Cellular Multiadaptive Analogic Architecture: A Computational Framework for UAV Applications

Csaba Rekeczky, *Member, IEEE*, István Szatmári, *Member, IEEE*, Dávid Bálya, *Member, IEEE*, Gergely Tímár, *Student Member, IEEE*, and Ákos Zarándy, *Member, IEEE*

***Abstract*—An efficient adaptive algorithm in real-time applications should make optimal use of the available computing power for reaching some specific design goals. Relying on appropriate strategies, the spatial resolution/temporal rate can be traded against computational complexity; and sensitivity traded against robustness, in an adaptive process. In this paper, we present an algorithmic framework where a spatial multigrid computing is placed within a temporal multirate structure, and at each spatial grid point, the computation is based on an adaptive multiscale approach. The algorithms utilize an analogic (analog and logic) architecture consisting of a high-resolution optical sensor, a low-resolution cellular sensor-processor and a digital signal processor. The proposed framework makes the acquisition of a spatio-temporally consistent image flow possible even in case of extreme variations (relative motion) in the environment. It ideally supports the handling of various difficult problems on a moving platform including terrain identification, navigation parameter estimation, and multitarget tracking. The proposed spatio-temporal adaptation relies on a feature-based optical-flow estimation that can be efficiently calculated on available cellular nonlinear network (CNN) chips. The quality of the adaptation is evaluated compared to nonadaptive spatio-temporal behavior where the input flow is oversampled, thus resulting in redundant data processing with an unnecessary waste of computing power. We also use a visual navigation example recovering the yaw–pitch–roll parameters from motion-field estimates in order to analyze the adaptive hierarchical algorithmic framework proposed and highlight the application potentials in the area of unmanned air vehicles.**

***Index Terms*—Adaptive computing, analogic cellular neural network (CNN), cellular computing, UAV, vision system.**

I. Introduction

In designing a fully functional vision system for both unmanned air vehicles (UAVs) and unmanned ground vehicles (UGVs), one of the most difficult problems the engineer faces is the continuously changing speed of the projected world acquired by the optical sensors of the system. In extreme cases, this variable speed could span over several orders of magnitudes and accordingly set the requirement for real time processing in a very broad range. For instance, the "artificial eyes" of a UAV flying at very high altitudes might only sense the changes of the surrounding world in subpixel resolution at 1 frame/s image-acquisition speed, while moving very close to the ground in some cases not even 1000 frames/s could be adequate to detect the quickly approaching obstacles. The example highlights at least two important consequences.

1) Real-time processing could only be defined as a process/event related requirement.
2) An efficient vision system should have a certain degree of spatio-temporal adaptation capability.

In this study, we outline an analogic architecture and the associated algorithmic framework capable of handling extreme variations (relative motion)[1] in the perceived environment in a process/event adaptive manner. In the course of this work, our primary motivation has been to develop a powerful and flexible algorithmic structure into which efficient terrain identification, navigation, and multiple object detection/tracking modules could be embedded.

II. Bioinspired Multitask System Architecture

The system description of the introduction envisions a platform with multitask processing capability efficiently combining topographic and nontopographic computing. Different versions of such architecture called the compact cellular visual microprocessor (COMPACT CVM) have been designed (Fig. 1) [19], and are currently being tested. These standalone vision systems largely build on the results of cellular neural networks (CNN) [1]–[4]) theory, CNN-chip prototypes (e.g., [9]–[13]), retinal analysis and modeling (e.g., [5]–[8]), and various bioinspired (Bi-i) topographic schemes developed for image-flow processing (e.g., [14]–[17]).

In the sequel, the COMPACT CVM will be introduced along with all major system components and functional block diagrams. The Bi-i[2] COMPACT CVM architecture (Fig. 1) builds on the state-of-the-art CNN-type (ACE16k) and digital signal processing (DSP)-type (TEXAS TMSC6x) microprocessors and its algorithmic framework contains several feedback and automatic control mechanisms between different processing stages (Fig. 2). The architecture is standalone and with the interfacing communication processor, it is capable of a 100 Mbit/s information exchange with the environment (over TCP/IP). The COMPACT CVM is also reconfigurable, i.e., it can be used as a monocular or a binocular device with a proper selection of a high-resolution CMOS sensor (IBIS 5)

Manuscript received August 18, 2003; revised January 9, 2004. This work was supported in part by the Hungarian National Research and Development Program TeleSense Project under Grant NKFP 035/02/2001, and in part by the National Aeronautics and Space Association and Jet Propulsion Laboratory under Contract 1 248 805. This paper was recommended by Guest Editor B. Shi.

The authors are with the Analogical and Neural Computing Laboratory, Computer and Automation Institute, Hungarian Academy of Sciences, H-1111 Budapest, Hungary (e-mail: rcsaba@sztaki.hu).

Digital Object Identifier 10.1109/TCSI.2004.827629

[1]The current study does not deal with adaptive visual sensing in response to changing and inhomogeneous illumination in the environment.

[2]"***Bi-i***" is also the brand name of the system (see [19]).

1057-7122/04$20.00 © 2004 IEEE

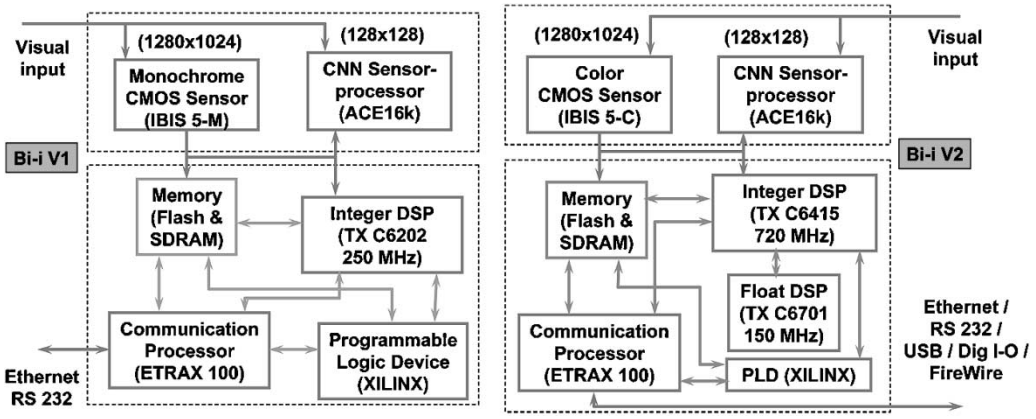

Fig. 1. Main hardware building blocks of the Bi-i COMPACT CVM architecture. The figure shows two different versions of this standalone system, both consisting of a sensor platform and a control platform. The first prototype (on the left: Bi-i V1) contains a high-resolution monochrome CMOS sensor array (IBIS 5-M), a low-resolution cellular sensor-processor (ACE16k), an integer digital signal processor (TX C6202) and a communication processor (ETRAX 100). In contrast, the second prototype (on the right: Bi-i V2) has a color CMOS sensor array (IBIS 5-C) and two high-end digital signal processors (TX C6415 and TXC6701). While both systems run an embedded Linux on the communication processor, Bi-i V2 has more complex external interfaces (USB, FireWire, and a general digital input/output in addition to the Ethernet and RS232 built into the Bi-i V1).

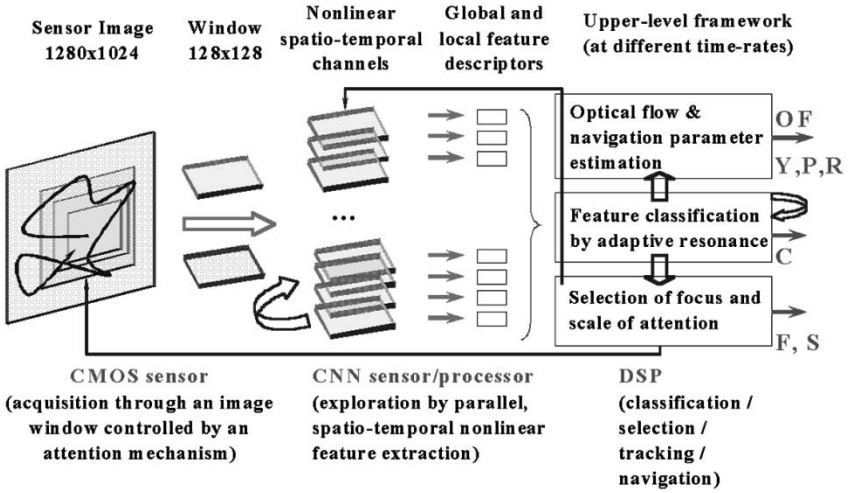

Fig. 2. General flow-processing diagram of the COMPACT CVM algorithmic framework. The image frame acquired by the CMOS sensor array is never evaluated at full resolution. Only a few subframes are processed identical to the resolution of the CNN sensor processor (the figure shows a single subframe acquisition in the active fovea of the system controlled by an upper level attention mechanism). In the first step of the evaluation, the selected image window is processed by topographic parallel nonlinear spatio-temporal algorithms. Then, local and global features are extracted and forwarded to the upper level where nontopographic calculations take place at different time rates. The peculiar property of the entire scheme is the existence of several feedback mechanisms between different levels of processing that makes it possible that top-down and bottom-up processing is combined within this framework.

and a low-resolution CNN sensor-processor (ACE16k). The high-resolution sensor could be used as a random access device (with built-in 1:2 digital zoom) acquiring CNN chip-size images from algorithm controlled positions at different time instants. With these two sensors having the same field of view, one can also design algorithms that make use of three integer numbered spatial scales (roughly 1:2:8 at 128×128 spatial resolution) while navigating in a 1280×1024 high-resolution projection of the surrounding world at a temporal rate of up to a few thousand frames per second.

Fig. 1 demonstrates two versions of COMPACT CVM systems (Bi-i V1 and Bi-i V2). The main difference between these prototypes lies in the complexity of their (functionally similar) hardware components: Bi-i V2 has a color CMOS sensor array, two different high-end DSPs for integer and float computations and more complex external interfaces than Bi-i V1.

The general algorithmic framework of the COMPACT CVM architecture incorporates an upper-level visual attention and selection mechanism adjusting the focus and scale (zoom) of processing as drawn and explained in Fig. 2. This mechanism can select a single or multiple windows from the same high-resolution frame at a certain time instant. It is important to note that the video flow is never processed in parallel at full resolution (much like biological systems). This framework allows the system of multitask execution at different temporal rates. The general algorithmic framework incorporates optical-flow estimation, feature classification and automatic attention mechanisms built on topographic nonlinear parallel feature processing performed by the CNN sensor-processor. Several feedback mechanisms are also built in at different processing levels that make it possible that bottom-up computing can be combined with top-down schemes.

In summary, the COMPACT CVM architecture can be attributed as follows (the explanations and the high-end specifications are related to the associated Bi-i V2 systems currently under testing).

- ***Fault-tolerant visual computer***
  The architecture is "fault tolerant" at the topographic cellular processing level, i.e., a low number of cells not functioning properly will not significantly affect the overall performance. This has been tested by masking a certain number of cells during the operation and also by using chips for which the yield has not been 100% for all cells.
- ***High-speed, compact, and (relatively) low-power system***
  The entire system consumes around 10 W (mainly due to the high power consumption of the ACE16k chip in the range of 1–5 W depending on the operating mode). However, it delivers an overall computing power (over 6000 MIPS digital and 1 TOPS analog) that is significantly higher than the performance of any competing standalone vision systems (see comparisons in [19]). In the computer complexity space (*speed-power-volume* measure), it can be stated that the system is relatively "low power" (the volume of the system with packaging and optics is less than 1000 cm^3 and weighs less than 1 kg).
- ***Multitask computing platform***
  This is composed of
  - multiple attention and selection mechanisms ([20]);
  - multichannel nonlinear feature detection-based terrain classification ([21]);
  - multigrid motion analysis-based navigation parameter estimation ([23]–[27]);
  - multiple target detection and tracking ([22]).
- ***Biologically inspired sensor–computer architecture and algorithmic framework***
  The system provides hardware support for both topographic and nontopographic computing at different levels of analysis. This framework makes it possible that a broad class of bio-inspired processing strategies can be developed and tested within a unified framework (see UAV and UGV related works in [28]–[32]).

In this section, we have described the COMPACT CVM architecture and have argued that it is a fault tolerant multitask visual computer and an ideal computational platform for Bi-i exploration, classification, selection, tracking, and navigation. In the sequel, we will describe an associated cellular multiadaptive framework of processing and focus on visual navigation tasks to highlight its capabilities.

III. Hierarchical Structure of Adaptive Spatio-Temporal Algorithmic Framework: Bottom-Up and Top-Down Computing

From the general algorithmic framework of the COMPACT CVM architecture described in the Section II, we select the exploration and optical-flow computing subsystems responsible for the spatio-temporal adaptive nature of the framework. First, the general algorithmic structure will be so built that it flexibly combines bottom-up and top-down computing. Then, a motion parameter-estimation task will be used to highlight how an important upper-level function and algorithmic module could be embedded into this scheme.

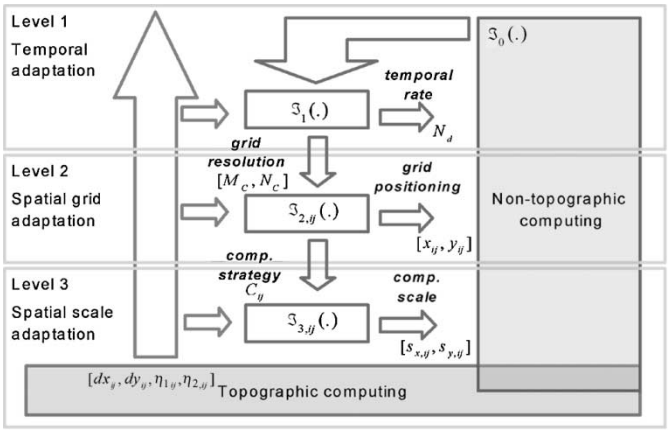

Fig. 3. Hierarchical structure of the three-level adaptive spatio-temporal framework with bottom-up and top-down computing. At the topographic (sensory) level, displacement and feature analysis is performed in parallel and the computed results are forwarded to the upper computational levels (bottom-up route). Nontopographic computing is present at all levels building on various evaluations of the bottom-up flow and directly influencing the topographic computing (top-down route). At the top level (Level 1), depending on the evaluation of the sensory input and the actual state of the top level functions (the tasks computed by the system), the temporal rate or/and the spatial coarse grid resolution is adjusted (in order to achieve the appropriate spatio-temporal sampling). At the mid level (Level 2), depending on the evaluation of the sensory input and possibly on the output of previous adaptation level (L1), the grid positioning and/or the (topographic) computation strategy is adjusted (in order to improve the precision of the displacement computation). At the lowest level (Level 3), dominated by topographic computing, depending on the evaluation of the sensory input and possibly on the output of previous adaptation level (L2), the spatial scale is adjusted in order to increase the information content within the analysis windows (subframes). While any of these adaptation mechanisms can be applied independently they can also be connected at the neighboring levels. When a higher level strategy affects a lower level strategy, the hierarchical structure describes a combined bottom-up and top-down computing scheme. In this process, the top-down mechanism is responsible for spatio-temporal adjustment and precision control of the topographic calculations at the sensory level ("shaping" the bottom-up flow).

