast algorithm for feature extraction which occasionally makes errors, than with a less error prone, but slower algorithm.

The controlled correlation as described requires less than 33 ms on the 8086 microprocessor of the BVV 2. A similar algorithm employing only three paths instead of five runs in less than 16 ms. Various versions of this method have been used in different applications, among them the automatic aircraft landing and the autonomous road vehicle described in the companion paper. They have also been tested extensively in laboratory simulations where videotapes and 8 mm films taken from within a landing airplane and from cars driving on freeways or in cities, were played back and analyzed in real time (Kuhnert 1986a; Eberl 1987). These experiments were greatly simplified by the robustness of the algorithms. It was not even necessary to synchronize the film projector with the TV camera of the vision system which picked up the projected images from a screen; the features could be tracked reliably in spite of the severe fluctuations in brightness of the digitized TV images caused by the lack of synchronism.

Controlled correlation can also be used to track features other than edge elements, for instance, corners. Figure 3 shows an example (Kuhnert 1988) based on a very simple mask. Applying the mask once takes about 50 μs on an 8086 microprocessor. The mask works quite well, even if a corner is rotated slightly or if the enclosed angle is not exactly 90 deg. The problem is that the response it gives for a perfect match is only twice as high as the response for, say, a straight horizontal edge. This lack of selectivity is both an advantage and a disadvantage. If a mask is not very selective it will yield many false responses in a natural scene. If the mask is very selective (such a mask can easily be constructed by increasing its size), very large mask sets are needed to cover all possible angles and orienta-

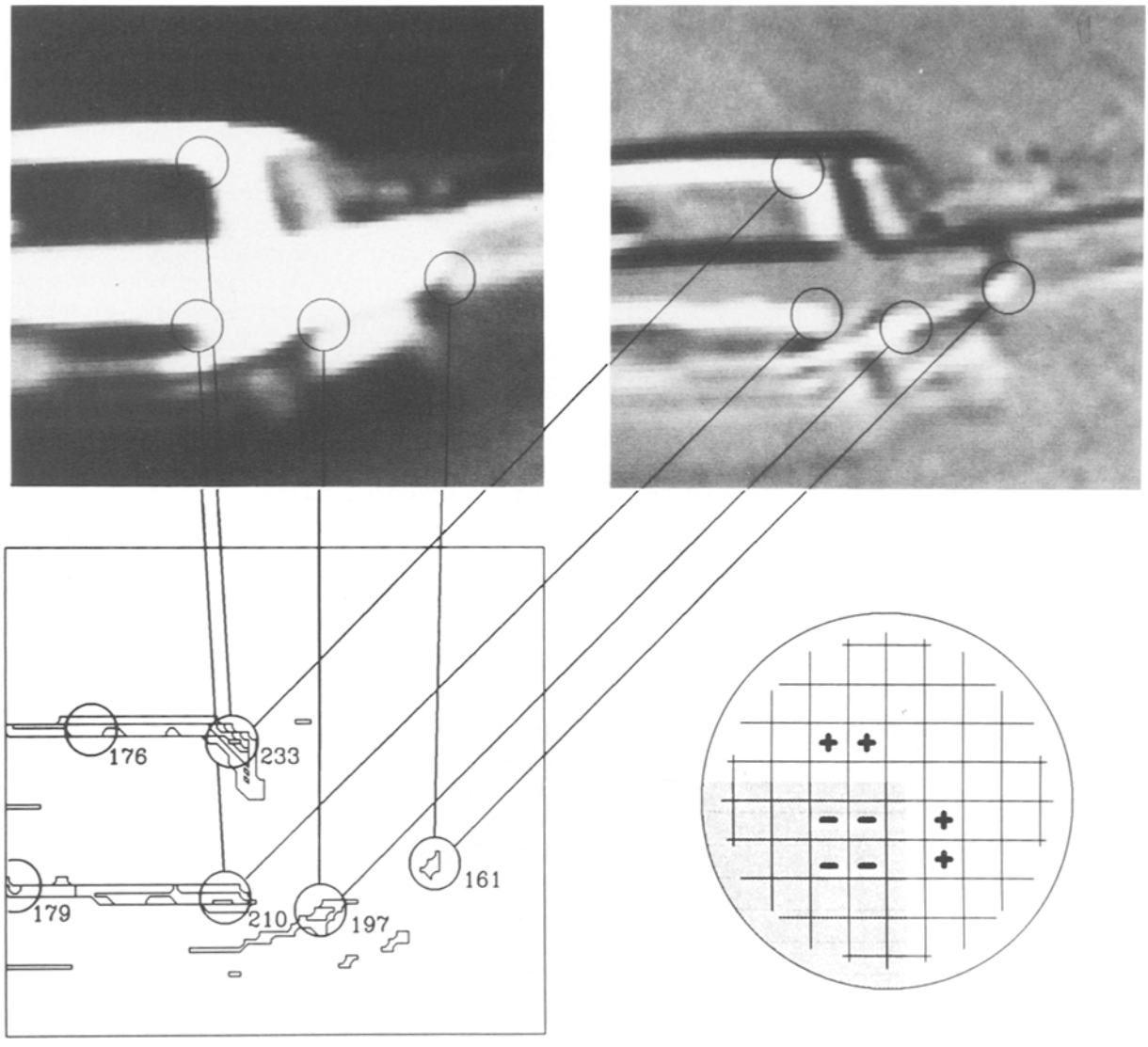

Figure 3. Detection of corners in a noisy image of a natural scene by correlation with a small ternary mask. Upper left: Original image (photo from the monitor of an image processing system); the figure shows only a small section of the entire image. Lower right: The correlation mask; "+" represents a value of +1 and "−" a value of −1; the shading shows the theoretical prototype corner of the mask. Upper right: Visualization of the correlation function; bright areas correspond to positive values and dark areas to negative values. Lower left: selected contours and the greatest maxima of the (non-normalized) correlation function; the numbers indicate the magnitudes. [The image stems from an 8-mm movie, taken from within a moving car; it has been projected onto a screen, picked up with a TV camera (262 TV lines) and digitized with 8 bits of resolution.]

tions of corners. This problem has to be investigated in greater detail, but probably it is better to look for two intersecting edges separately and then compute the point and angle of intersection, rather than looking for each specific type of corner directly.

