IAPR Workshop on CV - Special Hardware and Industrial Applications OCT.12-14, 1988, Tokyo

A HIGH SPEED IMAGE PROCESSING SYSTEM UTILIZED IN AUTONOMOUS VEHICLE GUIDANCE

Volker Graefe and Klaus-Dieter Kuhnert

Institut für Meßtechnik
Universität der Bundeswehr München
8014 Neubiberg, West-Germany

Abstract

Concepts for guiding the design of hardware structures for dynamic vision are presented. The most important ones are the utilization of temporal coherence of natural dynamic scenes by evaluating every single image produced by a TV-camera, and a high degree of flexibility in allocating resources within the sytem dynamically in a data dependent way. A multi-microprocessor system, BVV 2, implementing these ideas, has been realized. Its application in guiding an autonomous road vehicle at high speed is described.

Introduction

Robots and autonomous vehicles, if they are guided by computer vision, place very demanding requirements on the performance of their vision systems. In fact, the performance of the vision system is often the limiting factor for the performance of the entire system.

In industrial application autonomous robots and vehicles must frequently operate in an environment designed for people and interact with them. To minimize the dangers and annoyances resulting from such a co-existence and interaction of men and robots it will probably be advantageous to develop machines having similar reaction times and speeds of motion as people. This means that, when such a system is supposed to react to an external event, the delay time between the event and the beginning of the reaction should be less than 1 s and, preferably, not much longer than 0.1 s.

Considering the state of the art of computer vision systems this is not an easy task. One way to gain the necessary speed, is to increase the sheer processing power of the hardware. This paper describes, however, a totally different approach, based on a system architecture which is so well matched to the tasks of low- and intermediate-level dynamic vision that even standard micro-computers are a sufficient basis to process dynamic scenes in real time.

Dynamic vision

The interpretation of dynamic scenes is simplified to a high degree if the temporal coherence, which is a characteristic of almost all natural scenes, is utilized. To do this, the scene must be sampled at a sufficiently high frequency, depending on the dynamics of the objects in the scene.

The sampling is usually performed by a TV-camera at a rate of 25 to 60 images per second, depending on the TV standard used. TV standards have been defined in accordance with the abilities of the human eye which cannot detect any discontinuity of motion in a sampled scene, for instance in a movie, if the image frequency is higher than about 20 Hz. If the goal is to design a vision system of similar temporal resolution as the human eye the application of a standard TV camera and the evaluation of every image it produces seem a sensible approach.

Most material objects which are visible in any natural scene have a significant mass. Therefore, their motions are typically smooth and of limited acceleration. When such a scene is projected on a 2-D image plane, often the motions in the image plane are smooth, too. If the scene is sampled by a TV-camera subsequent images will normally be very similar, since the sampling interval is sufficiently short. Exceptions to this general rule exist, e.g. lights being switched on or off, or explosions. Apart from such relatively rare events the next image in the sequence can be predicted fairly well on the basis of the previous ones. In fact, a simple zeroth order prediction, stating that the next image in the sequence will be identical to the present one, is usually a good basis for starting the interpretation of the next image.

This means that almost perfect knowledge of the new image is available before its interpretation starts. The only problem remaining to be solved is to correct the <u>small</u> prediction errors related to the expected locations of a limited number of relevant features in the image. This is much easier than the interpretation of a totally unexpected image, and it requires much less computation. In other words, dynamic vision is relatively easy, provided <u>every</u> image generated by the TV-camera is evaluated and the thus available knowledge is, indeed, applied.

It may be concluded that exploiting the temporal coherence of natural scenes is a key factor in dynamic vision, and a dynamic vision system should, therefore, be able to analyze images as fast as they are produced by the camera. If it does not, the task to be solved becomes much more difficult.

Hardware concepts for dynamic vision

Low- and intermediate-level vision is a very computation intensive task. No ordinary computer is able to perform this task in real time; the necessary computation would require many seconds or even minutes of time per image and, in addition, such computers are unable to accept a continuous stream of pixel data at a rate of several megabytes per second.