The adaptive multirate, multigrid, and multiscale algorithmic framework can be viewed as a hierarchical structure described as follows (Fig. 3).

Level 1: Adaptive Multirate (Temporal) Framework
While Process Frame
  Compute For All Coarse Grid Points (Subframes) → **Level 2**
  Analyze Subframe Computation (Displacement Estimates)
  Drop Frames/Delay Frame Acquisition
  (Perform Nontopographic Calculations)
End Process

Level 2: Adaptive Multigrid (Spatial) Framework
Increase/Decrease Coarse Grid Resolution
Compute For All Coarse Grid Points (Subframes)
  Compute Subframe → **Level 3**
  Analyze Subframe Computation (Confidence Measures)
  Deform Grid/Reposition Grid Points
End Compute

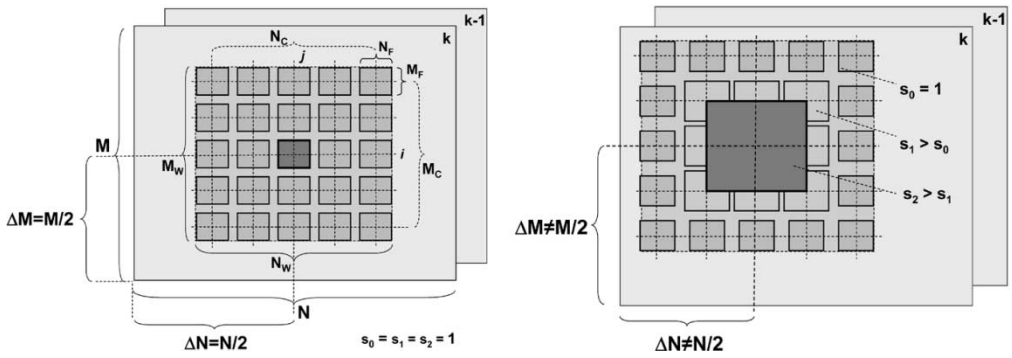

Fig. 4. Cellular multiadaptive framework of computation. The figure shows two typical arrangements with multiple active foveae. In both cases, the full frame has $M \times N$ resolution in which processing is focused on the $M_W \times N_W$ resolution image window. Within this area, an $M_C \times N_C$ uniform coarse grid is defined and the subframes (of $M_F \times N_F$ resolution) representing the active fovea are placed at the node points. The processing window can be nonbiased (left) or biased (right) with respect to the center of the $M \times N$ full frame and the subwindows can also be nonscaled (left) or scaled (right) with respect to the $M_F \times N_F$ default subframe resolution. In all cases, it is assumed that there is a single fovea with an enhanced role (this most active fovea represents the "focus of attention") where a more intensive analysis than within the rest of the subframes, takes place. This computational framework is ideally supported by a hardware architecture having a high-resolution array sensor with random access read out (with possible interpolation) and a low-resolution topographic array processor that can act in parallel on the image areas specified by the subframes. By changing the temporal rate of the full frame acquisition, the resolution/positioning of the coarse grid and the spatial scale/computation of the subframes various adaptation strategies can be defined in order to optimize the available computing power/memory of the underlying hardware–software architecture.

Level 3: Adaptive Multiscale (Spatial) Framework
Change Subframe Computational Strategies (Complexity)
Compute Subframe
  Detect/Extract Features
  Perform Feature Analysis
  Perform Displacement Analysis
  Compute Confidence Measures
  Increase/Decrease Scale
End Compute

The above multiadaptive spatio-temporal framework of computation builds on a controlled "navigation" in a high-resolution video flow with a few (structurally organized) low-resolution windows. This multigrid model is illustrated in Fig. 4 and the relevant specifications are given in the sequel.

The three-level algorithmic framework handles the analysis of the projected visual scene adaptively at $M \times N$ spatial (full) resolution and f temporal rate at some time instant t_0. To explain the underlying computational procedures and to devise both qualitative and quantitative descriptions of the adaptation, we rely on the following definitions.

Within the $M \times N$ spatial resolution (full-image frame), an $M_W \times N_W$ processing window is defined (processed frame), placing the window center at $[\Delta M, \Delta N]$ measured from the lower left corner of the full frame (see Fig. 4). Within this window, an $M_C \times N_C$ uniform coarse grid is defined and each coarse grid node point is associated with an $M_F \times N_F$ fine grid $(M_F < M_W < M;\ N_F < N_W < N;\ M_C < M_W;\ N_C < N_W)$. The coarse grid node locations define a multiple active fovea within the full frame each having an analysis window with a pixel number exactly determined by the fine grid. The coarse grid is "deformable" (all fovea could move independently) and its ijth node is defined by $(x_{ij}, y_{ij}, s_{x,ij}, s_{y,ij})$ representing the spatial locations and zoom parameters in horizontal and vertical directions, respectively (the zoom parameter is inversely proportional to the spatial scale). A fovea always "sees" an $s_y M_F \times s_x N_F$ part of the image at the ijth node. However, the associated area is always mapped (interpolated/extrapolated) to $M_F \times N_F$ during the topographic processing.

The spatial details which can be observed by the vision system are completely determined by its field of view (φ), spatial resolution $M \times N$, and the distance/height (h) from the objects/environment. The temporal rate of the analysis for a full frame is f which can be slowed down or speeded up by changing Δt the interval between two consecutive frames. Since the image acquisition could be either synchronous (clocked) or asynchronous, two different interexchangeable descriptions could be used for characterizing the temporal rate:

1) defining the f_d actual rate one should specify the number of frames N_d to be dropped compared to a maximum $f_{\max}$ temporal rate;
2) defining the time delay t_d between two consecutive frames.

In the latter case, obviously, $f_d = 1/t_d$ and in a synchronous mode $t_d = N_d \Delta t_{\min}$, where N_d is an integer number and $\Delta t_{\min}$ is the smallest time interval of temporal image sampling $(f_{\max} = 1/\Delta t_{\min})$.

As outlined in the introductory description of the three-level computational framework, the spatio-temporal adaptation could be based on displacement and feature analysis at each coarse grid location. In our specific case, making optimal use of the available computing hardware (a fast cellular nonlinear/neural net and a digital signal processor) we have chosen "feature-based" region displacement and histogram analysis methods (to be described in the subsequent section) in order to compute the optical-flow estimates with two relevant confidence measures $(dx_{ij}, dy_{ij}, \eta_{1,ij}, \eta_{2,ij})$. Computing these four-element vectors at each coarse grid node makes it possible to define efficient adaptive procedures at all hierarchical levels. A general strategy and notation for hierarchical self-adaptation is described in the sequel (a specific strategy within this framework using the same notation is given in the section discussing the qualitative control framework).

Let us first define a vector–scalar function checking (evaluating) various properties of the topographic calculations at a time instance t_0 and call it a "testing function" (the vector variables in the formula are composed of all associated local values at the coarse grid nodes using the same ordering for different components)

$$\Im_m(\Im_{m-1}(\cdot), \underset{\rightarrow}{dx}, \underset{\rightarrow}{dy}, \underset{\rightarrow}{\eta_1}, \underset{\rightarrow}{\eta_2}), \qquad m = 1, 2, 3 \tag{1}$$

where m is the computational level. As can be seen, this testing function could depend on a similar function of the preceding (higher) computational level (note that $m = 0$ is also meaningful since that could refer to an error measure of the top level function, see Fig. 3). A testing function can either be local (depends on the variables at the ijth coarse grid node) or global (depends on the variables at all coarse grid nodes). Now, we can proceed to define the general framework of the spatio-temporal adaptation using the testing functions.

Adaptation at level 1 (global)

$$\Im_1(\underset{\rightarrow}{dx}, \underset{\rightarrow}{dy}, \underset{\rightarrow}{\eta_1}, \underset{\rightarrow}{\eta_2})_{|\vartheta_1, \vartheta_2, \vartheta_{v\max}, \vartheta_{v\min}} \Rightarrow \{(M_C(\Im_1), N_C(\Im_1)); N_d(\Im_1)\} \tag{2}$$

where ϑ_1 and ϑ_2 are the confidence thresholds and $\vartheta_{v\max}$ and $\vartheta_{v\min}$ stand for the minimum and maximum magnitude of the mean displacement vector, respectively. Adaptation at the top level could change the resolution of the computational coarse grid (M_C, N_C) and the number of frames to be dropped (synchronous mode: N_d) or identically the time delay of the frame acquisition (asynchronous mode: t_d). Obviously, both adaptation types will directly affect the temporal rate of the processing (f).

Adaptation at level 2 (local for an arbitrary coarse grid node point $\sim$ "active fovea")

$$\Im_{2,ij}(dx_{ij}, dy_{ij}, \eta_{1,ij})_{|\vartheta_1} \Rightarrow \{(x_{ij}(\Im_{2,ij}), y_{ij}(\Im_{2,ij})); C_{ij}(\Im_{2,ij})\}. \tag{3}$$

This middle-level self-adjusting process is responsible for coarse grid deformation, i.e., repositioning the node points (x_{ij}, y_{ij}) where the fine grid analysis takes place and also for the complexity of the computational strategy used in topographic processing (C_{ij}). The repositioning is based on the value of the displacement measure $(\eta_{1,ij})$ with respect to the confidence threshold (ϑ_1).

Adaptation at level 3 (local for an arbitrary coarse grid node point $\sim$ "active fovea")

$$\Im_{3,ij}(\eta_{2,ij})_{|\vartheta_2} \Rightarrow \{s_{x,ij}(\Im_{3,ij}), s_{y,ij}(\Im_{3,ij})\}. \tag{4}$$

The bottom level adaptation process is responsible for spatial scale modification, i.e., changing the zoom of the window $(s_{x,ij}, s_{y,ij})$ of the fine grid analysis. The scale adaptation is based on the value of the contrast measure $(\eta_{2,ij})$ with respect to the confidence threshold (ϑ_2).

All the above three strategies could be applied independently or the neighboring hierarchical levels could affect each other through the "testing" functions, for example in a top-down scheme (see also Fig. 3)

$$\Im_3(\Im_2(\Im_1(\cdot), \cdot), \cdot). \tag{5}$$

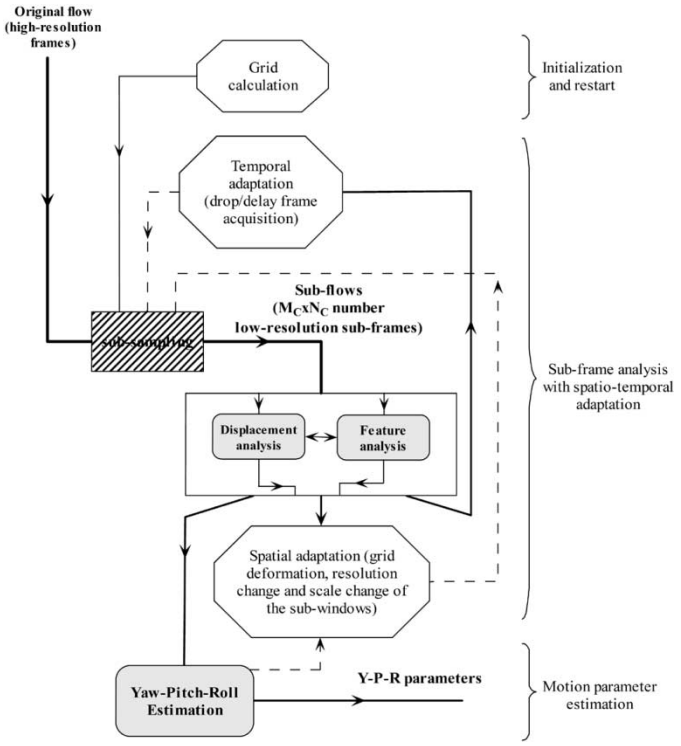

Fig. 5. Functional building blocks of the cellular multiadaptive spatio-temporal algorithmic framework for motion parameter estimation. Through subsampling of the original high-resolution flow (ideally this should be done at the sensor level) $M_C \times N_C$ number of low-resolution $(M_F \times N_F)$ subframes are received at the topographic processing level. Then, each of these subframes is processed in parallel (feature and displacement analysis) and the results (optical-flow estimates and the associated confidence measures) are forwarded to the motion parameter-estimation level. The output of the topographic processing is also evaluated in the algorithmic modules controlling the spatial and temporal adaptation of the cellular coarse grid of subframes. The internal error estimation at the motion parameter-estimation level might also affect the spatial adaptation properties (typically the resolution of the coarse grid, i.e., the number of optical-flow estimates sent to the Y–P–R estimation module).

The navigation parameter-estimation-related flow chart of the above-described adaptive algorithmic framework with its major functional building blocks is shown in Fig. 5, where the optical-flow estimates and the spatial adaptation depends on the internal error of motion parameter estimation (see the dashed lines). Using the notation introduced in this section, this can be described as follows:

$$\begin{aligned} &\Im_0(\cdot) = \mathrm{err}(\cdot) \\ &\Im_1(\Im_0(\cdot), \underset{\rightarrow}{dx}, \underset{\rightarrow}{dy}, \underset{\rightarrow}{\eta_1}, \underset{\rightarrow}{\eta_2})_{|\vartheta_1, \vartheta_2, \vartheta_{v\max}, \vartheta_{v\min}} \\ &\quad \Rightarrow \{(M_C(\Im_1), N_C(\Im_1)); N_d(\Im_1)\} \\ &\Im_2(\Im_1(\cdot)) \quad \Rightarrow \{C(\Im_2)\}. \end{aligned} \tag{6}$$

Observe that only global schemes have been used. This top-down control of the bottom-up topographic processing in adaptive motion parameter [yaw–pitch–roll (Y–P–R)] estimation will be specifically defined in Sections V and VI. The related measurement results will be shown and analyzed in Section VII.

Remarks:

- Modules implementing other higher level functions, such as region classification [36] and automatic attention control [37] are also placed at the motion parameter-estima-

tion level and are driven by more intensive feature analysis within a single [35] or multiple subframes. These modules could also affect the spatial adaptation properties by changing the central position of the coarse grid and/or modifying the attributes of the most active fovea (attention-selection control).

Below the pseudocode description of the motion parameter estimation is given which corresponds to the actual implementation. In the remaining part of the paper the name of these blocks will be referenced in time performance measurements and related discussions.

```
VisualNavigation::Run
{
PreProcFrame:      Frame Acquisition (with Interpolation)
CalibFrame:        Coarse Grid Generation and Temporal
Adaptation
SetMethod:         Set Topographic Processing Method
for k = 1:SubFrameNum
   {
   PreProcSubFrame:   Image Smoothing (and Noise Suppression)
   CalibSubFrame:     Histogram Analysis
                      Adaptive Threshold Selection
                      Feature Extraction (Patch, Edge, Skeleton)
   ProcessSubFrame:   Displacement (Optical Flow) Estimation
   SetSubFocus:       Coarse Grid and Scale Adaptation
   }
ProcessFrame:      Y–P–R Estimation
SetFocus:          Focus Window Repositioning
ProcessAllFrames:  Overall Statistics
}
```

IV. Spatio-Temporal Topographic Computation: Parallel Cellular Processing

The computation of "feature-based" optical-flow estimates could heavily rely on efficient nonlinear feature extraction and correspondence analysis on a CNN-type visual microprocessor. The methodology consists of three major steps:

1) parallel cellular nonlinear preprocessing and feature extraction;
2) displacement computation through binary feature matching;
3) finding the maximum of the matching results.

