Functional-Memory-Based Systems Enabling Recognition and Learning

Tetsushi Koide, Hans Juergen Mattausch,
Ali Ahmadi, M. Anwarul Abedin,
Takashi Morimoto, Kosuke Yamaoka

Research Center for Nanodevices and Systems (RCNS)
Graduate School of Advanced Sciences of Matter
Hiroshima University

http://www.rcns.hiroshima-u.ac.jp/

Functional Memory Based Intelligent Information System

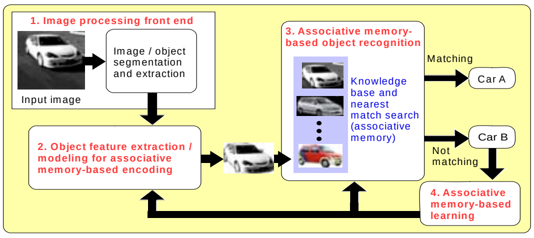
◆ The effective implementation of pattern recognition and learning, which are basic functions for building artificial systems with capabilities similar to the human brain, is of great technical and practical importance.

◆We are developing *a flexible memory-based architecture* for this purpose, which can be expected to allow intelligent processing (object-feature extraction / recognition, learning, judgment)

◆To realize the intelligent information processing, a very high system performance, which does not suffer from the memory bottleneck, can be required.

The unification of processing and memory part is a promising development direction, which can combine high performance of an integrated system with high integration density and low power consumption.

Structure of Functional Memory-Based Systems with Recognition and Learning Capability



1. Extract the object of interest from the input data. For an image as input data, this stage requires an image segmentation function and a procedure for selecting the segment (or object) of interest.
2. Prepares the data of the selected object for a comparison with the knowledge base of the system by extracting the objects characteristic features.
3. Knowledge base of the system which includes a search function for finding the best match to an input pattern from the 2nd stage.
4. The learning stage includes a feedback to the 3rd stage, the knowledge base, and possibly also to the 2nd stage for the characteristic-feature extraction.

Outline

Real-Time Moving/Still Object Tracking

Objective:
Development of a functional memory-based architecture of moving/still object detection/estimation for pre-processing of recognition systems

Application examples:

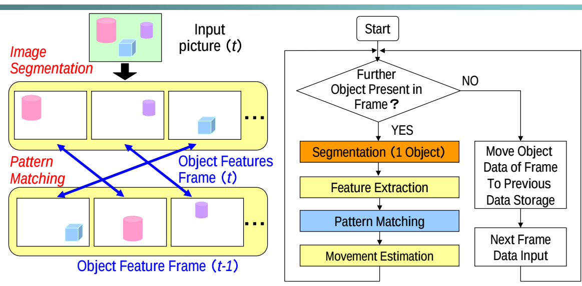
(1) Preprocessing of object recognition, detection of a suspicious person with a surveillance camera, etc

(2) Analysis of a motion of each part of a moving object, an understanding of operation of people, etc.

(3) Road/traffic environment image recognition seen from the driver's seat of a car, environment recognition by the autonomous robot, etc.

(4) High efficient data compression by separate coding of moving object from a background, etc.

Multi-Object Tracking by Segmentation and Matching



Properties of the proposed Algorithm

- Image segmentation algorithm can extract both still and moving objects.
- Rigid, non-rigid and occluding objects can be tracked.
- Due to the matching process multiple objects can be tracked simultaneously.
- Moving and still objects can be tracked even if the camera is moving.