Special hardware is, therefore, necessary for real-time dynamic vision. Some of the basic concepts which have guided the design of such a hardware shall be discussed. The important point is here that, by analyzing the nature of the tasks to be performed in dynamic vision, a design has emerged which is well matched to these tasks and, therefore, has led to a very powerful vision system based on standard microcomputers, augmented by some more or less conventional electronics.

Parallel processing

A multi-processor system can be much more powerful than a single processor, but such a system is efficient only if its structure matches the structure of the tasks for which it is meant to be used. The (generally unsolved) problem is an adequate partitioning of a task into subtasks which may then be shared between the several processors. If the task is vision, a good and commonly accepted approach is to divide the entire vision system conceptually into layers or levels and assign (at least) one processor to each layer (low-level, intermediate level, high level, etc.) This results in a pipeline structure with probably not more than about 2 or 4 stages.

This degree of parallelism is insufficient, particularly for the low and intermediate levels (the distinction is not always very clear) of a real-time vision system. However, within these levels different kinds of parallelism may easily be realized:

a) The image may be divided into regions and all regions may be processed in parallel by having a separate processor for each region. The degree of parallelism may be very large if many regions are defined.

b) The image (or a certain part of the image) may simultaneously be analyzed with different algorithms, where each algorithm is implemented on a separate processor (e.g. one processor searching for vertical edges, another one for a specific type of texture, etc.). Theoretically, the degree of parallelism may be very large here, too, but it would require the conception, implementation, and test of a correspondingly large number of algorithms.

Of course, both types of parallelism may be combined by defining regions and then analyzing each region with more than one algorithm.

Dynamic allocation of resources

When dividing an image into regions, a very naive approach is sometimes taken by imposing a fixed pattern of regions upon the image, regardless of its content. For instance, the image may be divided into fixed rectangular fields of m*n pixels, or each column, or each line in the matrix of pixels, or even each pixel, may be assigned one processor. While this approach is easily implemented in relatively simple and repetitive hardware structures, it leads to gross inefficiencies in processing. There are two reasons for this, both of which are related to the inflexibility of the approach: inability to adjust the size and location of a region dynamically in a data dependent way, and inability to allocate the available resources dynamically in a data dependent way. This will be explained in the sequel.

Inflexible partitioning of the image: The main task of low-and intermediate-level vision is feature extraction. Features may, for instance, be entire contours of objects or parts of such contours, and the presence or absence of such features is to be detected, and, if they are present, their location and visual appearance are to be determined. To extract a feature from an image a certain environment of the potential feature, a "context", must be available for analysis. The sizes of relevant features and their necessary contexts vary, and, in dynamic vision relevant features normally move in the image. It is, therefore, not to be expected that, if a fixed partitioning of the image is used, each relevant feature including its necessary context will always be contained entirely in one of the partitions. Rather, the borders of the predefined partitions will arbitrarily dissect features (similar to a puzzle), and, if a feature moves, it will be dissected in a time varying way. This artefact makes feature extraction difficult.

Inflexible allocation of resources: In dynamic vision, when it is used for controlling the motions of robots or vehicles, not all parts of an image contain relevant information. Usually, a limited number of features must be tracked, which is simplified by the fact that the locations where they will appear in the next image can be predicted with good accuracy from previous images. It is then unnecessary and inefficient to use precious resources, like processors and communication channels, for analyzing regions of the image which are known beforehand to contain no relevant features. Experience with dynamic vision shows that frequently at the most 10% of the image area contain all relevant features and their context. If the design of a vision system does not allow to concentrate its computing power on those parts of the image which actually contain relevant information in that moment, this inflexibility alone may thus reduce the efficiency of the system by more than 90%.