There are several important issues that should be balanced during an actual implementation.

Some simple nonlinear feature-extraction methods are shown in Fig. 6. In these calculations, adaptive thresholding is followed by edge detection or skeletonization-type operations (see [18] for standard iterated binary morphology-based CNN solutions and [14] for some advanced methods with a "single transient" computation). The topographic processing from (a) to (d) in Fig. 6 involves an increasing level of computational complexity.

It should be noted that the "feature-based" optical-flow estimation process does not assume an exact contour or structure computation. In the image flow typical features should be detected in a consistent way that could be matched to each other while balancing speed, precision and robustness. This is especially the case when a gray-scale image projection of the three-dimensional world is analyzed in a real-time process. Then the feature extractors cannot be exactly related to contour or structure detecting operators as in binary image processing. This example is visualized in Fig. 7 where parallel cellular nonlinear preprocessing and feature extraction examples are shown on a CNN-type microprocessor (ACE4k[11]). As illustrated, some of the nonlinear feature extractors capture the granularity,overall edginess, oriented edginess, skeleton structure or some other characteristic features of the scene. As

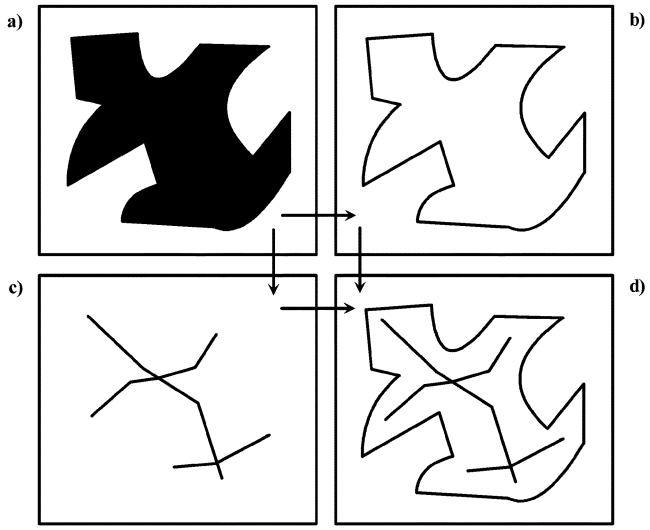

Fig. 6. Binary topographic results of some idealized feature extracting methods used in feature-based optical-flow computation. (a) Adaptive thresholding: Patch detection. (b) Edge detection: Contour from patches. (c) Skeleton detection: Structure from patches. (d) Structure and contour from patches. All processing is ideally supported by CNN topographic microprocessors including arbitrary shift and local logic operations, which are also used in various displacement estimation algorithms.

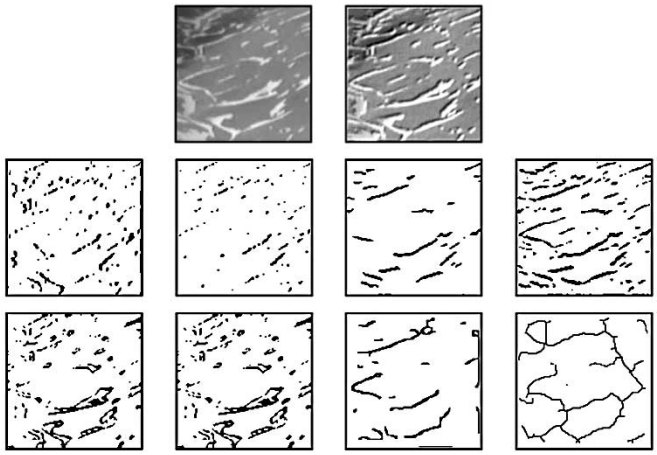

Fig. 7. Parallel cellular nonlinear preprocessing and feature extraction on a CNN-type microprocessor (ACE4k). The top row shows the original input (a sample from an aerial video flow: UAV_a) and the enhanced image after the cellular preprocessing. In the middle and bottom row, the output of several nonlinear feature extraction operations are shown. These operators capture the granularity, overall edginess, oriented edginess, skeleton structure, or some other characteristic features of the scene. Stable and robust features can be used in the preprocessing steps of a displacement analysis.

TABLE I

TIME PERFORMANCE OF SUBFRAME PREPROCESSING (SMOOTHING) AND CALIBRATION (ADAPTIVE THRESHOLD SELECTION FOLLOWED BY FEATURE CALCULATION) METHODS IN DIFFERENT DISPLACEMENT ESTIMATION STRATEGIES. MEASUREMENT RESULTS HAVE BEEN COMPLETED ON UAV_a TYPE FLOWS USING ACE4k VISUAL MICROPROCESSOR (EMBEDDED WITHIN THE ACE-BOX COMPUTATIONAL ENVIRONMENT). PROCESSING TIME IS GIVEN IN μs. (NUMBERED COLUMNS STAND FOR PIXEL WIDTH OF CONTOURS AND STRUCTURES IN CASE OF EDGE AND SKELETON EXTRACTION, RESPECTIVELY). OBSERVE THAT THERE IS ROUGHLY AN ORDER OF MAGNITUDE DIFFERENCE IN COMPUTATIONAL EFFICIENCY COMPARING EDGE EXTRACTION TO SKELETON EXTRACTION

Function	Operations	Time in micro seconds		
		1	2	3
PreProcSubFrame	smoothing	654	673	692
CalibSubFrame	histogram+threshold	256	-	-
	histogram+threshold+edge	332	336	339
	histogram+threshold+skeleton	2490	2496	2502
	histogram+threshold+edge+skeleton	2527	2532	2538

TABLE II

TIME PERFORMANCE OF SUBFRAME PROCESSING (FEATURE BASED BLOCK MATCHING) METHODS IN DIFFERENT DISPLACEMENT ESTIMATION STRATEGIES. MEASUREMENT RESULTS HAVE BEEN COMPLETED ON UAV_a TYPE FLOWS USING ACE4k VISUAL MICROPROCESSOR (EMBEDDED WITHIN THE ACE-BOX COMPUTATIONAL ENVIRONMENT). PROCESSING TIME IS GIVEN IN μsec (NUMBERED COLUMNS SHOW MAXIMUM NEIGHBORHOOD IN WHICH DISPLACEMENT ESTIMATION HAS BEEN PERFORMED). OBSERVE THAT BOTH SPIRAL AND DIAMOND SEARCH STRATEGIES ARE SIGNIFICANTLY QUICKER THAN THE FULL-SEARCH METHOD (IN CASE OF DIAMOND SEARCH, MINIMUM AND MAXIMUM VALUES ARE SHOWN DUE TO LARGE VARIATION FOR DIFFERENT FRAMES)

Function	Operations	Time in micro seconds			
		5x5	7x7	9x9	11x11
ProcessSubFrame	binary_full	8690	16583	27406	41391
	binary_spiral	2175	3886	6124	8888
	binary_diamond	992-1600	992-1600	992-2400	992-2400

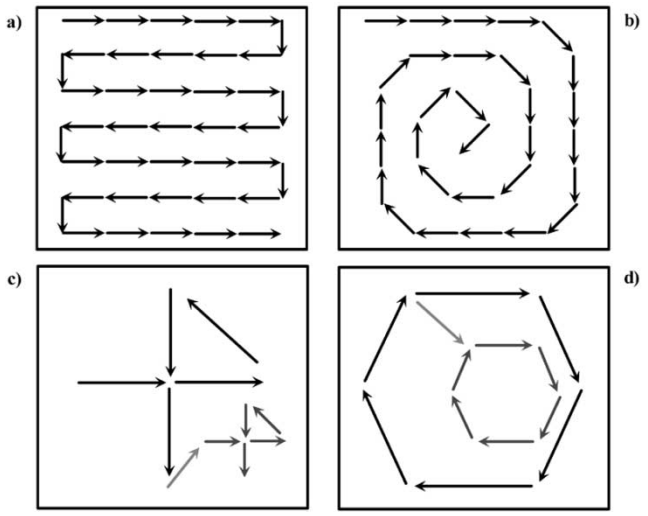

Fig. 8. Nonadaptive: (a) brute-force, and (b) spiral; and adaptive: (c) cross, and (d) hexagon-shaped diamond search routes in block matching strategies of feature-based optical-flow computation. While the adaptive solutions are computationally more efficient than the nonadaptive search mechanisms, their output quality might strongly depend on the selectivity and robustness of the preprocessing used.

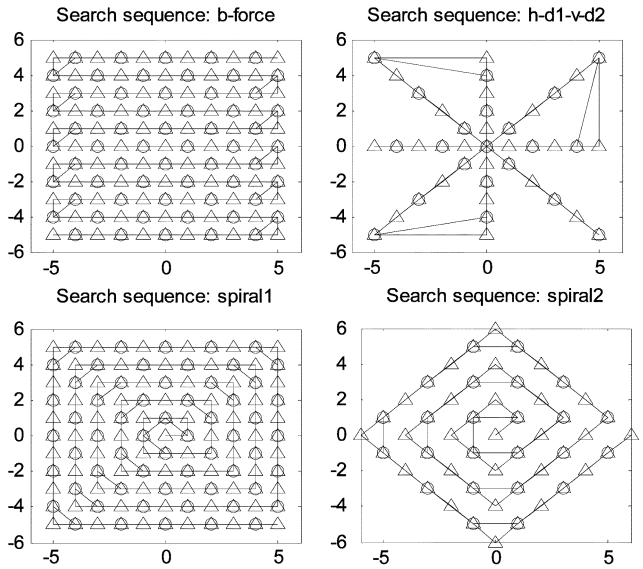

Fig. 9. Nonadaptive routes in block matching strategies of feature-based optical-flow computation with variable step size. The allowable subsampling strongly depends on the selectivity and robustness of feature preprocessing during the estimation process.

mentioned earlier, any consistent binary feature detector could be used as the base operator of the displacement estimation.

In our current experiments, we have thoroughly studied some fundamental feature extracting operators (used with histogram analysis-based adaptive thresholding) in order to reveal their computational complexity on a parallel cellular microprocessor (Table I).

Binary nonlinear feature extraction is followed by correspondence analysis (binary feature matching) comparing the newly detected features to the features extracted from the previous (reference) frame. The comparison is done along prescribed adaptive or nonadaptive routes as shown in Fig. 8 and the step size could also be variable as illustrated in Fig. 9. Though this process contains only binary image shift, spatial logic and pixel count operations (all these supported by the hardware) it is clear that its computational complexity could change dramatically depending on the strategy chosen. Brute-force methods will result in the best results; however with adaptive spiral or diamond search strategies the speed and precision requirements could be more efficiently balanced (Table II).

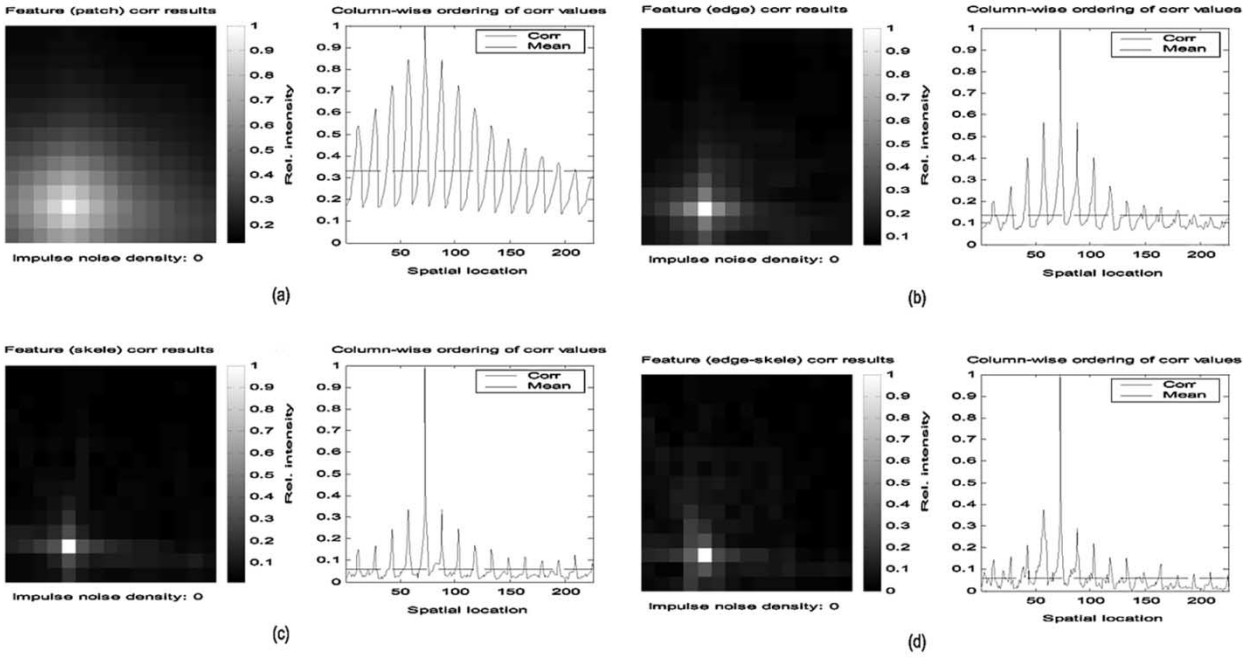

Fig. 10. Selectivity analysis of different image preprocessing methods in feature-based optical-flow computation (analyzing the local correlation map between two noiseless displaced input frames). The left column shows the intermediate correlation results of a displacement estimation using a full-search strategy within a 13×13 window with different preprocessing methods. The topographic processing strategies are as follows. (a) Simple thresholding: Patch matching. (b) Thresholding followed by edge detection: Contour matching. (c) Thresholding followed by skeletonization: Structure matching. (d) Combining (b) and (c): Contour and structure matching. The right column shows the column-wise ordering of the correlation values. The mean correlation value is also plotted (dashed line in the figures), which well characterizes the selectivity of the preprocessing strategies. Observe that the selectivity of both the contour and structural features significantly outperform the patch-based methods.

