

CNN-Based Local Motion Estimation for Image Stabilization Processing and its Implementation

Chin-Teng Lin, *IEEE Fellow*, Shi-An Chen, Ying-Chang Cheng, and Chao-Ting Hong

Abstract—The objective of this paper is to investigate a novel design for local motion vectors (LMVs) of image sequences, which are often used in a digital image stabilization (IS) system. The IS technique removes unwanted shaking phenomenon in image sequences captured by hand-held camcorders. It includes two main parts such as motion estimation and compensation. Most of computation power occurs in the part of motion estimation. In order to reduce this complexity, an idea, which integrates an adaptive-threshold method and cellular neural networks (CNN) architecture, is designed to improve this problem. The design only implements the most important local motion estimation with the array size of 19×25 pixels. Experimental results with HSPICE simulation and CNUM are shown that the proposed architecture fast searches the location of possible LVMs and has the capability of real-time operations.

I. INTRODUCTION

Image stabilization (IS) processing, also known as vibration reduction, which is a digital camcorder technology that helps preventing images from blurring. It reduces vibration caused by camcorder shake, slow shutter speed or when using a long telephoto lens without a tripod. For developed video camcorders, image stabilization is finding a way into more consumer and professional camcorders. Most of IS processing are implemented on PC or FPGA. However, other consumer camcorders are only for gyro control sensor in the optical stabilization processing.

Due to the complex computation found in local motion estimation, the IS system is hard to perform in real time. In order to further reduce this complexity in motion estimation processing, the architecture of cellular neural/nonlinear networks (CNNs) [4] is considered. Comparing with the conventional digital method, the CNN-based

computation is capable of realizing the trillions of operations per second (TeraOPS)-range image processing tasks in a cost-effective implementation [5]. In this paper, we propose a CNN-based chip which could greatly reduce the heavy computation problem for location motion estimation processing. The representative point matching (RPM) [2] method is used to find the local motion vectors. The pre-processing image information will first pass a digital-to-analog (D/A) converter and store incoming data to memories. Then CNN will search the global minimum position and decide x and y coordinates by using global output connected chains. Local analog memory (LAM) is designed to store the image difference information which is passed from an 8-bit D/A converter. With the aid of CNN technique [4], [6], the global minimum position according to the RPM method could be easily generated and stored in comparing with the traditional DSP computation. The method of global output connected chains is used to connect CNN output and decode the LMV address. A reliable local motion vector based on CNN architecture is extracted to determine global motion vector and then to compensate shaking image in practical applications.

This paper is organized as follows. Section II gives a computational analysis for the IS system and then proposes the adaptive-threshold motion estimation algorithm. Section III introduces the CNN method and adaptive template design for local motion estimation. The LMV circuit realization is described in Section III. Experimental results with HSPICE, CNN universal machine (CNUM) and chip implementation are shown in Section V. Conclusions are made in Section VI.

II. MOTION ESTIMATION METHOD

A. Image Stabilization

The architecture of the proposed image stabilizer technique shown in Fig. 1 is divided into two processing blocks as motion estimation and motion compensation [11]. The motion estimation block includes three estimators as LMV, the ill-conditioned motion vector (IMV), and the global motion vector (GMV) estimators. The motion compensation unit consists of the compensating motion vector (CMV) estimation and image compensation. The two incoming consecutive images (at time $(t-1)$ and time (t)) will

This work was supported by the Ministry of Economic Affairs, Taiwan R.O.C., under Grant 95-EC-17-A-02-S1-032.

C.-T. Lin is with the Department of Electrical and Control Engineering/ Department of Computer Science, National Chiao-Tung University, Hsin-Chu, Taiwan, R.O.C., and also with the Brain Research Center, University System of Taiwan, Taiwan, R.O.C. (phone:+886-3-5712121 ext 31228; fax: +886-3-5729880; e-mail:ctlin@mail.nctu.edu.tw).