Corners have been used as features in the vehicle docking experiment (Wünsche 1987; Kuhnert 1988). The method was knowledge based; however, it was not based on correlation. The contours of visible objects were extracted and maxima in their "curvature" were taken as corner points. This method was implemented on the older BVV 1 and cannot be generalized easily. It is, however, also based on the concept of prediction and correction, predicting the location of the corners in the image and analyzing only those parts of the contours which are close to the predicted locations.

3.3 Feature Detection

So far it has been assumed that the position of a feature in the image can be predicted using the

knowledge gained from processing the immediately preceding images. This is usually true, but there are at least three exceptions: the initialization phase, the recovery after a feature has been lost, and the appearance of a new object in the scene. What makes the initialization phase manageable, is the fact that it is not time critical. It is difficult to make general statements about the initialization phase, but enough scene- or task-specific knowledge can usually be built into the system to avoid searching the entire image for each single feature that is needed. In the case of the inverted pendulum, a horizontal search path through the center of the image will find the pendulum, and in the case of a road vehicle it can, perhaps, be assumed that the vehicle is initially standing on a road, oriented nearly parallel to it. This limits the regions in the image, where the borders of the lane can be expected, sufficiently for efficient search.

It is normal for a feature being tracked to get lost once in a while, for example, the center edge element in Figure 2 or the border of the road when passing under a bridge while the auto-iris has not had time to adjust to the darkness. Usually, a lost feature will soon reappear near the location where it has been found last or near a location which can be easily predicted using adjacent features. If this does not happen soon enough, or if the feature tracker locks on to a wrong feature, higher levels in the system, which have a more comprehensive knowledge of the situation, must guide the feature extraction level to reacquire the lost feature. Wünsche (1983, 1987) has shown that this can be done very effectively, yielding a remarkably robust system.

The discovery of objects which suddenly enter the scene is of a different nature and often difficult, in particular if little is known regarding the visual appearance of the new objects and the region of the image where they will first be visible. An example is the discovery of obstacles which might obstruct the path of a vehicle. What makes this class of problems particularly difficult is that, unlike in the initialization phase, only a small amount of time is available.

4. Feature Based 4-D Dynamic Scene Analysis Using Integral Spatio-Temporal World Models

In the previous section it has been shown how simple elementary features can be extracted efficiently in a robust manner at video rate with relatively modest computing power. But what are these simple features good for? They receive their significance from a method which is able to provide a link from 3-D features on objects moving in space to 2-D features in the image. Straight contour elements are especially suited for this purpose since on a proper scale many objects have straight or nearly straight contour elements and zero curvature is an invariance property under perspective projection.

As caricatures show in a most impressive way, lines and curves do carry most of the information characterizing a scene. If the linear features (contour or edge elements) detected are considered to be tangents to curves, they constitute very general elements for shape description in differential geometry terms (Dickmanns 1985a). If sets of features in the image move in conjunction, it can be hypothesized that they belong to the same object, though there are exceptional cases where this may not be true.

Objects in the real world may be described as 3-D shapes realized by a massive substance having a center of gravity. The dynamical models for motion of and around the center of gravity (CG) of a physical object are combined with representations of its 3-D shape, emphasizing the position of (contour element) features relative to the CG. Feature groupings (aggregations) in the image, interpreted as a perspective map, are used to recognize objects in a 3-D scene. However, only in the initialization phase, if at all, is this done in the usual nonunique inverse way. If the type of scene is known a priori, forward perspective mapping of generic models and adjustment of model and relative position parameters is done until the measured image is matched or until the model pool is exhausted. Once the real-time phase has been initiated, only the model based approach is applied. Up to now, features used have been limited to linear contour elements (edge elements) and corners. A theory for efficient curvilinear shape representation has been developed and is presently under investigation (Dickmanns 1985a, 1985b).

Figure 4 shows a juxtaposition of the conventional method in image sequence processing (top) with the integrated 4-D approach (bottom). In the former, a considerable amount of computation is done in image coordinates and in inter-image comparisons, whereas in the latter a 4-D representation of a model world is being maintained in the interpretation process and servo-controlled by measurement data from the last image of the sequence only.

This model based prediction–error–feedback of feature positions has several advantages: no differencing between noise corrupted images of a sequence is required for obtaining velocity components (as in optical flow computation). Motion interpretation does not have to be done in image coordinates first with a reinterpretation in space af-