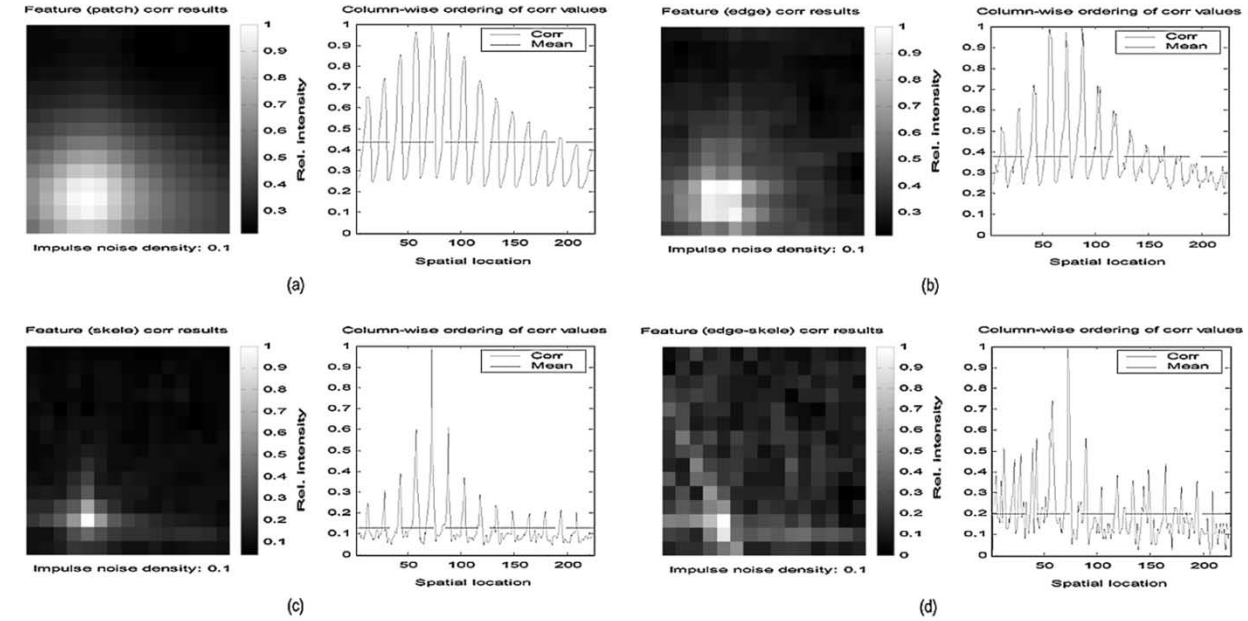

Fig. 11. Robustness analysis of different image preprocessing methods in feature-based optical-flow computation. The arrangement of the figures with respect to different topographic preprocessing strategies is identical to Fig. 10, but, in this case, the input frames have been corrupted by 10% impulse noise and the features have also been degraded by a single step binary opening. Observe that although the selectivity of the contour and structure matching strategies is still better compared to the patch-based methods, the robustness of the displacement estimation might be very poor if the correlation map is to be subsampled in the next computational step (e.g., in case of spiral or diamond search strategies). If the computational load, selectivity, and robustness of displacement estimation are all taken into the account, the contour-type (edge detection) preprocessing could be the best choice (see also Figs. 13–16 and Tables I–III, which also confirm the latter statement).

The last stage of feature-based displacement estimation is finding the maximum of feature matching results. As a straightforward solution N-point (typically 1- or 3-point) methods could be used to find the tradeoff between speed and robustness. It should be noted though that the time requirement of this maximum detection will always be negligible compared

TABLE III
DISPLACEMENT CONFIDENCE VALUES CALCULATED FOR DIFFERENT ESTIMATION STRATEGIES. IN THESE EXPERIMENTS VARIOUS FEATURE EXTRACTION STRATEGIES (ROWS, EACH TOPOGRAPHIC FEATURE PROCESSING IS COMBINED WITH A FULL SEARCH METHOD) ARE TESTED FOR INCREASING IMPULSE NOISE DEGRADATION (COLUMNS). STATISTICAL ANALYSIS SHOWS THAT THE COMBINED EDGE-SKELETON-TYPE FEATURE EXTRACTION (THAT IS HIGHLY SELECTIVE AT LOW NOISE LEVELS) IS ALSO A ROBUST TOPOGRAPHIC PROCESSING TECHNIQUE. OBSERVE THAT THE PURE SKELETON TYPE PREPROCESSING HAS VERY SIMILAR PROPERTIES AND THE EDGE TYPE PREPROCESSING IS ALSO WITHIN A NARROW RANGE OF CONFIDENCE

Preproc method \ Imp noise density	0.0%	4.0%	8.0%	12.0%	16.0%	20.0%	24.0%	28.0%	32.0%	36.0%	40.0%
Patch	**0.67**	0.65	0.64	0.63	0.61	**0.6**	0.58	0.56	0.55	0.53	**0.51**
Edge	**0.86**	0.83	0.80	0.77	0.74	**0.71**	0.68	0.67	0.62	0.62	**0.59**
Skele	**0.94**	0.91	0.87	0.84	0.82	**0.78**	0.75	0.71	0.67	0.61	**0.61**
Edge+Skele	**0.94**	0.91	0.89	0.86	0.82	**0.79**	0.76	0.74	0.68	0.68	**0.66**

to the preceding operations when displacement estimation is analyzed as a whole computational module.

We have defined two confidence levels characterizing the reliability of the optical-flow estimation. The primary one is called the *displacement measure* defined as

$$\eta_1 = 1 - \frac{x_{\text{aver}}}{x_{\max}} \tag{7}$$

where x_{aver} and $x_{\max}$ are the average and maximum values of the displacement search (feature matching) results, respectively. The secondary one is the *contrast measure* defined as

$$\eta_2 = \frac{I_{\max} - I_{\min}}{I_{\text{full}}} \tag{8}$$

where $I_{\max}, I_{\min}$, and I_{full} are the estimated maximum intensity, minimum intensity and full (dynamic range) intensity values, respectively.

In the course of the experiments, first, we tested the functional performance of some robust versions of the optical-flow estimation algorithm. Then, the selectivity (Fig. 10) and robustness (Fig. 11) of the nonlinear feature computation was thoroughly examined (see also Table III). In the latter case, we have found that structure-like features have the best selectivity (that directly affects precision) and robustness in case of a full search method, however in order to significantly improve the time performance contour-like features could be combined with spiral or diamond search strategies.

Based on the preceding analysis (see also the section on experimental results) the following (relatively robust) optical-flow estimation processes have been defined for detailed experiments with monotone increasing processing time requirement and associated performance precision measure (maximum search neighborhood should be larger than 10×10):

- C_1: Patch detection—spiral search—3-point maximum localization;
- C_2: Contour detection—diamond search—3-point maximum localization;
- C_3: Contour detection—spiral search—3-point maximum localization;
- C_4: Contour detection—brute-force search—3-point maximum localization;
- C_5: Structure detection—brute-force search—3-point maximum localization.

At the lowest end, patch detection is combined with adaptive spiral search since the nonselective nature of patch matching [Figs. 10(a), 11(a)] ensures that even in case of a noisy input, a rough sampling will still ensure an estimate close to the true values. Contour detection type preprocessing methods seem to be the best compromise in the tradeoff between computational complexity and selectivity [Figs. 10(b), 11(b)]. Combined with different search strategies from diamond to brute-force methods, the precision of the estimates will increase monotonically with increasing time needed to perform the computation. At the highest end, structure detection is combined with brute-force search giving the best expected results at the most complex computational level. In this latter case, other (rough) search strategies could not be used due to the very selective nature of the preprocessing method [Figs. 10(c), 11(c)]. In all cases, a 3-point maximum localization strategy is employed since it is much more robust for noisy inputs than a single-point search and there is no significant difference in its time performance compared to the preceding phases of the entire optical-flow estimation procedure.

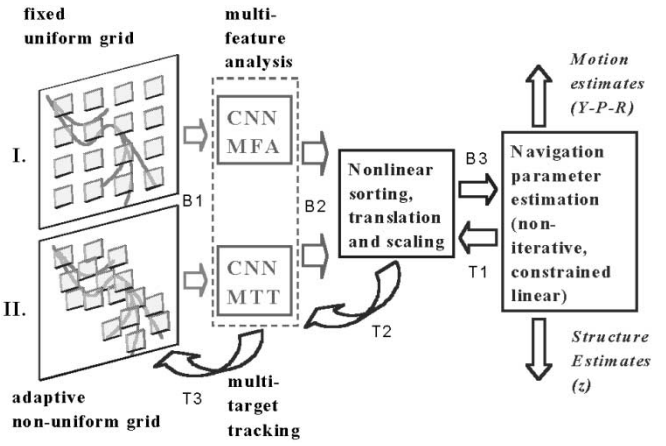

Fig. 12. Motion field and the associated flight control parameter estimation based on a uniform (fixed) and/or adaptive nonuniform (deformable) grid defined over the high-resolution video-flow. The figure displays two processing routes, one with a bottom-up processing (I.) and another with combined bottom-up and top-down processing (II.). In both cases, a selected number of subframes should be sent (B1) to the topographic level where the displacement of each window is estimated and forwarded (B2) to the displacement filter and, finally a subset of (possibly transformed) optical-flow estimates are reaching (B3) the navigation parameter-estimation block. The top-down scheme (T1–T3) is responsible for improving the precision on the bottom-up route (B1–B3). First, the selectivity of the nonlinear sorting could be enhanced (T1). Then, the spatial resolution of the coarse grid (the number of analyzed subframes) can be increased (T2), and finally the displacement computational method can be changed (T3) in order to improve the precision of the local displacement estimates.

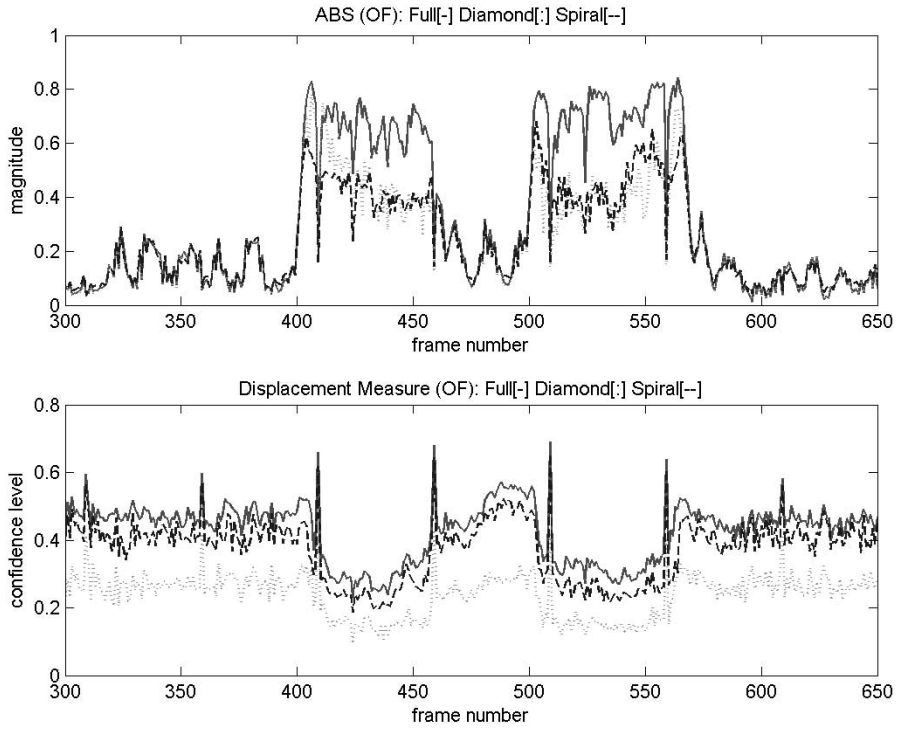

Fig. 13. Evaluation of edge-type topographic preprocessing technique combined with various search strategies for an input comprising of large textured areas (UAV_h). The figure shows the statistical results of optical-flow estimation for edge preprocessing with full (solid curve), diamond (dotted curve) and spiral (dashed curve) search. Note that in the example, full search can be considered as a reference. Top: Robust mean of displacement measures for different strategies. Bottom: Robust mean of displacement confidence measure for different strategies. Observe that spiral search provides a good compromise between quality and computing speed (see also Tables I–III).

V. Qualitative Multiadaptive Framework Control

In order to complete a meaningful comparison between the adaptive and nonadaptive frameworks, first, we have developed a simple qualitative multiadaptive framework control. Within this process, all three levels of adaptation could be used either independently or concurrently (in the letter case coupling the different computational levels in a top-down scheme).

A typical goal for which the framework could be optimized is to adaptively follow environmental changes (as the observed process) between $[v_{\min}, v_{\max}]$ average speed with typical values $1 \div 10$ pixel/frame. Within this interval, it is expected that feature-based optical-flow estimation will work quite well and the assumption on spatio-temporal continuity of the image flow is roughly valid. Then, the mutual coupling between different hierarchical levels could be typically established based on computational complexity. When there is more time available, then more complex methods could be run at the lower levels achieving higher precision and greater robustness. In the sequel, an example for such qualitative control framework is described in detail.

The adaptation at the temporal level (Level 1) changes the time delay t_d between two consecutive frames. Assuming a synchronous image acquisition where $\Delta t_{\min}$ is fixed, this procedure at the kth temporal sample can be described as follows (using the notation introduced in Section III, in case of the testing function formulation the dependence on k will be omitted):

$$t_d(k+1) = N_d(k+1)\Delta t_{\min}$$

$$N_d(k+1) = \begin{cases} N_d(k)+1, & \text{if } \Im_1(\cdot) = +1 \\ N_d(k)-1, & \text{if } \Im_1(\cdot) = -1 \\ N_d(k), & \text{otherwise} \end{cases} \tag{9.1}$$

$$\Im_1(\cdot) = \begin{cases} -1, & \text{if } v_{\text{med}} > v_{\max} \wedge t_+ > T_+ \\ +1, & \text{if } v_{\text{med}} < v_{\min} \wedge t_- > T_- \\ 0, & \text{otherwise} \end{cases} \tag{9.2}$$

$$v_{\text{med}} = \text{median}\{\forall v_{ij} \,|\, \eta_{1,ij} > \vartheta_1 \wedge \eta_{2,ij} > \vartheta_2\}$$

$$v_{ij} = |dx_{ij}| + |dy_{ij}| \tag{9.3}$$

where t_+ and t_- are the time durations for the associated events $v_{\text{med}} > v_{\max}$ and $v_{\text{med}} < v_{\min}$, respectively. These durations should reach certain temporal thresholds (T_+ and T_-) in order to change the sign of the testing function and consequently the temporal behavior of the algorithmic framework. In short, at this level, the robust statistics (the median of the most reliable estimates is calculated) of the absolute value of the displacement magnitudes is estimated and compared to the limit values given. When any of the limits is exceeded for some time $\Im_1(\cdot)$ signals the event type with a signed scalar value.