S.-A. Chen, Y.-C. Cheng, and Chao-Ting Hong are with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsin-Chu, Taiwan, R.O.C.(e-mail: cuteless.ece93g@nctu.edu.tw; chengyc.ece92g@nctu.edu.tw).

be firstly divided into four regions. A LMV will be derived in each region by the RPM algorithm [1], [3]. The motion estimation block also contains a reliability detection function. The GMV estimation determines a global motion vector among LMVs, IMV, and other pre-selected motion vectors through background-based evaluation function. Finally, the compensating CMV is generated according to the resultant GMV and the image sequences will be compensated based on the CMV in the motion compensation unit.

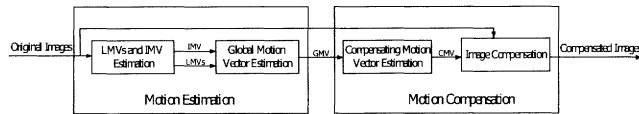


Fig. 1 : The architecture of the proposed image stabilization technique.

B. Motion Estimation

The motion estimation unit shown in Fig. 1 contains the LMVs, IMV, and GMV estimators. As shown in Fig. 2, the LMVs and IMV estimation is to generate the LMVs and IMV for global motion vector estimation [11]. A background-based evaluation function is proposed to generate GMV by using LMVs, IMV, zero motion vector (ZMV), and the past GMV. The motion estimation unit in Image Stabilizer Algorithm treats the image as 19×25 pixels' macro-blocks, and computes the motion vectors and errors separately between them. The better the motion estimation is, the better the image compensation is.

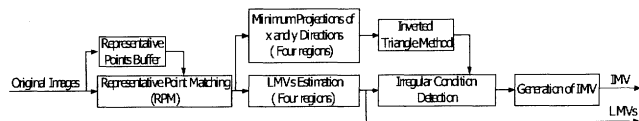


Fig. 2 : The block diagram of LMVs and IMV estimation.

In the absolute differences (SAD) matrix, all values are accumulated and then analyzed with a statistical method. The vector which calculates form the center point to the minimum SAD value position is considered as the LMV of each region.

III. CELLULAR NEURAL NETWORKS

Cellular Neural Network(CNN) is a two-dimensional analog processing array and can be considered an implementable alternative to fully connected neural networks and a remarkable improvement in hardware implementation of artificial neural networks. CNN is widely used in several application fields such as image processing and pattern recognition. Its state equation can be represented by

$$\dot{x}_{i,j}(t) = -x_{i,j}(t) + \sum_{k,j \in N_r(i,j)} A_{i,j,k,l} y_{k,l}(t) + \sum_{k,j \in N_r(i,j)} B_{i,j,k,l} u_{k,l}(t) + I_{i,j} \quad (1)$$

$$y(t) = f(x(t)) = \frac{1}{2} (|x(t) + 1| - |x(t) - 1|) \quad (2)$$

where (i, j) indicates a grid point associated with a cell on the 2-D grid, and $(k, l) \in N_r(i, j)$ is a grid point in the neighborhood within a radius r of the cell (i, j) .

If it is used for image processing, each pixel of the image corresponds to an analog processing element (cell) of CNN. The current-mode approach [7], [8] is used in CNN circuit design because it has superior mathematical addition properties. The summation of weighted currents is simply done by appropriate transistor sizing. The piecewise-linear function is achieved by cascading two current limiters as shown in Fig. 3.

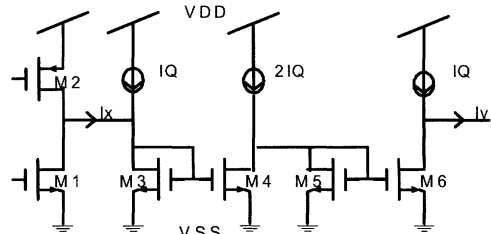


Fig. 3 : Schematic of basic CNN cell.