Image Segmentation Algorithm (Region Growing Approach)

1. Connection-weight calculation : Similarity of luminance, color, and hue, etc between pixels
2. Leader cell selection : Sum of connection weights with neighboring pixels has to be larger than a pre-defined threshold
3. Segmentation : Repeat self-excitation, excitation, and inhibition
4. Labeling : Result of the segmentation of each object

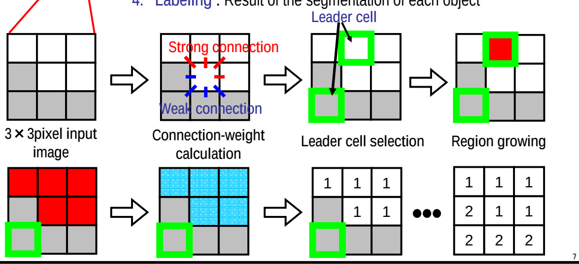
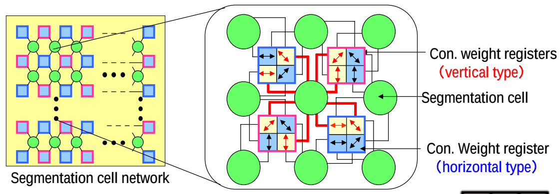


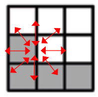
Image Segmentation Cell Network

Segmentation cell: State registers, Adders, Comparators
(Calculate whether to excite based on connection weights)

Connection weight registers: 4 registers with 3 bit storage capacity
(Transmit connection weight to segmentation cells)

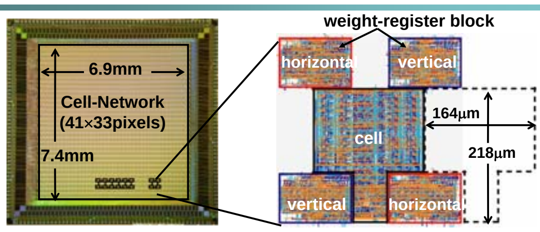


Unification of processing and memory part enables:



- Higher performance due to parallel network operation
- Small area due to sharing of weight registers
- Small interconnect length due to interleaved placement

41 × 33 Pixel Cell Network Test Chip



Technology	**0.35μm 2-Poly 3-Metal CMOS**
Measured Clock Frequency	**10MHz**
Measured Power Dissipation (3.3V, segmentation)	**45.8mW@10MHz (average)** **94.0mW@10MHz (worst)**
Segmentation Time	**34μsec@10MHz (worst)**
Pixel Integration Density	**26.5pixel/mm²**
Peak Data-Access Bandwidth	**338Gbps@10MHz**

Image Segmentation Results for Region-Growing by using connection weight among pixels (1)

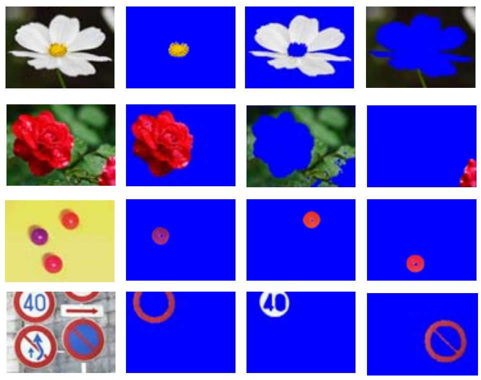


Image Segmentation Results for Region-Growing by using connection weight among pixels (2)

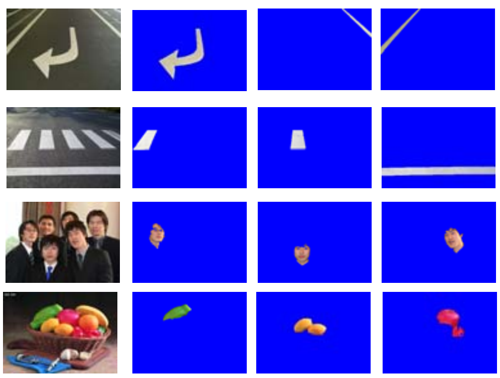
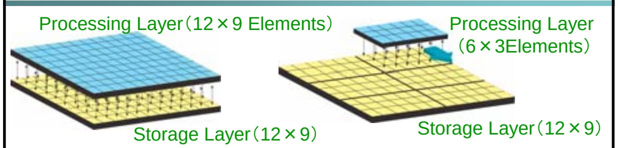


Image Scan Picture Segmentation Architecture (Concept)