The adaptation at the spatial grid level (Level 2) changes the resolution (the number of fine resolution windows: $M_C \times N_C$)

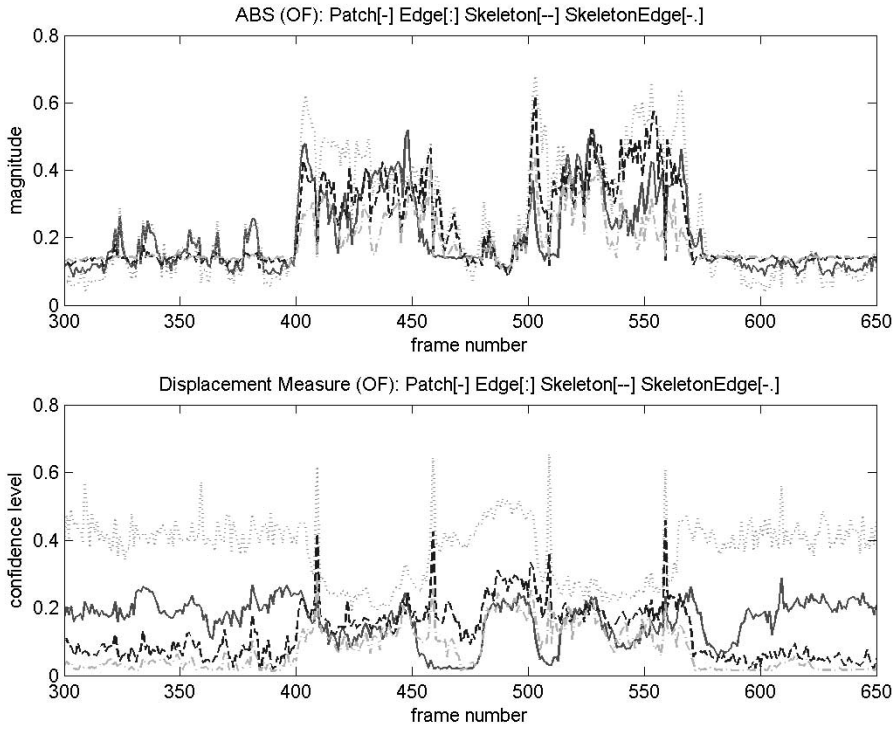

Fig. 14. Evaluation of different topographic preprocessing techniques combined with spiral search strategy for an input comprising of large textured areas (UAV_h). The figure shows the statistical results of optical-flow estimation for spiral search with patch (solid curve), edge (dotted curve), skeleton (dashed curve) and edge-skeleton (dashed-dotted curve) preprocessing. Note that in this example none of the measurements can be considered as a reference. Top: Robust mean of displacement measures for different strategies. Bottom: Robust mean of displacement confidence measures for different strategies. Observe that edge-type preprocessing clearly outperforms, in quality, all the other preprocessing methods (see also Tables I–III).

and/or the deformation (the exact placement of the fine resolution windows $[x_{ij}, y_{ij}]$) of the coarse grid.

The first (global) adaptation type is related to temporal self-adjustment (and could couple the adaptation at the top and middle levels) since changing the coarse grid resolution will change significantly the amount of computation required between two consecutive frames. This process is given as follows:

$$(M_c(k+1), N_c(k+1)) = \begin{cases} (M_c(k)+1, N_c(k)+1), & \text{if } \Im_1(\cdot) = +1 \\ (M_c(k)-1, N_c(k)-1), & \text{if } \Im_1(\cdot) = -1 \\ (M_c(k), N_c(k)), & \text{otherwise.} \end{cases} \quad (10)$$

The second (local) adaptation type is responsible for positioning the fine resolution windows and can be described as follows:

$$[x_{ij}(k+1), y_{ij}(k+1)] = \begin{cases} [x_{ij}(k) + \xi dx_{ij}, y_{ij}(k) + \xi dy_{ij}], & \text{if } \Im_{2,ij}(\cdot) = +1 \\ [x_{ij}(k), y_{ij}(k)], & \text{otherwise} \end{cases} \quad (11.1)$$

$$\Im_{2,ij}(\cdot) = \begin{cases} +1, & \text{if } \eta_{1,ij} \geq \vartheta_1 + \delta_1 \\ 0, & \text{otherwise.} \end{cases} \quad (11.2)$$

In this adaptation, the local testing function will detect the reliable displacement estimates (based on the displacement confidence measure) and the associated coarse grid nodes will be repositioned. Setting $\xi = 1$ tracking of the perceived motion is performed by the fine resolution window based on a constant velocity kinematic model. The deformation of the coarse grid through tracking is an attempt to keep the detectable features within the analysis window.

The adaptation at the spatial scale level (Level 3) modifies the scale ("digital zoom" of the fine resolution windows: $[s_{x,ij}, s_{y,ij}]$) and/or the computational procedure (C_{ij}).

The first (local) adaptation type is given as (only a few integer numbered scales are considered)

$$[s_{x,ij}(k+1), s_{y,ij}(k+1)] = \begin{cases} [s_{x,ij}(k)+1, s_{y,ij}(k)+1], & \text{if } \Im_{3,ij}(\cdot) = +1 \\ [s_{x,ij}(k)-1, s_{y,ij}(k)-1], & \text{if } \Im_{3,ij}(\cdot) = -1 \\ [s_{y,ij}(k), s_{y,ij}(k)], & \text{otherwise} \end{cases} \quad (12.1)$$

$$\Im_{3,ij}(\cdot) = \begin{cases} +1, & \text{if } \eta_{2,ij} \leq \vartheta_2 - \delta_2 \\ -1, & \text{if } \eta_{2,ij} \geq \vartheta_2 + \delta_2 \\ 0, & \text{otherwise.} \end{cases} \quad (12.2)$$

Similar to the previous case, in this adaptation, the local testing function will detect the reliable displacement estimates (based on the contrast confidence measure) and the spatial scale of the analysis at the associated coarse grid nodes will be changed. Low contrast forces a "zoom out" while high

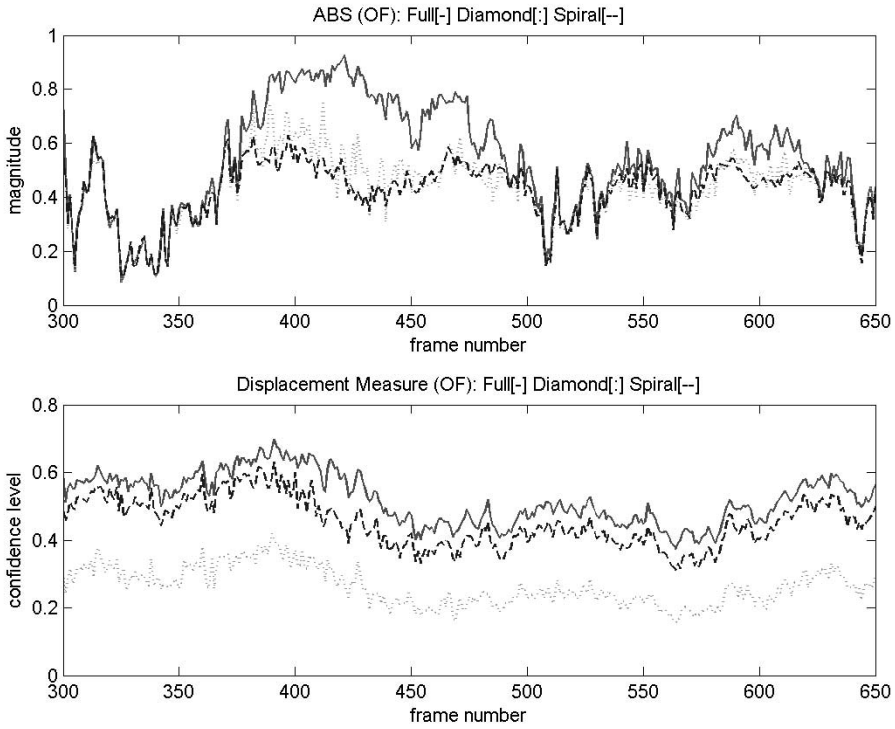

Fig. 15. Evaluation of edge-type topographic preprocessing technique combined with various search strategies for an input comprising of a highly structured indoor scene (UGV_rp). The figure shows the statistical results of optical-flow estimation for edge preprocessing with full (solid curve), diamond (dotted curve) and spiral (dashed curve) search. Note that in the example, full search can be considered as a reference. Top: Robust mean of displacement measures for different strategies. Bottom: Robust mean of displacement confidence measures for different strategies. Observe that spiral search provides a good compromise between quality and computing speed (see also Tables I–III).

contrast a "zoom in" type change in an attempt to increase the contrast/spatial details within the analysis window.

The second (global) adaptation type is related to temporal self-adjustment (and couples the adaptation at middle and bottom levels) since changing the computational method in all fine resolution windows will change significantly the amount of computation required between two consecutive frames. This process is given as follows:

$$C(k+1) = \begin{cases} C(k)+1, & \text{if } \Im_2(\cdot) = +1 \\ C(k)-1, & \text{if } \Im_2(\cdot) = -1 \\ C(k), & \text{otherwise} \end{cases} \tag{13.1}$$

$$\Im_2(\cdot) = \Im_1(\cdot). \tag{13.2}$$

Observe that with this latter scheme, we have somewhat simplified the original framework in Section III where a locally adjustable computing strategy has been defined. On the other hand, this allows us to devise a simple but efficient top-down scheme for further experiments (navigation parameter estimation).

As has been already noted, the interaction of the neighboring adaptation levels could be accomplished through changing the computational complexity of the optical-flow estimation procedures (Level 3), changing the resolution of the coarse grid (Level 2), and changing the time delay between two consecutive frames (Level 1). Accordingly, a simple "state machine" can be designed controlling the interactions in top-down direction first changing the timing of the frame acquisition at Level 1, then modifying the coarse grid resolution at Level 2, and finally, increasing/decreasing the estimation complexity at Level 3. During this simple procedure, only the first testing function output $(\Im_1(\cdot))$ is used and a predefined precedence order between different levels. As a simple rule, a particular level can trigger the next one when the modifications will reach the predefined limit values (e.g., L1: reaches the minimum frame rate $\rightarrow$ L2: reaches the maximum spatial coarse grid resolution $\rightarrow$ L3: increases the computational complexity within all subframes). See a related further discussion in the next section.

Remarks:

- During the described adaptation, there is no mechanism ensuring that the system will return to its initial state. This can be re-enforced by periodic re-initialization to the default values.
- The example given defines a simple qualitative control framework where the adaptation is decoupled at all levels or follows a top-down scheme. More sophisticated versions could be easily developed by using concurrent adaptation strategies at certain (or all) levels. In this case the main difficulty lies in the reliable assessment of the stability and consistency of the overall system performance.
- Using the deformable grid approach in the adaptive framework it can be noticed that the fine resolution windows (defined at the coarse grid nodes) perform a multitarget

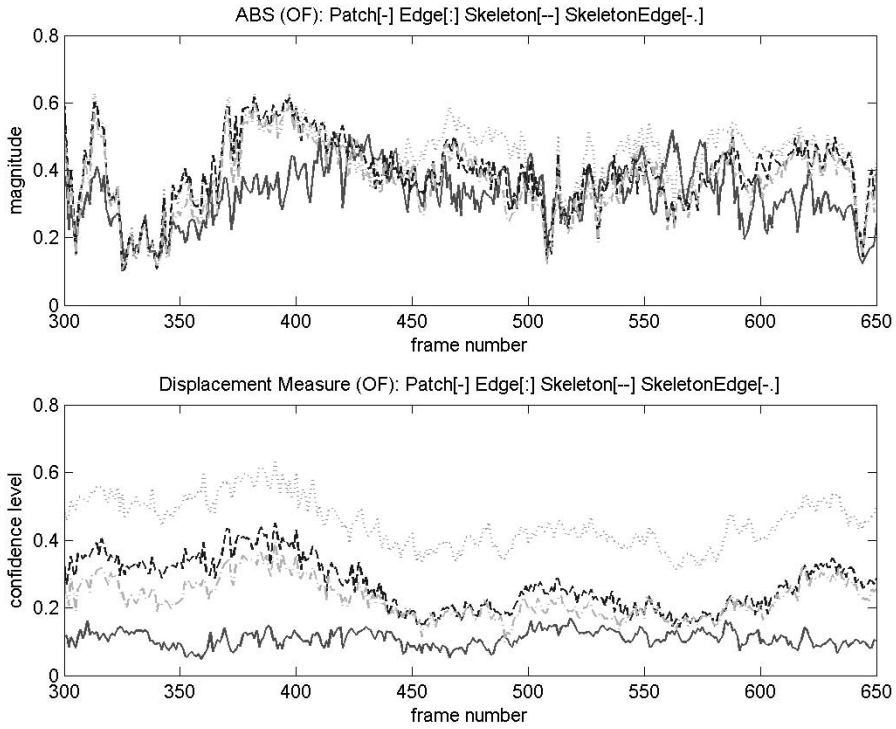

Fig. 16. Evaluation of different topographic preprocessing techniques combined with spiral search strategy for an input comprising of a highly structured indoor scene (UGV_rp). The figure shows the statistical results for spiral search with patch (solid curve), edge (dotted curve), skeleton (dashed curve) and edge-skeleton (dashed-dotted curve) preprocessing. Note that in this example, none of the measurements can be considered as a reference. Top: Robust mean of displacement measures for different strategies. Bottom: Robust mean of displacement confidence measures for different strategies. Observe that edge-type preprocessing clearly outperforms in quality all the other preprocessing methods (see also Tables I–III).

tracking in a classical sense with simple gating, local nearest neighbor data assignment and constant velocity model-based state prediction [22].

VI. Navigation Parameter Estimation

The main processing blocks of an integrated motion field and navigation parameter-estimation procedure is shown in Fig. 12. In its simplified form (see flow description I. in Fig. 12), this could be a multifeature analysis over a fixed uniform grid (no dynamic descriptors) or the enhanced version of the multitarget tracking scheme over an adaptive nonuniform grid (see flow description II in Fig. 12). The output of both types of processing should undergo a robust nonlinear sorting, translation and scaling before it enters the stage of navigation parameter estimation.

The flight control parameter-estimation module requires the motion field estimates (the optical flow, [23]) for each frame as an input and it calculates the unit translation direction along with the rotation parameter estimates (Y–P–R) and possibly the (relative) structure estimates. The main steps of an efficient navigation parameter-estimation procedure are described in [24]–[27] (the "8-point algorithm"). This constrained linear approach assumes a series of singular value/eigenvalue decompositions and general vector-matrix operations that can be optimized for high-end DSPs (iterative nonlinear strategies at this level could not be the basis for a real-time solution). However, the reliability of the entire solution depends strongly on the precision of the two-dimensional (2-D) motion field estimation performed at the topographic level. In order to justify this statement first we will formulate the essence of the navigation parameter-estimation problem (Y–P–R estimation) by embedding it into the multiadaptive algorithmic framework developed in the previous sections.

Let us denote the Y–P–R parameters by $[\alpha, \beta, \gamma]$, respectively. Then, the estimation procedure can be described as follows:

$$[\alpha, \beta, \gamma] = \mathrm{YPR}(d\hat{\underset{\rightarrow}{x}}, d\hat{\underset{\rightarrow}{y}})$$
$$[d\hat{\underset{\rightarrow}{x}}, d\hat{\underset{\rightarrow}{y}}] = \Psi(d\underset{\rightarrow}{x}, d\underset{\rightarrow}{y}, \underset{\rightarrow}{\eta_1}, \underset{\rightarrow}{\eta_2}) \tag{14.1}$$

where $\mathrm{YPR}(\cdot)$ stands for the rotation parameter estimation and $\Psi(\cdot)$ represents the nonlinear sorting, translation and scaling of the optical-flow estimates.