Even worse is the inefficiency of so-called SIMD machines which have many processors acting in parallel, each one on its own data, but with the restriction that all processors must always execute the same instruction. It has been advocated to use such systems for low-level vision and to allocate one processor to each pixel. In this case all pixels must be processed according to the same algorithm. Experience indicates that typically only about 1% of all pixels, at the most, should be processed by any specific piece of program, therefore, less than 1% of all processors of such an SIMD machine would perform useful work at any given moment.

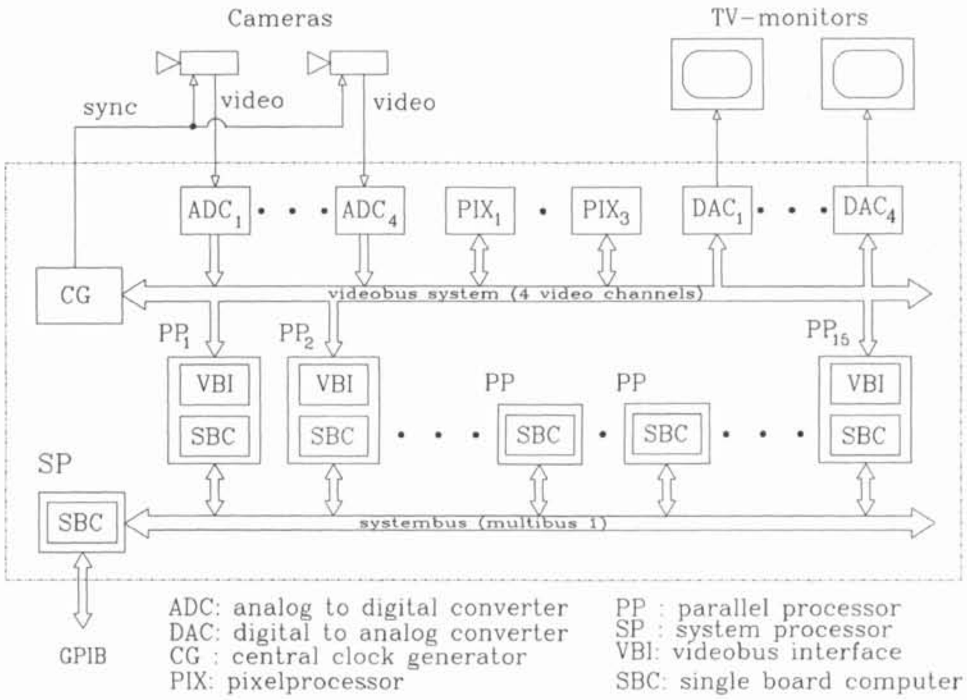
It may be concluded that flexibility in partitioning the task and in allocating resources is of utmost importance for a dynamic vision system.

The dynamic vision processor BVV 2

A multi-processor system for dynamic vision, the BVV 2, has been designed and built according to these concepts (Graefe 83; Dickmanns, Graefe 88a, b). Figure 1 shows a block diagram of the BVV 2.

Video data are digitized and displayed by specialized circuit boards (A/D and D/A converters). The data are distributed via a video bus system containing 4 video channels and thus allowing up to 4 different image sequences to be distributed simultaneously, e.g. from multiple cameras. A main clock generator produces the various necessary timing signals. It is adaptable to several video standards (CCIR, EIA, and others).

The main image processing parts of the system's hardware are up to 15 parallel processors, PP. Each of these consists of a standard single-board computer



GPIB: General purpose interface bus connecting the BVV 2 to other equipment, e.g. a host computer

Figure 1
Architecture of the image processing system BVV 2

(based on the microprocessor Intel 8086), SBC, and an optional video bus interface, VBI. This board contains a local image memory which is organized according to the double buffer principle with two memory banks allowing image acquisition and processing of the data to proceed in parallel.