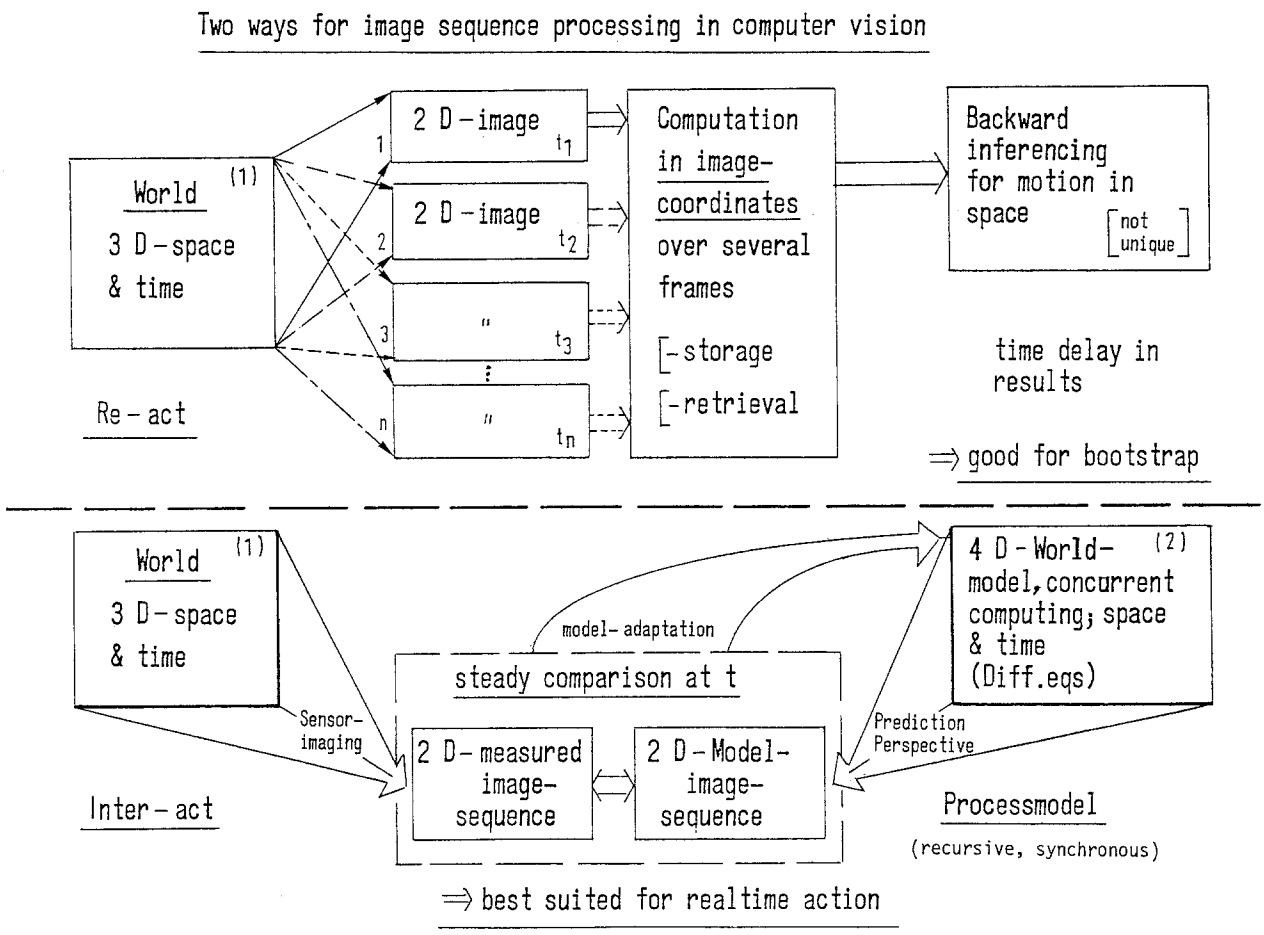

Figure 4. Two basically different methods of image sequence processing.

terwards; instead, all interpretation is immediately performed in the 4-D space–time continuum.

Similar to using shape models for object recognition, temporal models appear to be of great advantage for motion parameter recognition: As the term *re*cognition tells, the interpretation process does have background knowledge of what it is going to "see," at least as a generic class from which special objects are being instantiated through data based hypothesis generation. This usual approach for shape recognition has been augmented by associating the object with its environment and the viewing conditions for image sequence taking: If the object is at rest and the camera moves, a generic dynamical model with state variables x for the camera motion is introduced; if the camera is at rest and the object moves, a model for this motion is selected. In both cases physical motion constraints and optional control or disturbance inputs are included. (The case where both camera and object are moving is much more difficult and presently under investigation.)

The general standard form of a generic dynamical model is a set of n differential equations for n state variables, usually nonlinear, sometimes with time-varying coefficients. As in modern control theory for sampled systems, locally linearized approximations with transition matrices for the sampling period T and influence coefficients for the control are being used. All coefficients are assumed to be constant over T. This basic cycle of period T for model based measurement interpretation and control action has been selected around 0.1 s (10 Hz); the more complex situation analysis on a higher level may be slower.

The goal of basing visual process recognition on an integral spatio-temporal world model is threefold:

1. Eliminate the need to access past images
2. Determine spatial velocity components by smoothing numerical integration
3. Bypass the nonunique inversion of the perspective projection by doing recursive least squares state estimation exploiting the Jacobian matrix of the measured image features (their partial derivatives with respect to the state variables of the dynamical model).

This matrix contains rich information for the direct interpretation of feature prediction errors in spatial coordinates; this information becomes accessible through the measurement model including perspective projection and sensor properties.

One might say that the problem of dynamic scene analysis is solved by doing a servo-controlled synthesis, based on feature–prediction–error feedback, using generic world models for shape, motion, and the measurement process. This is in contrast to today's common approaches that try to achieve geometric reconstructions from several images (shown schematically in the top part of Figure 4).

The recursive estimation based on the smoothing numerical process of integration proceeds as follows: An estimate $\hat{x}$ of the complete state vector x describing the process to be interpreted is assumed to be given; this hypothesis generation in the initialization phase is a hard task in partially or fully unknown environments. Knowing the n vector $\hat{x}$ and a dynamical model of the process in the discrete form for sampled measurement and control, a state prediction x^* for the next measurement at time $(k + 1)T$ can be made

$$x^*[(k + 1)T] = A[k]\hat{x}[kT] + Bu[kT] \tag{1}$$

A is a $n \cdot n$ state transition matric over one sampling period T and B is the $n \cdot q$ control effectiveness matrix for the q components of the control vector u assumed to be constant over one period T.

If the shape of the objects observed and the relative geometry is known, maybe even described in terms of state components x_i^*, the predicted position y^* of features in the next image can be derived by forward application of the laws of perspective projection exploiting a model of the actual camera used for measurement. In general, this will be a nonlinear relationship containing measurement parameters p

$$y^* = f(x^*,p) \tag{2}$$

Both the process modeled in Eq. (1) and the measurements will be corrupted by noise, designated $v(kT)$ for Eq. (1) and $w(kT)$ for the measurements. The problem is to determine best estimates for the state

$$x = x^* + \delta x \tag{3}$$

given the measured quantities

$$y = f(x,p) + w \tag{4}$$

Assuming that the influence of the noise is small and its average is zero, a linearized relationship between y, y^* and x, $\hat{x}$ may be a good approximation to reality

$$\begin{aligned}\delta y &= y - y^* \\ &= f(x^* + \delta\hat{x},p) + w - f(x^*,p) \\ &\approx \frac{\partial f(x^*,p)}{\partial x^*}\delta\hat{x} + w \\ &= C\delta\hat{x} + w\end{aligned} \tag{5}$$

Note that the Jacobian matrix C contains the $m \cdot n$ partial derivatives of the m measurement quantities y^* as predicted, relative to the state variables x in 3-D space including the spatial velocity components.