Within the multiadaptive algorithmic framework the internal error estimation $\Phi(\cdot)$ of the "8-point algorithm" (based on first order perturbation [24]) can be taken as a top level "testing function" initiating the process of the top-down adaptation mechanism

$$\varepsilon = \Im_0(\cdot) = \Phi(d\hat{\underset{\rightarrow}{x}}, d\hat{\underset{\rightarrow}{y}}) \tag{15.1}$$

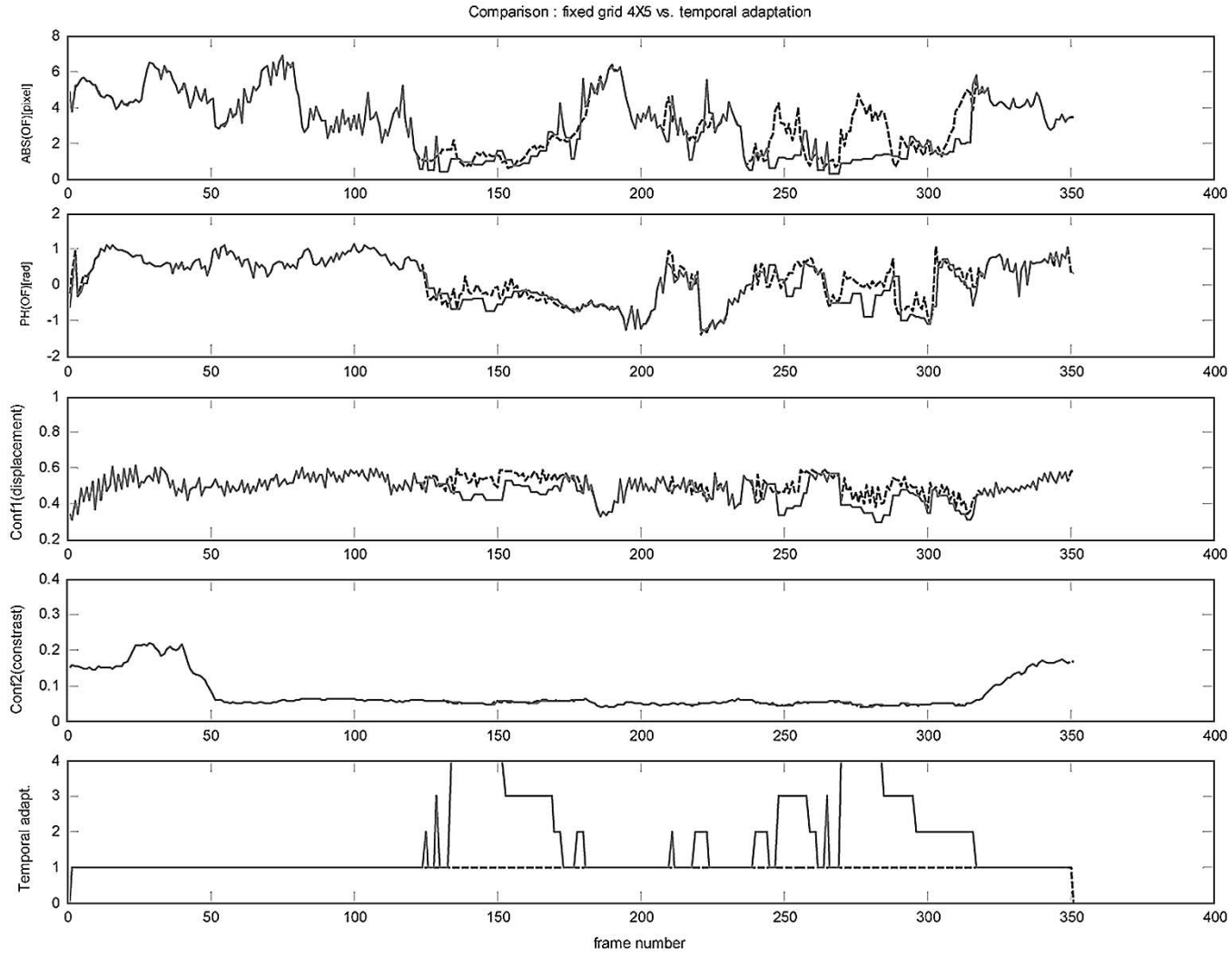

Fig. 17. Temporal adaptation (solid curves) in cellular multiadaptive algorithmic framework processing a UAV video input (UAV_a). The flow has been evaluated through a fixed 4×5 spatial coarse grid to create a reference (blue dashed curves). The figures from the top show the robust statistical mean of (i) magnitude of local optical-flow estimates, (ii) the phase of the local optical-flow estimates, (iii) the displacement confidence measures, (iv) the contrast confidence measures, and (v) the number of frames dropped during the temporal adaptation. Observe that for most cases, the control is adequate while the processing load is reduced in proportion with the sampling speed during the process (see also Table IV).

$$\Im_1(\cdot) = \begin{cases} -1, & \text{if } \varepsilon > \varepsilon_{\max} \\ +1, & \text{if } \varepsilon < \varepsilon_{\min} \\ 0, & \text{otherwise.} \end{cases} \tag{15.2}$$

When $\varepsilon > \varepsilon_{\max}$ or $\varepsilon < \varepsilon_{\min}$ then, the spatial grid adaptation is triggered at Level 2 [and *the modification of the spatial coarse grid resolution* is performed as defined in (10)]

$$\Im_1(\underset{\rightarrow}{dx}, \underset{\rightarrow}{dy}, \underset{\rightarrow}{\eta_1}, \underset{\rightarrow}{\eta_2})_{|\vartheta_1,\vartheta_2,\vartheta_{v\max},\vartheta_{v\min}} \Rightarrow \{(M_C(\Im_1), N_C(\Im_1))\}. \tag{16}$$

When $\varepsilon > \varepsilon_{\max}$ or $\varepsilon < \varepsilon_{\min}$ and, in addition, the coarse grid resolution has reached the upper or lower limit values, respectively, then, the spatial scale adaptation is triggered at Level 3 [and *the modification of the computational complexity* is performed as defined in (13)]

$$\Im_2(\cdot) = \begin{cases} -1, & \text{if } \Im_1(\cdot) = -1 \wedge M_c N_c = M_{c\max} N_{c\max} \\ +1, & \text{if } \Im_1(\cdot) = +1 \wedge M_c N_c = M_{c\max} N_{c\max} \\ 0, & \text{otherwise} \end{cases} \Rightarrow \{C(\Im_2)\}. \tag{17}$$

The above adaptive (combined bottom-up and top-down) method has been compared to the nonadaptive (simple bottom-up) calculations and has been found that it can significantly improve the precision of the Y–P–R estimation (see the related measurements in the Section VII).

VII. Experimental Results

While we have started the measurements of both COMPACT CVM prototypes (Bi-i V1 and Bi-i V2) with the ACE16k [13] CNN-UM chip, the architecture and the algorithmic

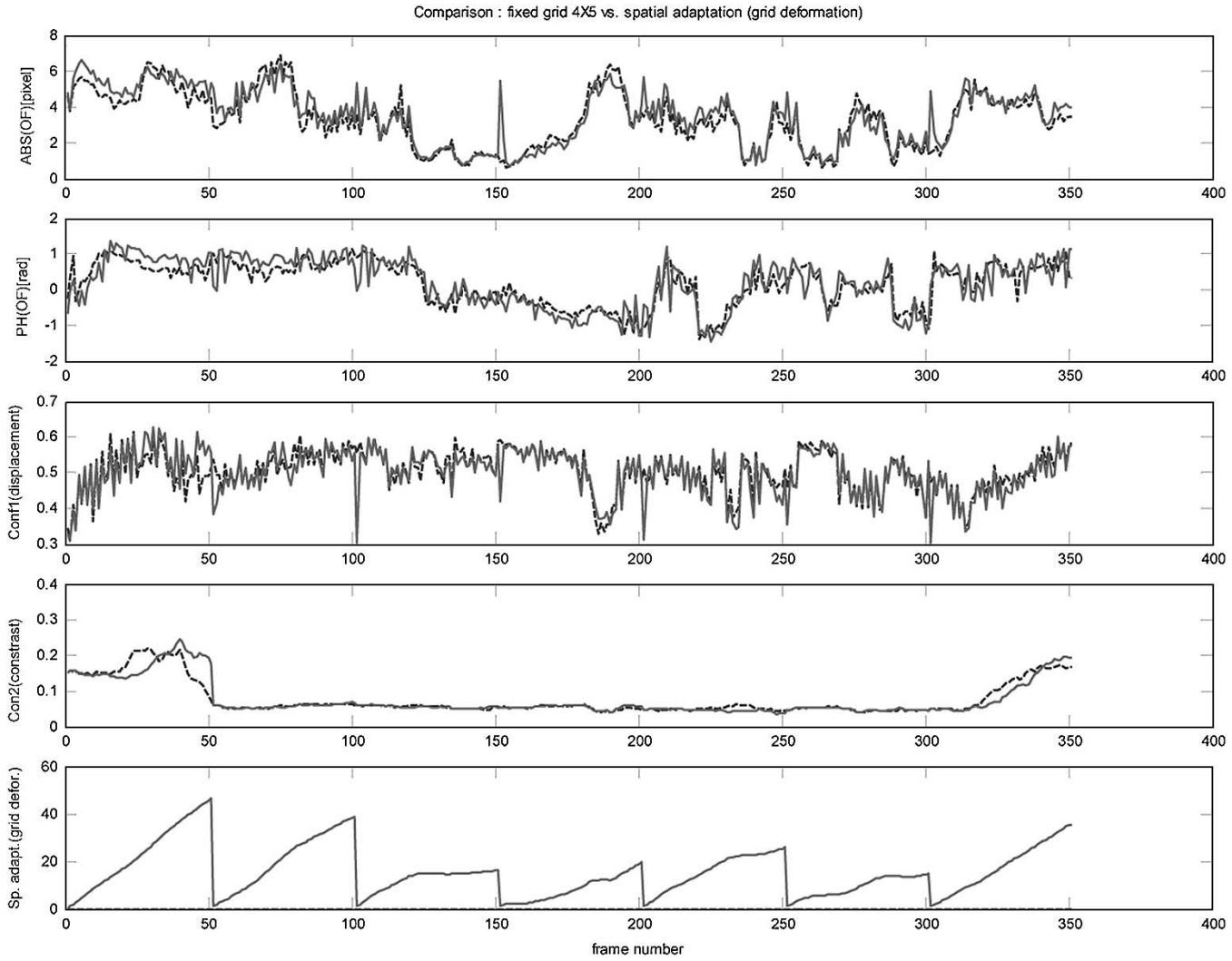

Fig. 18. Spatial adaptation (grid deformation— solid curves) in cellular multiadaptive algorithmic framework processing a UAV video input (UAV_a). The flow has been evaluated through a fixed 4×5 spatial coarse grid to create a reference (dashed curves). The figures from the top show the robust statistical mean of (i) the magnitude of local optical-flow estimates, (ii) the phase of the local optical-flow estimates, (iii) the displacement confidence measures, (iv) the contrast confidence measures, and (v) the deformation measured as the mean distance of the coarse grid nodes from their initial location (the zero values correspond to the periodic re-initializations of the coarse grid nodes). Observe that for most cases, the control is adequate while the grid deformation allows a parallel multitarget (in this case salient feature) tracking behavior over a large number of frames (see also Table IV).

framework described in this paper have been emulated within the ACE-BOX environment [19] hosting the ACE4k [11] chip the previous generation CNN-UM-type microprocessor. The ACE-BOX system is also capable of processing at or above video frame-rate with external video input acquired through a frame grabber device or loaded from the hard disk.

A. Video Input of the Experiments

In the sequel, the descriptions of the input flow types are given (along with a type code for reference throughout the paper) that were used in the experiments testing the cellular multiadaptive framework and the various algorithmic solutions.

1) UAV (airplane) acquired visual flows (different terrain videos)
   - partial ground-truth references are available for Y–P–R estimation [microelectromechanical systems (MEMS) gyro]
   - type code: **UAV_a**.
2) UAV (helicopter) acquired visual flows (different terrain videos)
   - partial ground-truth references are available for Y–P–R estimation (MEMS gyro)
   - type code: **UAV_h**.
3) Synthetic (flight simulator) visual flows acquired by a UAV flying above different terrains
   - exact ground-truth reference is available for Y–P–R estimation (generated)
   - type code: **UAV_fs**.
4) Synthetic (computer generated) OF and Y–P–R data
   - exact ground-truth reference is available for Y–P–R estimation (generated)
   - type code: **UAV_cg**.
5) UGV (robot platform) acquired visual flows (videos showing structured indoor environment)
   - ground-truth reference is not available for Y–P–R estimation
   - type code: **UGV_rp**.