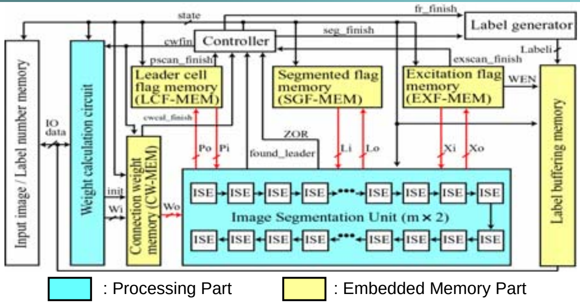
Fully parallel (Cell network) Partially Parallel

Special properties of the image scan architecture

- Input picture is divided into several blocks; parallel processing of the smaller block-pixel number, but block sequential.
- High access bandwidth embedded memories for reading and writing of the processed pixel data are utlized.
- Efficient mapping of the necessary data for block processing on the memory.

Processing block size can be optimized for each application considering the trade-off between speed and area.

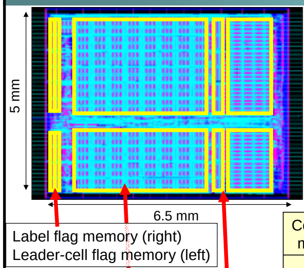
Block Diagram of Proposed Image-Scan Architecture



•Stored data for processing are read from multi-bank embedded memory with high access bandwidth in one clock cycle.

•High bandwidth of the embedded memories enables parallel processing.

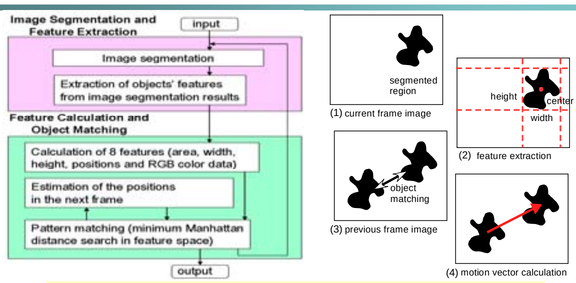
Layout of Image Scan Picture Segmentation Design



Technology	180 nm 5-Metal CMOS
Input image size	80 × 60 pixels
# of Image Segmentation Element	80×2
# of transistors	2,517,256
Core are	32.5mm^2
Max frequency	66 MHz

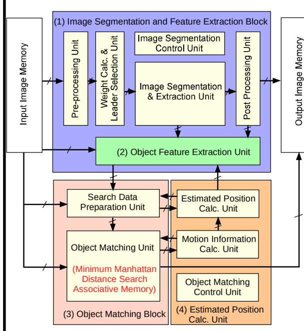
Composition of macro blocks	Word #	Word length	Bank #	Area [mm^2]
ISE			160	3.55
CW memory	32	12	240	7.61
LN memory	32	8	160	3.99
EXF memory	32	32	10	0.65
LCF memory	32	32	5	0.32
LF memory	32	32	5	0.32

Proposed Multi-Object Tracking Algorithm



- The algorithm works well for pictures including multiple moving/still objects, occlusion of objects, rotating objects, and non-rigid objects.
- If mistracking occurs at some frames (e.g. by reason of occlusion), the algorithm can recover correct tracking after a couple of frames.
- Using object feature extraction and pattern matching techniques, the algorithm can be extended to object recognition.

Multi-Object Tracking Architecture using Image Segmentation and Pattern Matching based on Functional Memories



(1) Image segmentation block
All objects of the frame are extracted.

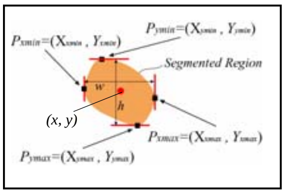
(2) Object feature extraction block
Object features for each segmented object are calculated using the image segmentation results.

(3) Object matching block
The most similar object is searched among the reference data from the previous frame (*associative memory*).