Figure 5 shows the physical meaning of the C matrix (some components) for a simple top-down view by the reader onto a camera at point P imaging a rectangular box O_o. The present state is given by the distance r and the polar angle coordinate υ to the center and the angular orientation ψ of the object around its center. In the image plane (normal to r and to the plane containing the figure as shown) the edges of the box, designated M_{10}, M_{20}, and M_{30}, may be tracked laterally by three line element features. In the two subfigures at the bottom of Figure 5 (exploded view with 90 deg plane change), the feature position in the image is shown for the nominal state vector (index 0) and for state vectors with perturbed components, one each individually:

1. Radial translation (index r)
2. Azimuthal translation (index υ)
3. Rotational displacement (index ψ)

the left subfigure shows the feature displacements δy_{2i} for feature 2 for each perturbation in one state variable keeping the other ones at the nominal value; the right subfigure shows the same for feature 1.

Combining the state variable change, for example, δr, with the feature position change (δy_{1r}), one obtains the partial derivative: for example

$$C_{y1r} \approx \delta y_{1r}/\delta r$$

Performing this analytically, based on the 4-D model for all features and all feature coordinates (besides the image coordinates y_B and z_B, a rotational feature angle Φ_B may also be measurable) with respect to all state components yields the C matrix. It is rich information provided by the integral spatio-temporal world model which allows the bypassing of the nonunique inverse perspective

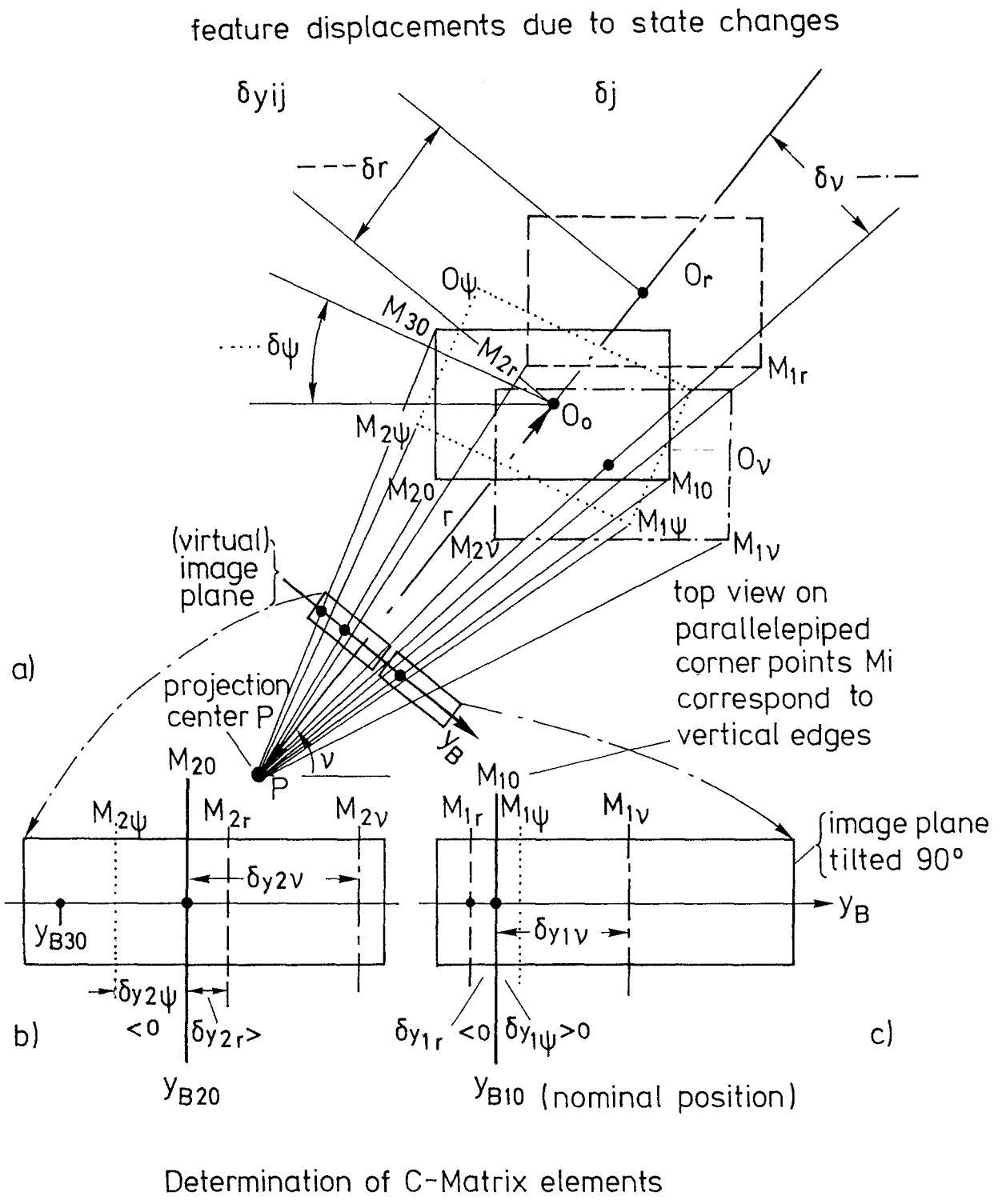

Figure 5. Physical meaning of C matrix elements for recursive least squares state estimation by 4-D vision. (a) Object and imaging geometry, nominal and perturbed. (b) y position shifts for feature 2. (c) y position shifts for feature 1 due to singular state changes.