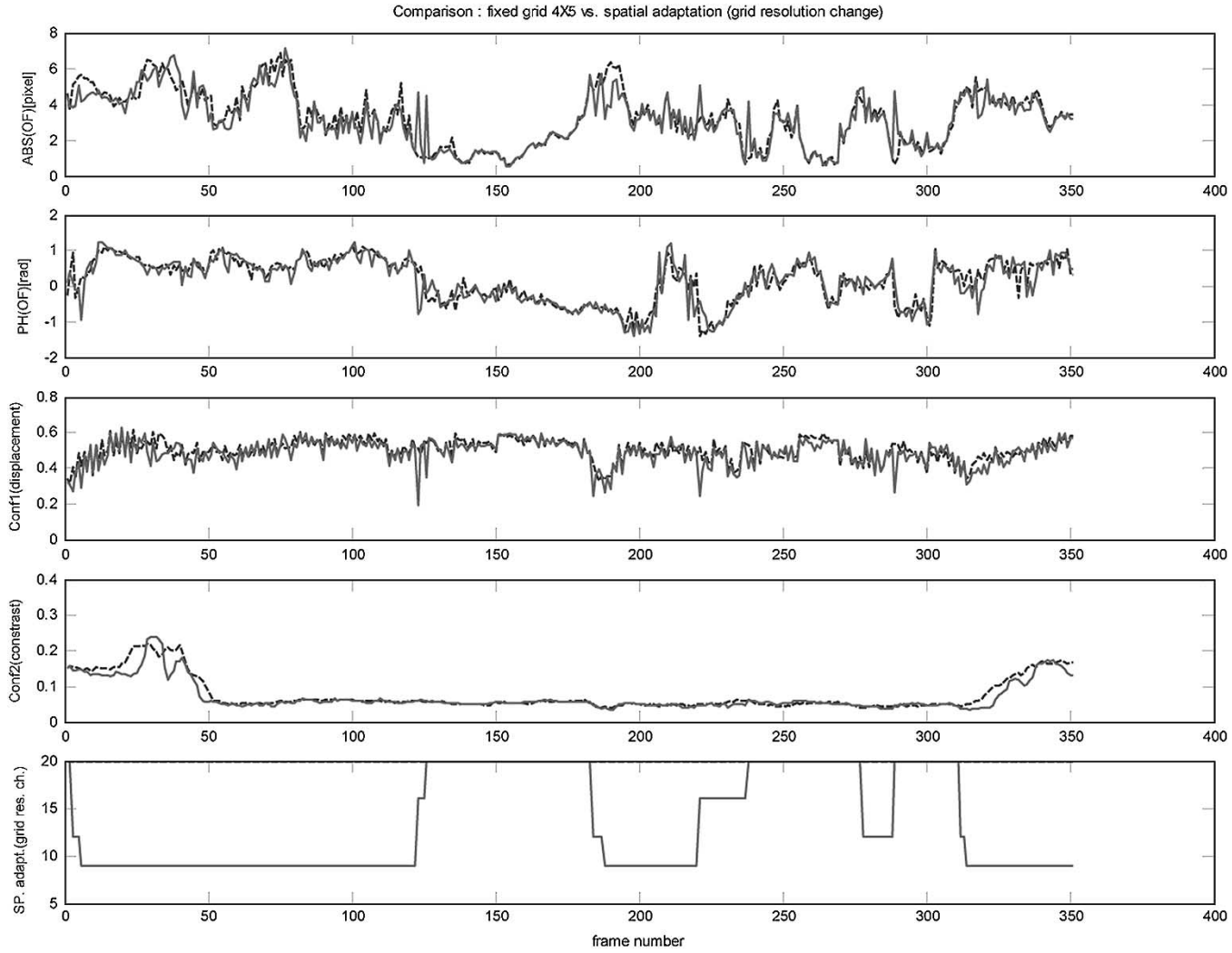

Fig. 19. Spatial adaptation (grid resolution change—solid curves) in cellular multiadaptive algorithmic framework processing a UAV video input (UAV_a). The flow has been evaluated through a fixed 4×5 spatial coarse grid to create a reference (dashed curves). The figures from the top show the robust statistical mean of (i) the magnitude of local optical-flow estimates, (ii) the phase of the local optical-flow estimates, (iii) the displacement confidence measures, (iv) the contrast confidence measures, and (v) the number of active subframes evaluated at the coarse grid nodes. Observe that for most cases the control is adequate while the processing load is reduced in proportion with the number of subframes evaluated during the process (see also Table IV).

B. Displacement Analysis—Optical-Flow Estimation

Displacement analysis related measurements are shown in Figs. 13–16, examining different preprocessing techniques with various search strategies. The analysis suggests that if both computation complexity and performance quality are also taken into consideration then spiral search with edge-type preprocessing is a good compromise in optical-flow estimation for both textured (outdoor natural scene) and structured (indoor environment with artificial objects) input flows.

C. Cellular Multiadaptive Framework—Qualitative Control

The qualitative control related measurements illustrate the effect of temporal adaptation (Fig. 17), grid resolution change (Fig. 18), and grid deformation (Fig. 19) when these strategies are used individually (see also Table IV). In summary, it can be stated that by applying various simple strategies, a significant computing power can be spared (and used for other calculations) while the tracking quality of some robust properties remains adequate.

D. Navigation Parameter Estimation Within the Adaptive and Nonadaptive Frameworks

Samples from the extensive experimental results are shown in Figs. 20 and 21 for navigation parameter estimation, comparing the algorithm calculated values to ground truth references. In both cases, a fixed 5×5 nonadaptive computational coarse grid was used as the reference framework and compared to an adaptive variable grid computing with 16×16 maximum coarse grid resolution (see the description of the adaptive algorithm in the previous section). As can be seen in all examples, the adaptive scheme significantly improved the noise reduction capability and reduced the cumulative rotation error of the parameter estimates. These experiments also show how a properly designed top-down processing can efficiently control (or

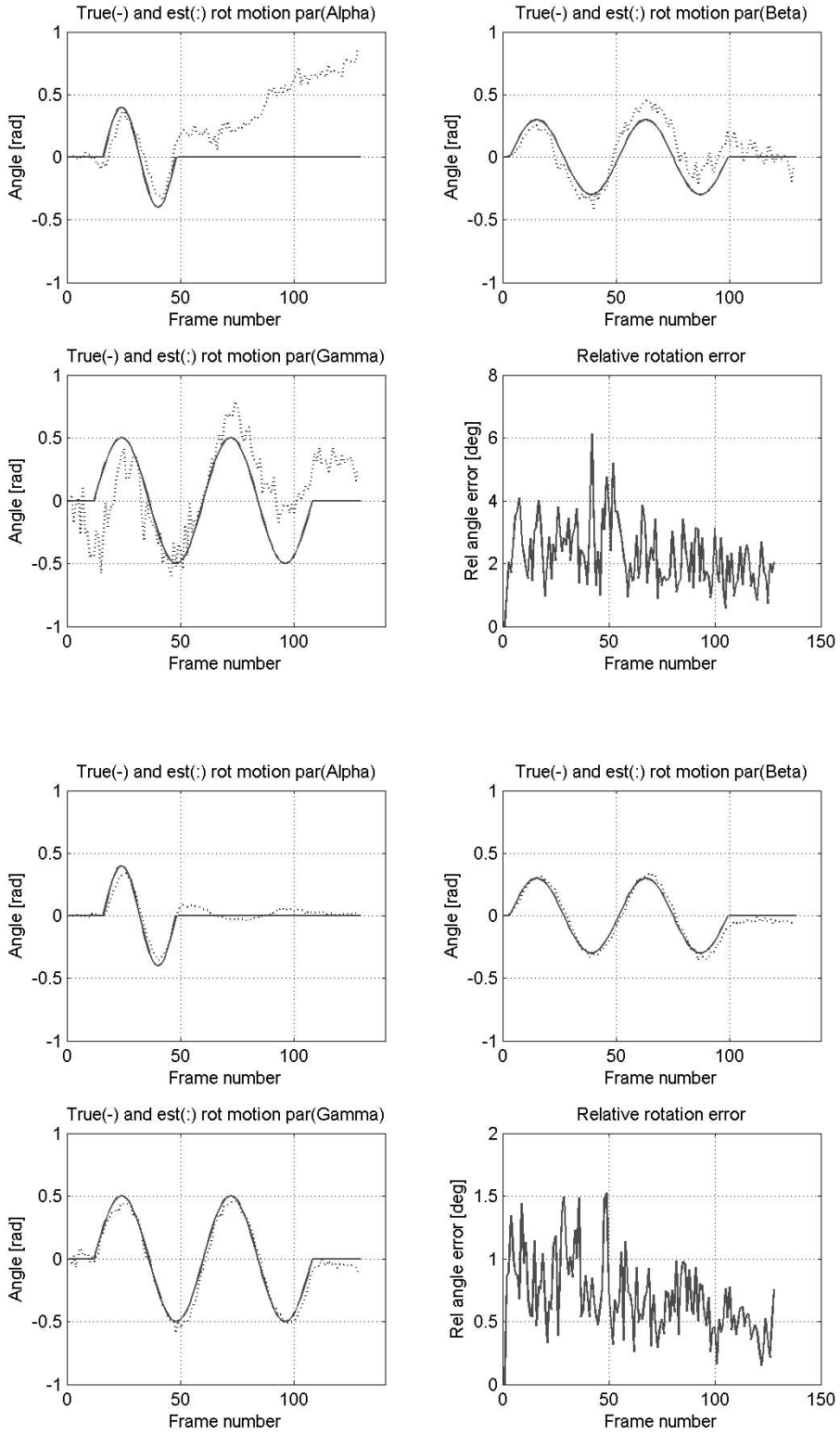

Fig. 20. Y–P–R estimation results (dashed curves) compared to ground-truth (solid curves) references (input flow type: UAG_cg). (a) Four figures show the results derived from a nonadaptive 5×5 computational coarse grid (the mean cumulative rotation error is 2.24°, see the bottom right figure). (b) Four figures display the output of the adaptive algorithm when the spatial coarse grid resolution is allowed to vary up to 16×16 (the mean cumulative rotation error is 0.68° see the bottom right figure). The adaptive algorithm adjusts the coarse grid resolution based on the results of internal error estimation. Observe that in this latter case, Y–P–R angles are recovered with great accuracy.

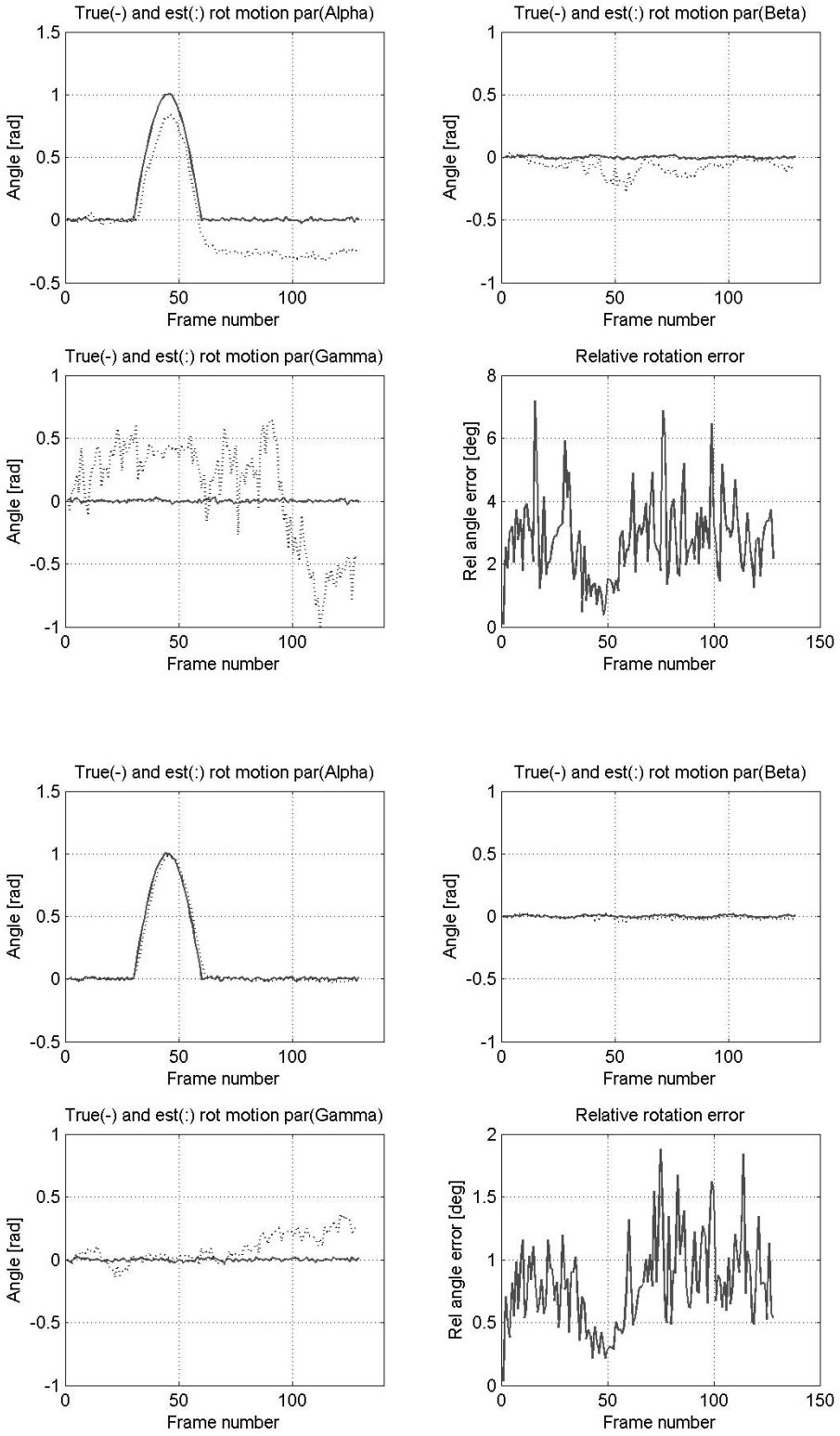

Fig. 21. Y–P–R estimation results (dashed curves) compared to ground-truth (solid curves) references (input flow type: UAG_cg). (a) Four figures show the results derived from a nonadaptive 5×5 computational coarse grid (the mean cumulative rotation error is 2.73°, see the bottom right figure). (b) Four figures display the output of the adaptive algorithm when the spatial coarse grid resolution is allowed to vary up to 16×16 (the mean cumulative rotation error is 0.79°, see the bottom right figure). The adaptive algorithm adjusts the coarse grid resolution based on the results of internal error estimation. Observe that in this latter case, Y–P–R angles are recovered with great accuracy.

TABLE IV
QUANTITATIVE EVALUATION OF THE CELLULAR MULTIADAPTIVE ALGORITHMIC FRAMEWORK FOR SOME SELECTED STRATEGIES (BASED ON QUALITATIVE CONTROL SCHEMES DESCRIBED IN THE PAPER). TEMPORAL ADAPTATION, SPATIAL-GRID DEFORMATION, AND SPATIAL-GRID RESOLUTION CHANGE HAVE BEEN EVALUATED WITHIN THE ACE-BOX COMPUTATIONAL ENVIRONMENT FOR DIFFERENT INPUT FLOW TYPES COMPARED TO RESULTS OF NONADAPTIVE COMPUTING OVER FIXED 4×5 COARSE GRID. TRACKING ERROR ANALYSIS HAS BEEN COMPLETED FOR MAGNITUDE AND PHASE TRACKING. AND FOR DISPLACEMENT AND CONTRAST CONFIDENCE MEASURES. FIRST FOUR COLUMNS SHOW RELATIVE ERROR VALUES CALCULATED AS $\mathrm{mean}(|\xi - \xi_{\mathrm{ref}}|/(\max(\xi_{\mathrm{ref}}) - \min(\xi_{\mathrm{ref}}))$, i.e., MEAN ABSOLUTE DIFFERENCE NORMALIZED TO DYNAMIC RANGE OF REFERENCE SIGNAL. IN THIRD AND FOURTH COLUMNS, SIGNED DIFFERENCE HAS BEEN USED FOR CONFIDENCE MEASURES (IN THESE CASES, NEGATIVE SIGN REPRESENTS RELATIVE IMPROVEMENT!). LAST COLUMN IN ALL TABLES SHOWS THE RELATIVE COMPUTATIONAL LOAD OF THE ANALYZED METHOD COMPARED TO REFERENCE MEASUREMENT

Adaptation method \ Analysis	Magnitude tracking	Phase tracking	Displ. measure	Contrast measure	Computing load
Input flow type: UAV_a					
Temporal adaptation	0.069	0.052	0.025	0.000	**0.761**
Spatial grid deformation	0.073	0.108	0.004	0.001	**1.000**
Spatial grid resolution change	0.065	0.072	0.016	0.006	**0.671**
Input flow type: UAV_h					
Temporal adaptation	0.056	0.169	0.004	0.000	**0.301**
Spatial grid deformation	0.029	0.063	0.000	-0.002	**1.000**
Spatial grid resolution change	0.017	0.061	-0.005	0.003	**0.943**
Input flow type: UAV_rp					
Temporal adaptation	0.066	0.105	0.031	0.000	**0.620**
Spatial grid deformation	0.097	0.087	-0.006	-0.016	**1.000**
Spatial grid resolution change	0.026	0.026	0.008	0.002	**0.832**

adjust) the bottom-up topographic calculations and perform an optimization for the top-level functions of the system.