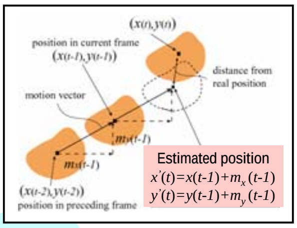
(4) Estimated position calculation block
The estimated position of each object in the next frame is calculated.

Feature Extraction of Segmented Object

Extraction of object features



Position estimation in the next frame



Estimated position
$x'(t)=x(t-1)+m_x(t-1)$
$y'(t)=y(t-1)+m_y(t-1)$

8 Object features

Area : a =(number of pixels)

Position : $x=(X_{xmax}+X_{xmin})/2$
$y=(Y_{ymax}+Y_{ymin})/2$

Size : $w=X_{xmax}-X_{xmin}$
$h=Y_{ymax}-Y_{ymin}$

Color : $R=[R(P_{xmax})+R(P_{xmin})+R(P_{ymax})+R(P_{ymin})]/4$
$G=[G(P_{xmax})+G(P_{xmin})+G(P_{ymax})+G(P_{ymin})]/4$
$B=[B(P_{xmax})+B(P_{xmin})+B(P_{ymax})+B(P_{ymin})]/4$

Motion vector calculation

$m_x(t-1)=x(t-1)-x(t-2)$
$m_y(t-1)=y(t-1)-y(t-2)$

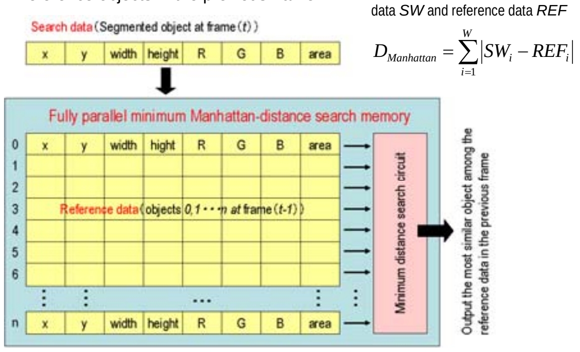
These features are extracted during image segmentation simultaneously.

Object Matching with Minimum Manhattan-Distance Search Associative Memory

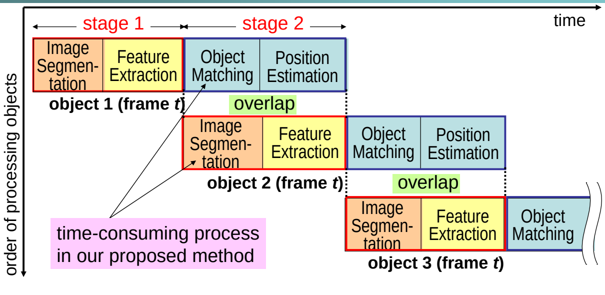
Find the most similar object among the reference objects in the previous frame

Manhattan-distance between search data *SW* and reference data *REF*

$$D_{Manhattan}=\sum_{i=1}^{W}\left|SW_i-REF_i\right|$$



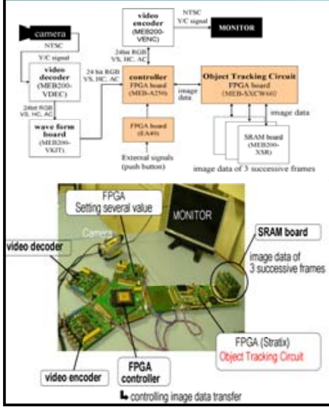
Pipeline Processing of Image Segmentation and Object Matching



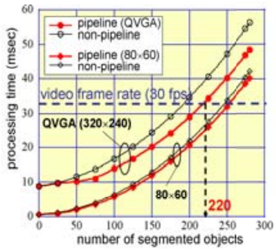
In the architecture, each object is extracted in a sequential order

By dividing image segmentation and object matching into different pipeline stages, higher processing speed can be achieved.