transformation. In fact, it allows going to a 4-D representation of the scene observed from the image data in just one step. This is performed recursively, based on data from the last image only. Besides exploiting this information for state estimation in a numerical (procedural) way, it may also be used for spatial reasoning and general inferencing in a supervisory process on hierarchically higher levels. The recursive least squares state estimation is done using well-known techniques: If the covariance matri-

ces of the noise processes v and w are known, Kalman filter techniques or derivatives thereof may be used (Bierman 1975; 1977; Maybeck 1979). The new best estimate $\hat{x}$ then becomes

$$\hat{x} = x^* + K(y - y^*) \tag{6}$$

where the gain matrix K for innovation is determined depending on the method used; sequential formulations well suited for time varying measurement vector lengths, such as those due to occlusion, are available (Wünsche 1987). One set of equations for the K update showing the role of the Jacobian matrix C is given below:

$$\begin{aligned} K(k) &= P^*(k)C^T(k)\,[C(k)P^*(k)C^T(k) + R(k)]^{-1} \\ P^*(k) &= A(k-1)P(k-1) + Q(k-1) \\ P(k-1) &= P^*(k-1) - K(k-1)C(k-1)P^*(k-1), \end{aligned} \tag{7}$$

where Q = covariance matrix of process noise
R = covariance matrix of measurement noise
$P^*(0)$ is selected according to the confidence in the initial values $x^*(0)$.

Figure 6 shows the recursive 4-D image sequence interpretation method in the form of a block diagram. At the top left the real world is sensed by a television camera (TV) the video signal of which is digitized by the image sequence processing system BVV and given onto a video bus. In the initialization mode, the processors of the BVV are coordinated to do a feature search over the entire search space; based on the distribution and orientation of the features found, object hypotheses are generated consisting of shape, relative position, and angular orientation components (upper right). The 3-D shape instantiation proposed is transferred into the "geometric reasoning" block of the 4-D real-time world representation (center of Figure 6, "world 2" in the interpretation process); the relative position and orientation estimates are installed as the initial conditions for state prediction together with the proper dynamical model (lower center left). The predicted state of the CG motion is combined with the shape description of the object (shown as a circled "and" (Λ) sign in the geometric reasoning box) to yield the internal representation of this object at the point in time when the next measurement is going to be taken. According to this state the visibility of various features is checked and those which are best suited for relative state estimation are selected for tracking (upper line of geometric reasoning block). This automatic dynamic allocation of processing power to meaningful areas of interest, decided upon at a high interpretation level

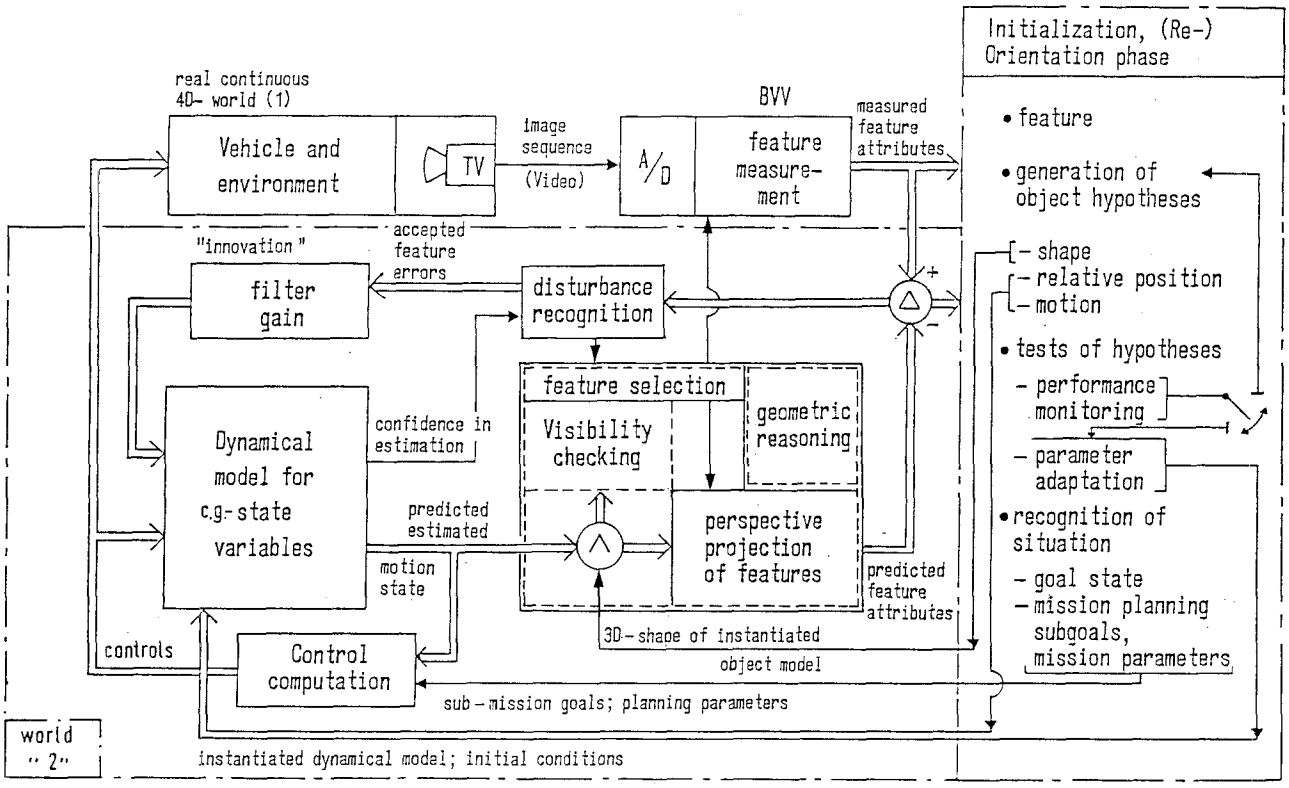

Figure 6. Block diagram showing the information flow in 4-D recursive state estimation for dynamic machine vision.

with a rich world representation as a basis for inferencing, is considered to be most beneficial for efficient dynamic scene analysis. It has been developed in (Wünsche 1987): The determinant of the pseudo-inverse of the properly scaled and weighted C matrix as occurring in a Gauss/Markov estimator formulation is maximized by selecting different feature combinations (global version); in a local version suited for real-time application, single features are substituted by other ones, one at a time, and the local maximum always is adopted.