All processors may independently have access to the video data via their video bus interfaces. No arbitration is required on the video bus, since the video bus interfaces only read data from the bus. Each video bus interface is able to capture a rectangular, and optionally subsampled, region of the image from the stream of video data. The size of the region is presently limited to 4 K pixels, its position in the image may be controlled dynamically by the microcomputer. The distributed image memories are provided to avoid a bottleneck that would otherwise exist if many processors had to share a common image memory.

Each parallel processor is assigned to one freely movable region in the image. It is of no concern if the regions belonging to different processors are overlapping, and conceivably all processors could simultaneously process the same small region in the image, each one with a different algorithm. Normally, however, each processor detects and tracks a specific feature, or a group of features, trying to keep the moving feature approximately centered in its region.

The system bus used for inter-processor communication is a standard multibus 1, managed by the system processor, SP, also a single board computer (8086). The system processor contains an IEC-bus interface as a link to a host computer or other measuring equipment.

Depending on the application, some of the processors in the system may have tasks not requiring direct access to video data, e.g. combining results of other processors into more complex scene descriptions or controlling a camera platform. Such processors do not need a video bus interface.

The design of the BVV 2 was begun in 1983. It is currently used for experiments in dynamic vision and autonomous mobility. A successor, the BVV 3, has been designed according to the same basic concepts and is currently under development (Dickmanns, Graefe 88a; Kuhnert, Graefe 88). It is expected to be more powerful by about 2 orders of magnitude.

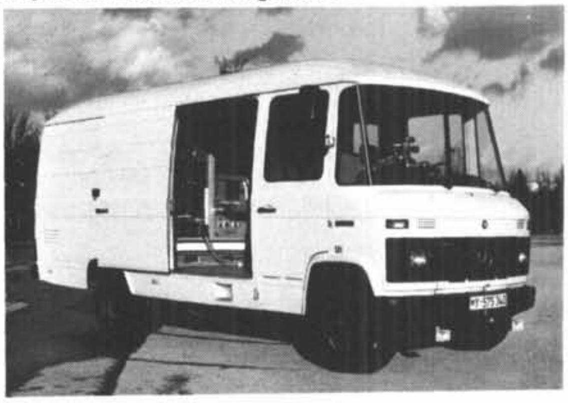


Figure 2
The experimental vehicle of the UniBwM: "VaMoRs"

Vehicle guidance as an application of the BVV 2

The dynamic vision system BVV 2 has been utilized in several applications, e.g. in the autonomous road vehicle VaMoRs (figure 2) (Dickmanns 87; Dickmanns, Graefe 88a, b; Zapp 88) which has been running at speeds of up to 96 km/h on unobstructed highways without traffic and of 50 km/h on campus roads, faster than any other autonomous road vehicle. Also, an automatic landing approach of a vision guided airplane (Eberl 87) was realized with the BVV 2 (real-time simulation with hardware in the loop).

To give an idea of how the BVV 2 may be used in an application, the algorithms developed for recognizing and tracking the edges of a road for an autonomous vehicle will be discussed in detail. The task is to detect and track edge-elements of the road border in natural scenes in real time. As has been pointed out before, preferably every image produced by the camera should be interpreted (60 images per second).

After an investigation of different feature extraction methods for natural scenes, the method of "controlled correlation", a generalization of the standard correlation method, was developed (Kuhnert 86, 88).

Correlation has some similarities to the human visual system and, as a very essential feature when processing natural scenes, a good noise immunity. In communication theory it is proven that, for detecting a signal buried in additive ergodic, normal distributed white noise, correlation is the optimal linear detection scheme. Even if these conditions are not exactly met, correlation exhibits a very robust behavior.

There are two problems when using correlation for a real-time application: the heavy computational load and the many irrelevant maxima of the correlation function (false responses).

In the sequel the method of controlled correlation will be described. In a special implementation it needs only 15 ms per image to localize and track 3 elements of a road border, permitting 60 images per second to be evaluated. This rapid processing also alleviates the problem of false responses by making it possible to utilize knowledge gained from preceding images to resolve ambiguities.