A. Image Difference

The first step of RPM method will subtract the present sub-region pixels with past representative point pixel color. We can implement subtraction step with image inversion and current addition. The inversion template [9] lists in Eq. (3). The input of CNN is grayscale representative sub-region.

$$T_A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad T_B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3)$$

B. Global Minimum

To search the minimum position in a specify area not only takes time but also consumes lots power. Comparing previous value and storing the minimum value is the basic processing step. The larger area need to be determined, the more clock cycle, ie., power, it takes. Therefore, we propose to use CNN adaptive threshold template with capability of finding the global minimum position in larger array and can process with less clock period. The adaptive threshold template lists in Eq. (4).

$$T_A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad T_B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{matrix} z = -z^*, \\ -1 < z^* < 1, \end{matrix} \quad (4)$$

The adaptive threshold template not only simplifies the CNN state equation, but also makes the template easier to implement in VLSI. We can write an equation to represent Eq. (5) as:

$$\dot{X} = -X + AY + Bu + I = -X + 2Y + u + I. \quad (5)$$

IV. CIRCUIT DESIGN OF CNN-BASED LMV ESTIMATION

The process of finding LMV is very computationally intensive, requiring billions of operations for each image. The steps of finding LMV are listed below.

- 1) Cut each incoming image into four equal regions.
- 2) Segment the prescribed sequence region ($t-1$) and (t) into sub regions which each of them is 19×25 pixels as shown in Fig. 4(a).
- 3) Represent all the pixels with the central point in each sub-region($t-1$). The mapping array is called representative point macro-block (RPM) as shown in Fig. 4(b).
- 4) Subtraction: $|Msub(t)| := \text{sub region}(t) - \text{RPM}(t-1)$ to provide absolute error for the $Msub(t)$ array.
- 5) Add : Add all the $|Msub(t)|$ in the prescribed region to form a 19×25 difference value array.
- 6) Min: Find the minimum difference position of the difference value array as the vibrante noise vector for the prescribed region.

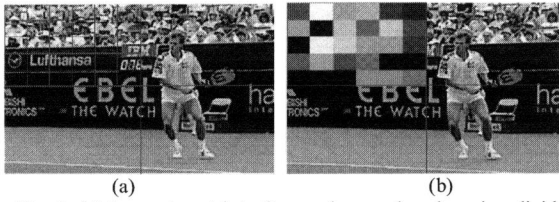


Fig. 4: (a) Image is cut into four regions and each region divides into the size 19×25 pixels. (b) Representation all pixels with the central point in each sub-region ($t-1$).

The most complexity operations occur in 1) computing the motion vector and the difference value and 2) storing the difference value with the position information if it is smaller than any previous value. Since the operation slows down the computation, the CNN architecture is suitable for motion computation and is done by CNN with a fixed template and the tunable bias current circuit for each cell. The minimum difference value and the position information can be found less than 32 clock cycle no matter how large the array is by the architecture of the circuit. The circuit architecture is shown in Fig. 5. The sequential images are segmented into a 19×25 pixels' array for each region, and do the sum of difference (SOD) process. The difference value array is then fed into the LAM of the CNN array.

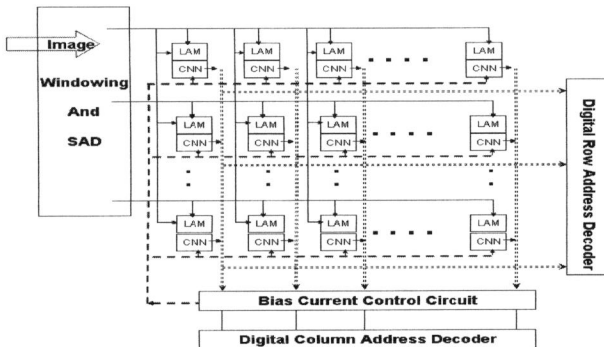


Fig. 5. AS-CNN chip architecture.