The features selected are communicated to two locations:

1. To the BVV system for attention focussing and nonuniform image processing
2. To the perspective mapping module within the geometric reasoning block for computing the "model image" as the reference for prediction–error–feedback.

Note that at this point the Jacobian matrix C [Eq. (5)] is computed which is instrumental for bypassing the nonunique inversion of the perspective 3-D → 2-D mapping by recursive least squares filtering; the matrix C contains all the essential first order dependencies needed for intelligent interpretation of the measured feature data, given the scene model.

The difference between the measured and the predicted feature positions, formed at the circle designated with △ (upper right) in Figure 6, is used for adapting the model state to the measurements. First, outlyers are removed exploiting a confidence measure which later on results from the estimation process itself. Note that through proper use of the C matrix the accepted feature measurement data are directly interpreted in spatio-temporal state variables (3-D position, orientation, and velocity components).

The interpretation process is monitored through prediction error checking (at the △ circle, upper right). If systematic errors occurring over longer periods in time are detected, parameters of the model may be changed or other models may be activated. Up to now this has been done off line but will be done on line in the future.

If the prediction errors converge initially and remain small thereafter, the process is considered to be recognized. Note that shape and motion are recognized simultaneously assuming the model parameters to represent the invariant properties even though the image features may change continuously.

Knowledge of the complete state vector allows to apply state feedback controllers for high performance (lower left). There may be a direct feedback for fast reflex-like behavior. By checking the state against a sequence of goal states for coordinated mission performance, and by superimposing a control model switching, very flexible and highly efficient behavioral competences may be achieved. The controls actually output are fed both into the real-world machine being controlled and the (world 2) model in order to generate "expected" motion state components [see Eq. (1), last term; signal flow in Figure 6 left].

5. Applications

The general method as described above evolved during application to four problem areas.

5.1 Balancing of an Inverted Pendulum

The first real-time hardware application was in the early 1980s in the balancing of rods of various lengths in one degree of freedom on an electrocart. Rods from 0.4 to 2 m length have been investigated (Haas 1982; Meissner 1982; Meissner and Dickmanns 1983). Closed-loop eigenvalues of up to 1 Hz have been achieved. More details are given in section 3 of the companion paper.

5.2 Vehicle Docking

A frequent task in robotics is to position a controllable 3-D vehicle relative to another 3-D object. Using the dynamic approach to computer vision described above, H. J. Wünsche developed several important implementational details and demonstrated its performance and efficiency in fully autonomous docking maneuvers in the laboratory (Wünsche 1987).

A table-top air cushion vehicle with computer controlled reaction jets has the task of autonomously recognizing the situation in its (technical) environment, consisting of several objects of known 3-D shape but unknown position and orientation. Then, it has to perform a docking maneuver with a particular one of these objects.

More details are given in section 4 of the companion paper and in Wünsche (1986; 1987) and Dickmanns and Wünsche (1986b).

5.3 Road Vehicle Guidance

The tasks of a vision system for road vehicle applications may be manifold. Both the vehicle state relative to the road and environmental parameters may be determined in order to support the driver or for achieving autopilot capabilities. By continuously observing the road and its environment the

following items relevant to safe road vehicle guidance can be estimated or recognized in principle: road curvature, both horizontal and vertical, lane width and number of lanes, surface conditions such as smoothness, surface states such as dry, wet, or dirt or snow covered, presence of obstacles or other vehicles and traffic signs. A discussion of possibilities and problems related to the application of machine vision to road vehicle guidance is given in Dickmanns (1986).

Only the guidance task proper will be discussed in the companion paper, section 5, demonstrating the specific application of the integrated 4-D approach described in general terms in section 4. The following results have been achieved: Fully autonomous runs starting from rest have been performed under various road and weather conditions, including bright sunshine and light rain as well as road surfaces with and without lane markings. Increasing the maximum speed limit step by step, in August 1987 the maximum speed of the vehicle ($V_{max} \approx 96$ km/h on a level surface) has been reached. In order to obtain some results with respect to reliability, the total run length of more than 20 km has been driven several times (in both directions) without the need for intervention by the safety driver in the driver's seat.

Lane changes from right to left and back have been performed on free lanes as well as highway entry maneuvers from the acceleration strip.

The following conclusions for road vehicle guidance can be drawn: The capability of real time image sequence processing for guiding high speed road vehicles along well structured roads is becoming a reality. The method derived has been shown to allow autonomous vehicle guidance even at high speed with a relatively small set of today's microcomputers.

More computing power will be needed to improve the checking for obstacles and other objects in a less restricted environment. In principle, the vision system considered has the growth potential to allow the development of autonomous vehicles that fit neatly into the traffic system developed up to now for the human driver. Both gradual deployment and mixed human and automatic traffic seem to be possible.

5.4 Aircraft Landing Approach

This is the most complex real-time motion control task solved by computer vision by our group up to now. Aircraft motion occurs simultaneously in six degrees of freedom: three translatory and three rotatory ones. In each degree of freedom, according to Newton's law, one differential equation of second order is required in order to model the dynamical behavior. So 12 state variables are necessary to describe the rigid body motion.

An aircraft is controlled by selecting four control variable time histories: elevator for pitch and altitude, aileron for roll, rudder for yaw and sideslip, and throttle for thrust level control; in addition, diverse flaps may be set for certain flight regimes (takeoff and landing). To direct such a vehicle in a well controlled maneuver requires skill and concentration even for a trained human pilot; he has to acquire this capability in an extended learning process.

Exactly this knowledge, coded in differential equations as side constraints to the development of trajectories, should be available to an automatic system for recognizing and controlling landing approaches by machine vision.

Simultaneously exploiting spatial and temporal models as shown in section 4 and Figure 6, Eberl (1987) has shown that the problem of controlling landing approaches by computer vision may be tackled successfully relying on present-day microprocessors. In a six degree of freedom fixed base simulation with real-time image sequence processing hardware in the loop, complete landings starting from 2 km distance have been performed fully autonomously with airspeed V being the only quantity not determined from vision. It seems unlikely that such a complex task can be handled by computer vision without using integrated spatio-temporal process models.