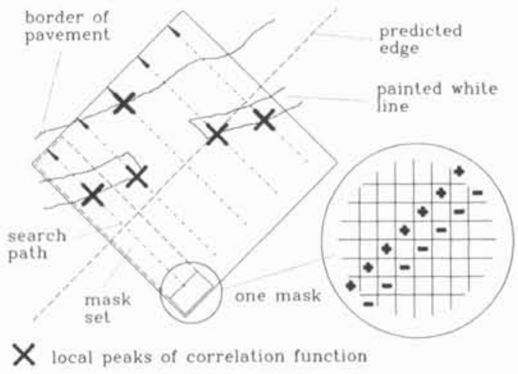


Figure 3
Prinicle of road boundary detection

Figure 3 illustrates the method. From previous images the expected location of the road edge is predicted. Only a small region (typically about 2% of the image), centered on the expected location is processed. The correlation function is computed along straight lines ("search paths") perpendicular to the predicted direction of the road edge. Computing the correlation function on a few short search paths only, rather than in the entire image, reduces the required effort by 3 orders of magnitude.

To further reduce the computational cost of controlled correlation, sparsely populated masks may be used which contain many zero elements. Since the algorithm is implemented in such a way that zero elements do not cause any operation whatsoever, using such masks results in a significant reduction of computational cost.

The microprocessors used in the BVV 2 add much faster than they multiply. Therefore, ternary masks are used, containing only values of -1, 0, and +1; all multiplications may then be replaced by additions. Experiments indicate that, when sparsely populated ternary masks meet certain symmetry conditions, the desirable properties of standard correlation, especially its noise immunity, are conserved to a large extent.

In the autonomous vehicle driving experiments 2 to 6 copies of the above algorithm were simultaneously executed by 2 to 6 parallel processors. The road border information was fed to an IBM-AT-compatible computer connected to the BVV 2 via an IEC-bus. It was used to maintain an internal 4-D world model which was updated 10 times per second (Dickmanns 87; Dickmanns, Zapp 87).

Acknowledgement: Part of this work has been funded by the Ministry of Research and Technology (BMFT) and by the Daimler-Benz AG.

References

Dickmanns, E.D. (87): 4D-dynamic scene analysis with integral spatio-temporal models. In R. Bolles (ed.): 4-th International Symposium on Robotics Research, Santa Cruz, MIT Press.

Dickmanns, E.D.; Graefe V. (88a): Dynamic Monocular Machine Vision. To appear in Machine Vision and Applications.

Dickmanns, E.D.; Graefe V. (88b): Applications of Dynamic Monocular Machine Vision. To appear in Machine Vis. and Applications.

Dickmanns, E.D.; Zapp, A. (87): Autonomous High Speed Road Vehicle Guidance by Computer Vision. Preprint, 10th IFAC-Congress, Munich, Vol. 4, pp 232-237.

Eberl, G.(87): Automatischer Landeanflug durch Rechnersehen. Dissertation, Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München.

Graefe, V. (83): Ein Bildvorverarbeitungsrechner für die Bewegungssteuerung durch Rechnersehen. In H. Kazmierczak (Ed.): Mustererkennung 1983, NTG Fachberichte, VDE-Verlag, pp 203-208.

Kuhnert, K.-D. (86): A Model Driven Image Analysis System for Vehicle Guidance in Real Time. In Proceedings of the Second International Electronic Image Week, CESTA Nice, pp 216-221.

Kuhnert, K.-D. (88): Zur Echtzeit-Bildfolgenanalyse mit Vorwissen. Dissertation, Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München.

Kuhnert, K.-D.; Graefe, V. (88): Vision Systems for Autonomous Mobility. IEEE International Workshop on Intelligent Robots and Systems, Tokyo, Nov. 88.

Zapp, A. (88): Automatische Straßenfahrzeugführung durch Rechnersehen. Dissertation, Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München.