The ASCNN processing unit including the CNN bias circuit is shown in Fig. 6. With the aid of 8 bit voltage current converter, the 256 difference value information can be transformed into current and feed in to the LAM. After loading all the difference information into the LAMs, the digital control circuit will turn on the CNN array switch in order to saving power during the loading step. An adaptive-minimized threshold template is used for searching the minimum value in the CNN array. If the bias current is smaller than any input current from the CNN array, the output voltage will goes to high. Therefore the digital position decoder and bias control circuit can simply judge whether the bias current should raise or not by checking if any row or column obtain the low information. The search min architecture could save the tasks of reading out error matrix and comparing process by using DSP processor.

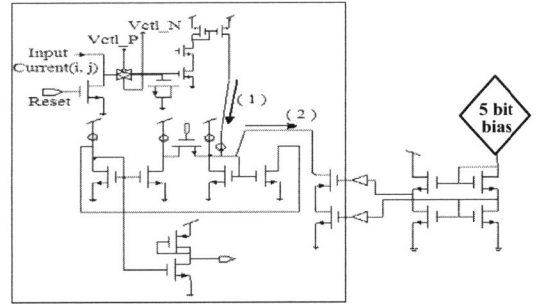


Fig. 6. Schematic of an ASCNN processing unit (inside the rectangle) with CNN bias circuit.

V. SIMULATION RESULTS

According to the proposed CNN-based local motion estimation algorithm designed in Section IV, the simulated results are verified by the transistor-level simulator and CNUM. Fluctuated video sequences with various irregular conditions are used for the testing experiment. Each video sequence contains 200 frames with the resolution of 312×200 pixels.

A. Simulation Results of CNUM

The CNUM [10] provides the framework for the definition of an algorithmically programmable analog array computer with supercomputer power on a chip. The IS algorithm could both verified by CNUM and the circuit simulation. The result of the SAD array (19×25 pixels) after CNUM processing is shown in Fig. 7(a). After analysis these SAD values, the minimum difference value is 85 pixel counts and locates in (6,4) while the top left point is defined as (1,1) The sub-image subtraction and searching global minimum can be done by using image difference, and search global minimum templates as described in section III. With proper choice of threshold bias current I_{bias} , only the minimum position appear with white color and others remain black color. The final result is shown in Fig. 7(b). In order to compare the position information with the simulation of circuit, the shifts to vertical and horizontal

templates are used and their output is shown in Fig. 7(c) and (d). The output results shows the correct coordinate (X, Y) should be (6,4) and the position is the same as in Fig. 7(a).

B. Chip Simulation Results

Figure 8(a) is the output of 19×25 difference value array image. Fig. 8(b) and (c) shows the row and column global connect chains output corresponding to the minimum SAD position. The simulation only checks the 18 LAM (contained in the dot line) value and CNN switches turns on at 78us. With the help of automatic tuning bias current circuit, the output flips from high to low with a proper threshold of CNN. Fig. 8(d) shows other global connect chains for the rest of 42 outputs.

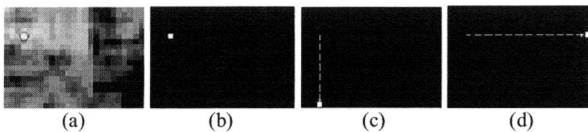


Fig. 7. (a) The output of 19×25 difference value array image. (b) The output of (a) is fed into CNNUM using adaptive minimized threshold template in Eq. (4). (c) Vertical output of (b) with shift to vertical template. (d) Horizontal output of (b) with shift to horizontal template.