Space does not allow us to go into more detail here. A somewhat more extended discussion of machine vision for flight vehicles is given in Dickmanns (1988) containing a summary of the dissertation (Eberl 1987) in English.

6. Development Perspectives

The following principles have been found to be essential for real-time dynamic machine vision.

1. Tangency detection by correlation of linear contour elements
2. Active control of the viewing direction, both top-down (feature search, mode switching) and bottom-up (feature tracking, fixation), coupled with nonuniform image processing
3. Analysis by synthesis, that is, building up internal spatio-temporal representations via 4-D models, the parameters of which are adjusted exploiting the sensor data input.

They will be discussed in the sequel with respect to future applications in machine vision.

6.1 Tangency Detection for Shape Recognition

In Kuhnert (1988) it has been shown that by performing interpolations over the correlation values of shifted and rotated bar masks of about 10 pixel length, angular resolutions of about 1–2 deg and edge position localizations of subpixel resolution can be achieved. Taking this type of data as the basic input for shape description in differential geometry terms, a very efficient signal-to-symbol transition has been proposed in Dickmanns (1985a, 1985b). Using local coordinates only, by simple weighted summing and differencing of the slopes, the two parameters of a linear curvature element can be determined. For direction changes smaller than about $\pi/6$ (30 deg), this element closely corresponds to a spline curve element of third order. Whole contours may be pieced together and corners can be incorporated as curvature impulses. These can be isolated by locating slope discontinuities using the same tangency operations.

This representation is by itself position and rotation invariant. It can be made scale invariant by normalizing the contour length; this may be achieved by some natural scale length, such as normalizing the contour length to the range 0, 1 (or 0, 2π) or dividing it by the square root of the area enclosed. The resulting very compact representation of smooth contours has been termed normalized curvature function (NCF). The interesting point is that NCFs yield a nice basis for shape idealizations, symmetry detection, and the definition of shape terms (e.g., circle, n angle, concavity). Complex features of a 2-D object may be decomposed easily on a local basis into an aggregation of simple edge element features that will allow object tracking and motion interpretation based on edge element tracking using the 4-D model based approach. Thus, it is in combination with point 3 mentioned above, that edge element correlation appears as a powerful tool in 3-D dynamic scene analysis.

The hardware under development presently for low level vision will increase performance by two orders of magntidue for correlation based feature extraction, yielding the capability of measuring many dozens of edge element features per video cycle by a single processor. This will allow us to tackle visually much more complex scences.

6.2 Active Gaze Control

When egomotion and object motion occur simultaneously, signals from inertial sensors may help considerably in discriminating the motion components. High angular relative speeds will lead to motion blur. It may result from two components: egomotion and/or object motion. Angular egomotion may be compensated by stabilizing the camera platform inertially. Angular velocities due to external object motion may be cancelled for the imaging process by active feature tracking and corresponding viewing direction control by the camera platform. This allows (close to static) shape analysis in the image and motion measurement via the platform orientation.

The tuning of the feedback control loops both with inertial and visual signals is presently under investigation in a simulation facility. The quality of stabilization achievable will influence the range of the teleoptics useful with the second camera (for high resolution). During periods when the viewing direction is quickly changed, the image data are blurred and evaluation has to be suppressed; in these short intervals, motion control is done purely based on the internal models. Experience will have to show what are the best strategies for compromising foveal fixation and the consequent motion in the wide angle image.

6.3 Representing the World by a Servo-Maintained Internal "World 2"

The basic feedback approach chosen (see bottom, Figure 4), using integral spatio-temporal world models including perspective (forward) projection, may be expanded by incorporating active gaze control and long term memory for models. A somewhat unconventional block diagram of such a system is shown in Figure 7. The basic arrangement is the same as in Figure 4; the viewing direction control has been introduced as the central horizontal bar. It actively controls the camera pointing and takes care of the corresponding geometrical transformations. For orientation relative to the dominating gravity vector, inertial and other conventional sensors (upper left) are being used, yielding a stabilized internal 4-D world model (center right). The actual models instantiated there are installed by a hypothesis generator (center top) basing its decisions on bottom up feature data and on rules implemented for this purpose. Models for both shapes and motion processes may be selected from a long term memory called model store (upper right). Instantiations are first done separately per object (second block from top, upper right) and then integrated into the world model as a representation of the situation as recognized by the interpretation process (world 2).

If the interpretation process has a set of goals against which it checks the situation recognized, it may start or continue action planning in order to achieve the goals. Actions may be both attention focussing through active vision (lower center and right) taking other sensory modalities into account