Remark: It should be noted that commercially available MEMS gyros are sufficiently advanced that they will provide the Y–P–R measurements with greater accuracy than an optic-flow-based method. In these experiments effectively existing gyros could be used to provide a "ground truth" to demonstrate the accuracy of the system that will probably be of greater use for measuring linear translation than rotation. Performing investigations toward visual odometry could be one of the future main directions in further development of our system.

VIII. BIOLOGICAL MOTIVATIONS AND ONGOING WORKS

There is a strong biological motivation behind building a multichannel multiadaptive algorithmic framework for visual search and navigation. Since the initial development of the current architecture [33] and algorithmic framework [34], we have also been working on several bio-inspired aspects of UAV applications, these include:

1) terrain feature extraction based on multichannel cellular topographic algorithmic processing [35];
2) terrain feature classification employing artificial neural networks [36];
3) single fovea feature and/or classifier driven attention-selection mechanisms [37].

When building the computing blocks of the hierarchical algorithmic framework, several key processing strategies learned from retina modeling (e.g., [5]–[8]) and biological vision related experiments (e.g., [20]) have been considered. In particular, we are implementing and testing the following biologically motivated functions within the framework of our current system (the associated image processing and/or system design arguments are given in italic):

— spatial, temporal and spatio-temporal decomposition of the input flow: *an efficient geometric distortion analysis requires a sparse signal representation*;
— signal flow normalization: *dynamic range optimization*;
— parallel ON-OFF channel processing: *dc-component compensation*;
— narrow and wide-field wave-type interaction: *efficient binary patch shaping with noise suppression*
— "vertical" interaction of the decomposed channels: *forming a unique detection output through optimized "cross-talk" of the individual channels*;
— attention and selection mechanisms: *efficient content and context dependent processing*;
— saccade detection mechanisms: *proper handling of large shifts in the field of view*

IX. CONCLUSION

We have described a CNN technology-based vision system and computing platform for UAVs. The architecture discussed supports a multirate, multigrid and multiscale computing making the solutions scene adaptive and optimized for the observed processes. As an example, Y–P–R estimation was considered relying on optical-flow computation over a variable resolution coarse grid. It has been argued that such framework is also needed in other types of applications including terrain identification/recognition, multiple target tracking etc., practically in all those problems where a spatio-temporal consistency and continuity of the input image flow (and the observed processes) are crucial for properly working higher level functions.

Future experiments will be performed testing the combined adaptive nature of the experimental algorithmic framework and architecture. We will specifically focus on the following areas and research topics: attentive learning in terrain recognition, feature classifier driven attention-selection mechanisms, and visual odometry based on topographic displacement analysis.

ACKNOWLEDGMENT

The authors wish to thank the entire development team at AnaLogic-Computers Ltd. for implementing the algorithmic framework and the various chip level cellular topographic solutions within the ACE-BOX and Bi-i computational infrastructures (Aladdin Pro 3.x software system [19]). Special thanks are due to G. Erdõsi for completing numerous measurements and the related statistical analysis.

REFERENCES

[1] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1272–1290, Oct. 1988.
[2] L. O. Chua and T. Roska, "The CNN Paradigm," *IEEE Trans. Circuits Syst. II*, vol. 40, pp. 147–156, Mar. 1993.
[3] L. O. Chua, "CNN: A vision of complexity," *Int. J. Bifurcation Chaos*, vol. 7, no. 10, pp. 2219–2425, 1997.
[4] T. Roska and L. O. Chua, "The CNN universal machine," *IEEE Trans. Circuits Syst. II*, vol. 40, pp. 163–173, Mar. 1993.
[5] F. S. Werblin, T. Roska, and L. O. Chua, "The analogic CNN universal machine as a bionic eye," *Int. J. Circuit Theory Applicat.*, vol. 23, pp. 541–569, 1995.
[6] B. Roska and F. S. Werblin, "Vertical interactions across ten parallel stacked representations in mammalian retina," *Nature*, vol. 410, pp. 583–587, 2001.
[7] C. Rekeczky, B. Roska, E. Nemeth, and F. Werblin, "The network behind spatio-temporal patterns: Building low-complexity retinal models in CNN based on morphology, pharmacology, and physiology," *Int. J. Circuit Theory Applicat.*, vol. 29, pp. 197–239, 2001.
[8] D. Bálya, B. Roska, T. Roska, and F. Werblin, "A CNN framework for modeling parallel processing in a mammalian retina," *Int. J. Circuit Theory Applicat.*, vol. 30, pp. 363–393, 2002.
[9] S. Espejo, R. Carmona, R. Domingúez-Castro, and A. Rodrigúez-Vázquez, "CNN universal chip in CMOS technology," *Int. J. Circuit Theory Applicat.*, vol. 24, pp. 93–111, 1996.
[10] A. Paasio, A. Dawidziuk, K. Halonen, and V. Porra, "Minimum size 0.5 micron CMOS programmable 48 by 48 CNN test chip," in *Proc. IEEE Eur. Conf. Circuit Theory and Design (ECCTD'97)*, Budapest, Hungary, 1997, pp. 154–156.
[11] S. Espejo, R. Domínguez-Castro, G. Liñán, and Á. Rodríguez-Vázquez, "A 64 × 64 CNN universal chip with analog and digital I/O," in *Proc. Int. Conf. Electronics, Circuits and Systems (ICECS'98)*, Lisbon, Portugal, 1998, pp. 203–206.
[12] B. E. Shi, "Focal plane implementation of 2D steerable and scalable gabor-type filters," *J.VLSI Signal Processing*, vol. 23, no. 2/3, pp. 319–334, 1999.
[13] G. Liñán, R. Domínguez-Castro, S. Espejo, and A. Rodríguez-Vázquez, "ACE16k: A programmable focal plane vision processor with 128 × 128 resolution," in *Proc. IEEE Eur. Conf. Circuit Theory and Design (ECCTD'01)*, Espoo, Finland, Aug. 28–31, 2001, pp. 345–348.
[14] C. Rekeczky and L. O. Chua, "Computing with front propagation: Active contour and skeleton models in continuous-time CNN," in *J. VLSI Signal Processing*, vol. 23. Norwell, MA, 1999, pp. 373–402.
[15] C. Rekeczky, T. Roska, and A. Ushida, "CNN-based difference-controlled adaptive nonlinear image filters," *Int. J. Circuit Theory Applicat.*, vol. 26, pp. 375–423, 1998.
[16] K. R. Crounse and L. O. Chua, "Methods for image processing in cellular neural networks: A tutorial," *IEEE Trans. Circuits Syst.*, vol. 42, pp. 583–601, Oct. 1995.
[17] B. E. Shi, T. Roska, and L. O. Chua, "Estimating optical flow with cellular neural networks," *Int. J. Circuit Theory Applicat.*, vol. 26, pp. 343–364, 1998.
[18] T. Roska and L. Kék, Eds., "Analogic CNN Program Library," Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, Version 3.0, Tech. Rep., Analogical and Neural Computing Systems Laboratory, 2003.
[19] Aladdin Pro: Computational Environment for Cellular Visual Microprocessors (2003). [Online]. Available: http://www.analogic-computers.com/
[20] E. Niebur and C. Koch, "Computational architectures for attention," in *The Attentive Brain*, R. Parasuraman, Ed. Cambridge, MA: MIT Press, 1998, pp. 163–186.
[21] E. R. Daugherty, *An Introduction to Morphological Image Processing*. Philadelphia, PA: SPIE, 1992.
[22] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.
[23] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Circuit Theory Applicat.*, vol. 12, no. 1, pp. 43–77, 1994.
[24] J. Weng, T. S. Huang, and N. Ahuja, "Motion and structure from two perspective views: Algorithm, error analysis, and error estimation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 451–476, May 1989.
[25] J. Weng, N. Ahuja, and T. S. Huang, "Optimal motion and structure estimation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 864–884, Sept. 1993.
[26] R. I. Hartley, "In defence of the 8-point algorithm," in *Proc. 5th Int. Conf. Computer Vision (ICCV'95)*, 1998, pp. 586–593.
[27] H. C. Longuet-Higgins, "A computer algorithm for reconstructing the scene from two projections," *Nature*, vol. 293, pp. 133–135, 1981.
[28] M. V. Srinivasan, J. S. Chahl, M. G. Nagle, and S. W. Zhang, "Embodying natural vision into machines," in *From Living Eyes to Seeing Machines*, M. V. Srinivasan and S. Venkatesh, Eds. Oxford, U.K.: Oxford Univ. Press, 1997, pp. 249–265.
[29] M. Nagle, M. V. Srinivasan, and D. Wilson, "Image interpolation technique for measurement of egomotion in six degrees of freedom," *J. Opt. Soc. Amer.*, vol. 12, pp. 3233–3241, 1997.
[30] M. V. Srinivasan, S. Venkatesh, and R. Hosie, "Qualitative estimation of camera motion parameters from video sequences," *Pattern. Recogn.*, vol. 30, pp. 593–606, 1997.
[31] H. Dahmen, R. M. Wüst, and J. Zeil, "Extracting egomotion from optic flow: Principal limits for animals and machines," in *From Living Eyes to Seeing Machines*, M. V. Srinivasan and S. Venkatesh, Eds. Oxford, U.K.: Oxford Univ. Press, 1997, pp. 174–198.
[32] G. L. Barrows, J. S. Chahl, and M. V. Srinivasan, "Biomimetic visual sensing and flight control," *Aeronaut. J.*, to be published.
[33] C. Rekeczky, D. Balya, G. Tímár, and I. Szatmári, "Bio-inspired flight control and visual search with CNN technology," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'03)*, vol. 3, Bangkok, Thailand, May 2003, pp. 774–777.
[34] C. Rekeczky, I. Szatmári, G. Tímár, and D. Balya, "Adaptive multirate, multigrid and multiscale algorithms running on analogic architecture," in *Proc. IEEE Eur. Conf. Circuit Theory and Design (ECCTD'03)*, Krakow, Poland, Sept. 2003, pp. I-401–404.
[35] I. Szatmári, D. Balya, T. Gergely, C. Rekeczky, and T. Roska, "Multichannel spatio-temporal topographic processing for visual search and navigation," in *Proc. SPIE Microtechnologies for the New Millennium*, Grand Canaria, Spain, May 2003, pp. 297–306.
[36] D. Bálya, T. Gergely, I. Szatmári, and C. Rekeczky, "Classification of spatio-temporal features: The nearest neighbor family," in *Proc. IEEE Eur. Conf. Circuit Theory and Design (ECCTD'03)*, Krakow, Poland, Sept. 2003, pp. III-129–132.
[37] G. Tímár, D. Balya, I. Szatmári, and C. Rekeczky, "Feature guided visual attention with topographic array processing and neural network-based classification," in *Proc. Int. Joint Conf. Neural Networks(IJCNN'03)*, Portland, OR, July 2003, pp. 1492–1496.

Csaba Rekeczky (S'99–M'00) received the M.S. degree in electrical engineering from the Technical University of Budapest, Budapest, Hungary, and the Ph.D. degree in electrical engineering from the Budapest University of Technology and Economics, Budapest, Hungary, in 1993 and 1999, respectively.

In 1993, he was with the Analogical and Neural Computing Systems Laboratory of the Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, Hungary, working on nonlinear network theory and signal processing, computational neurobiology, and noninvasive medical diagnosis. In 1994 and 1995, he was a Visiting Scholar at the Tokushima University, Tokushima, Japan, working on cellular neural network projects related to medical image processing. In 1997 and 1998, he conducted research on nonlinear image processing and neuromorphic modeling of the vertebrate retina at the University of California at Berkeley. Currently, his research interest is focused on computational aspects of cellular nonlinear arrays and includes neuromorphic modeling of biological sensing, nonlinear adaptive techniques in signal processing, multimodal fusion, and special topics in machine vision.

Dr. Rekeczky is the co-recipient of the Best Paper Award for a contribution published in *International Journal of Circuit Theory and Its Applications* in 1998. In 2001 and 2002, he served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: FUNDAMENTAL THEORY AND APPLICATIONS.

István Szatmári (M'01) received the M.Sc. degree in electrical engineering and the Ph.D. degree in information technology from the Technical University, Budapest, Hungary, in 1995, and 2002, respectively.

He is currently a Research Fellow at the Analogical and Neural Computing Laboratory of the Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, Hungary. In 1997 and 2004, he was a Research Engineer in the Nonlinear Electronics Laboratory, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley. Since 2000, he has also been an Application Development Manager for AnaLogic Computers, Budapest, Hungary. In recent years, he has participated in several successful research and development projects supported by U.S., European, Hungarian grants. His research interests are in image processing, neural networks, and nonlinear dynamics. His development experience and competency includes mix mode (analog and digital) architecture design and measurement, multiprocessor Texas DSP environment development. He is the author or coauthor of more than 30 publications.

Dr. Szatmári received the Hungarian State Eötvös Scholarship in 1997 and the Bolyai János Hungarian State Fellowship in 1998. He is a member of the IEEE Circuits and Systems Society.

Dávid Bálya (S'99–M'04) received the M.Sc. degree in computer science from the Technical University of Budapest, Hungary, in 1999. He is working toward the Ph.D. degree in the Neuromorphic Information Technology Interdisciplinary Graduate Program at the Budapest University of Technology and Economics and the Analogic and Neural Computing Laboratory, Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary.

In 1999 and 2000, he was a Visiting Scholar at the Vision Research Laboratory, University of California at Berkeley, cooperating with neurobiologists to develop a multichannel mammalian retina model based on a complex-cell cellular nonlinear network. His main research interests include neuromorphic modeling of the visual and other sensory systems, bio-inspired spatio-temporal continuous algorithm design, and application of adaptive learning systems.

Gergely Tímár (S'03) received the M.S. degree in computer science from the Budapest University of Technology and Economics, Budapest, Hungary, in 2001. He is working toward the Ph.D. degree in the Neuromorphic Information Technology Interdisciplinary Graduate Program at the Pázmány Péter Catholic University and the Analogic and Neural Computing Laboratory, Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary.

His research interests include high-speed image processing, multitarget tracking, offline handwriting recognition, and applied artificial intelligence.

Ákos Zarándy (M'98) received the M.Sc. degree in electronic engineering from the Technical University of Budapest, Hungary, and Ph.D. degree in electrical engineering from the Hungarian Academy of Sciences, Budapest, Hungary, in 1992 and 1997, respectively.

He is currently a Research Engineer at the Analogical and Neural Computing Systems Laboratory, Computer and Automation Research Institute, Hungarian Academy of Science, Budapest, Hungary. He has been dealing with cellular neural/nonlinear networks since 1990. His research activity is currently devoted to adaptive optical sensing.