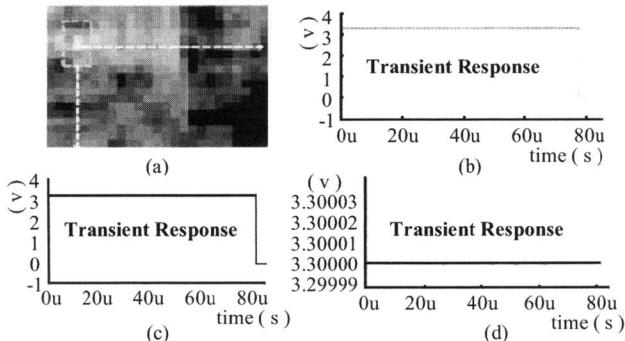


Fig. 8 : Simulation of CNN output inside the dot line. (a) The output of 19×25 difference value array image.(b) (c) Row and column global connect chains output corresponding to minimum SAD position. (d) Global connect chains for the rest of 42 outputs.

VI. CONCLUSIONS

In this paper, we propose a novel circuit design of local motion estimation integrated with the adaptive-threshold method and CNN architecture considered for image stabilization realization to reduce computational complexity. CNN is used easily on VLSI implementation. The proposed method has also superior performance compared with other DSP processors. In particular, an adaptive and adjustable bias current circuit is integrated into CNN to increase automatically the threshold level and to detect fast the global minimal SAD position. In addition, the global output connected chains are used to locate the decided LMV position without wasting extra processing time for searching addresses. Finally, the proposed CNN-based local motion estimation can not only be implemented on VLSI, but also be performed on CNNUM.

REFERENCES

- [1] C. Morimoto, D. DeMenthon, L. Davis, R. Chellappa, and R. Nelson, "Detection of independently moving objects in passive video," *Proc. of Intelligent Vehicles Workshop*, pp. 270-275, Sep. 1995.
- [2] Y. Egusa, "An application of fuzzy set theory for an electronic video camera image stabilizer," *IEEE Trans. on Fuzzy Systems*, vol. 3, pp. 351-356, Aug. 1995.
- [3] S. W. Jeon, "Fast digital image stabilizer based on Gray-coded bit-plane matching," *IEEE Trans. Consumer Electronics*, vol. 45, pp. 598-603, Aug. 1999.
- [4] L. O. Chua and L. Yang, "Cellular neural networks: theory and application," *IEEE Trans. on Circuits and Systems*, vol. 35, pp. 1257-1290, Oct. 1988.
- [5] R. Carmona Galan, A. Rodriguez Vazquez, S. Espejo Meana, R. Dominguez-Castro, T. Roska, T. Kozek, and L. O. Chua, "An 0.5- μm CMOS analog random access memory chip for TeraOPS speed multimedia video processing," *IEEE Transactions on Multimedia*, vol.1, pp. 121-135, Jun. 1999.
- [6] J. E. Varrientos, "A current-mode cellular neural network implementation," *IEEE Trans. on Circuits and Systems-II*, vol. 40, pp. 147-155, Mar. 1993.
- [7] T. Roska and L. O. Chua, "Cellular neural networks with nonlinear and delay-type template elements," *Proc. IEEE int. Workshop on Cellular Neural Networks and Their Applications*, pp. 12-25, 1990.
- [8] J. E. Varrientos, E. Sanchez-Sinencio, and J. Ramirez-Angulo, "A current-mode cellular neural network implementation," *IEEE Trans. on Circuits and Systems*, part II, vol.40, pp. 147-155, Mar.1993.
- [9] Hyongsuk Kim, T. Roska, H. Son, and I. Petras, "Analog addition/subtraction on the CNN-UM chip with short-time superimposition of input signals," *IEEE Trans. on Circuits and systems- I: Fundamental Theory and Applications*, vol.50, Mar. 2003.
- [10] T. Roska and L. O. Chua, "The CNN universal machine: An analogic array computer," *IEEE Trans. Circuits Syst. II*, vol. 40, pp. 163-173, Mar. 1993.
- [11] S. C. Hsu, S. F. Liang, and C. T. Lin, "A robust digital image stabilization technique based on inverse triangle method and background detection," *IEEE Transactions on Consumer Electronics*, vol.51, pp.335-345, May 2005.