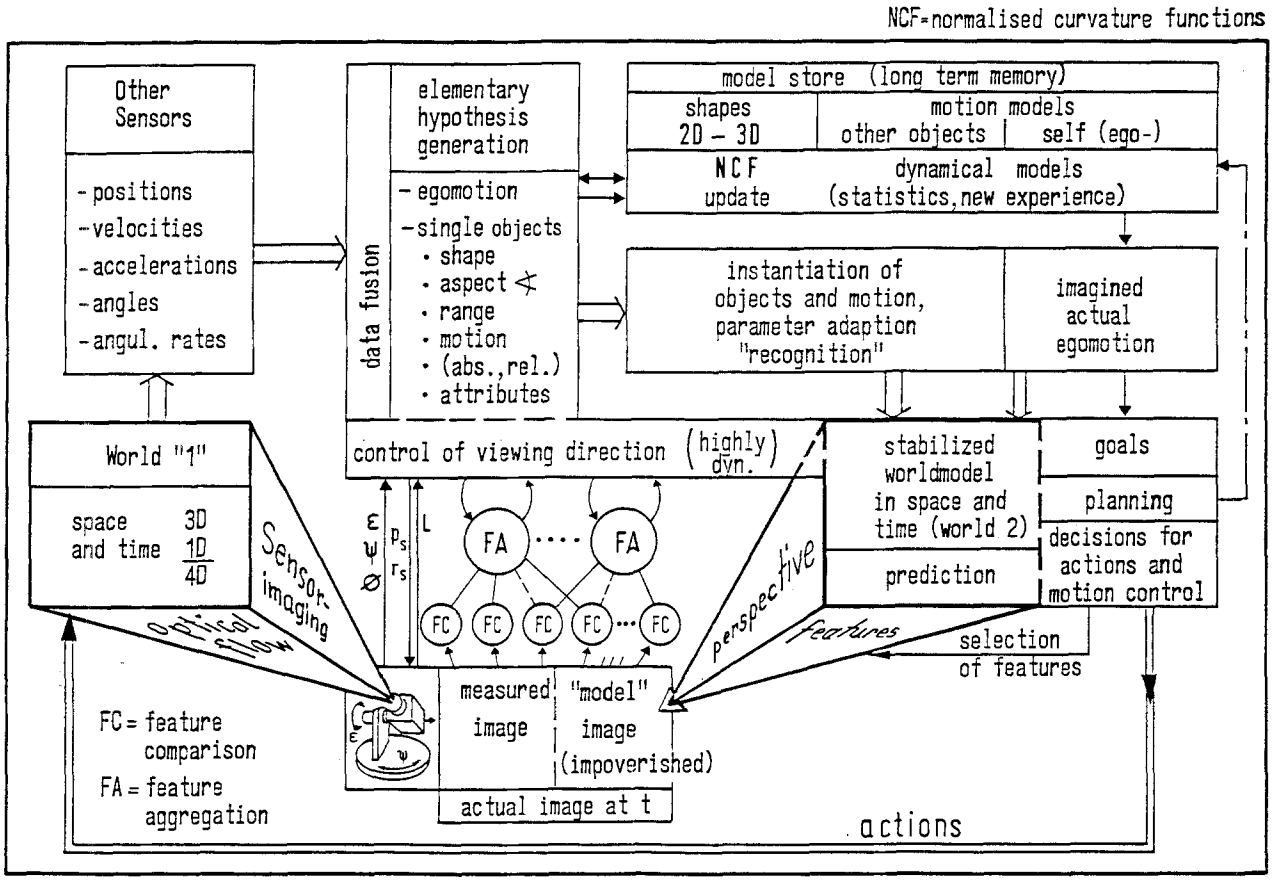

Figure 7. Block diagram of 4-D feature based vision concept including active gaze control, long term model storage, and goal driven activity planning.

(upper left), and application of motion control through effectors (bottom). Since dynamical models are available, direct state variable feedback may be used in order to achieve reflex-like behavioral competences. Thus, for example, in road vehicle guidance, lane-keeping and proper speed control may be achieved without continuously running cumbersome planning activities. Monitoring subprocesses just have to provide "road recognized" and "road free of obstacle" signals. As long as these are true and the goal is not yet achieved, the system continues in this mode. All logical variables required for mode continuation form the set of continuation control tags.

Their value, in turn, may be changed either by sensory data including situation variables derived therefrom or by decisions taken in the continuously active mission planning and monitoring subprocesses. Depending on the particular continuation control tag becoming false, specific other behavioral modes with proper sensing activities and feedback control laws, if necessary adaptable by situation dependent parameters, may be invoked, taking care of a gradual transition from the old mode to the new one.

A sufficiently rich set of behavioral modes including smooth transitions has to be developed and stored in long term memory. In addition, knowledge has to be implemented in the interpretation process as to which behavioral competences should be activated with which set of parameters, depending on the situation and the goals to be achieved.

In the long run, the system should be able to learn from statistics it accumulates during each mission. This is, however, far off in the future.

The systems we have developed up to now only have very simple reflex-like behavioral competences. Some interesting questions arise when we try to imagine what kind of behavior much more complex systems might display (in a not very near future), if they continue to be based on the general principles explained in the previous sections.

The actual world 2 instantiated in the interpretation process is forced to remain close to the real world by critical feature comparison and corresponding model adaptation based on the measured image data and the data from other real-world sensors (left column in Figure 7). What could happen if all these sensory inputs would be cut off and the central and right blocks would continue working on

their own? Would the system be "daydreaming"? Exciting fields of research may open up with respect to the general problem of cognition.

If a model generator could be added on top of the upper right corner, capable of creating new shape models and new motion models including new feedback rules for the generation of new behavioral modes, what would be the benefits for adpating to a changing real world? Certainly, there will have to be some capability for critical evaluation, that is, distinguishing useful models that can catch some part of reality from "phantasmal" ones; otherwise the system may exhibit "idiotic" behavior.

Can a very much refined model of this basic type serve as a model for studying biological intelligent systems? Surprisingly enough, the basic paradigm of the modern school of philosophy named "hypothetical realism" (Vollmer 1975) is consistent with this scheme: "A recognizes B as C." The interpretation process A comes to the conclusion that the sensory data on some object or event B in the real world are identical or similar to those which an object or event with the internal representation C would yield; therefore, B is considered to be what C means in the framework of the internal world 2.

The distinction between worlds 1 and 2 used here corresponds to (and was in fact named after) the terminology introduced by the philosopher Karl Popper for clarifying the semantics in the usage of the word "world" (Popper 1977).

The idea of strictly separating the world one talks about in philosophical discussion or in everyday conversation from the real process everybody is a part of, was first introduced by the philosopher A. Schopenhauer almost 180 years ago (Schopenhauer 1819) in trying to lay a new foundation for modern philosophy after Kant's "Critiques" and the upsurge of German "Idealismus" (Flichte 1792, Hegel 1806).

It is a surprising experience for an engineer and a scientist, that a "world as an internal representation" (Schopenhauer) is not only technically implementable on today's computers (admittedly in a very crude form), but is numerically very efficient for motion control through machine vision. It may be that methods in high performance graphics, like 3-D animation, can contribute to this line of development.

Acknowledgments. The support by Madeleine Gabler and Sigrun Hausmann in preparing this manuscript is gratefully appreciated.

References

[The list of references is given at the end of the companion paper—see pp. 260–261.]