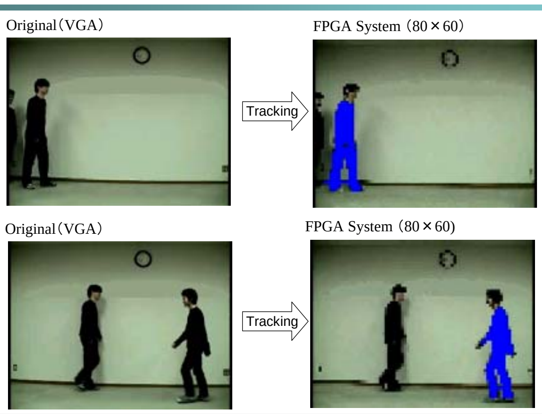
FPGA Prototype System for Object Tracking



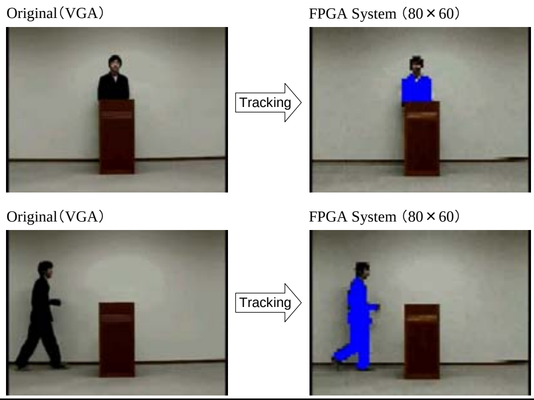
- Processing time estimation of QVGA(320×240) (@20MHz)
- About 220 objects can be tracked in real time with QVGA image size.



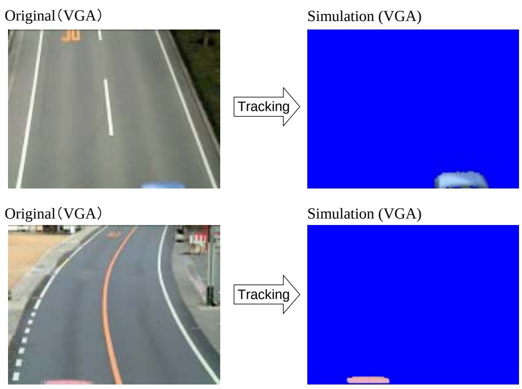
Tracking Results for Testing Videos (1)



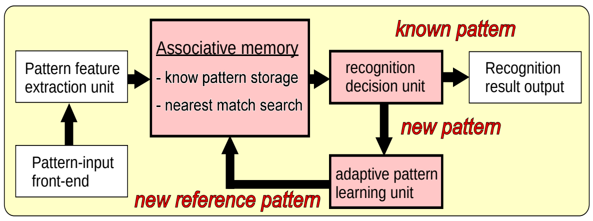
Tracking Results for Testing Videos (2)



Tracking Results for Testing Videos (3)



Associative Memory based System with Learning Capability

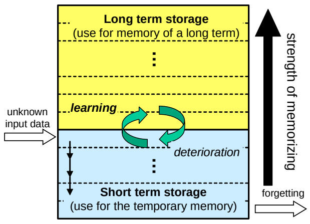


- New Learning Method, surpassing the capabilities of neural networks.
- Learning algorithm and architecture which implement the short-term and long-term memory principle of the human brain.
- Architecture which is suitable for LSI integration.
- Distance information (Hamming/Manhattan/Euclidian Distance)

The realization of a learning function is relatively easy with a minimum-distance-search associative memory.

Associative-memory-based Short / Long Term Storage

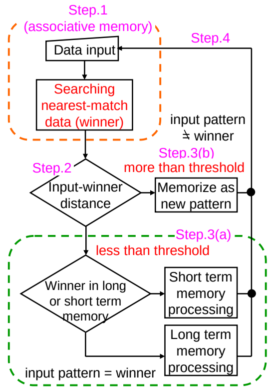
Short term storage
New information is temporarily memorized.
Long term storage
Reference pattern can be memorized for a longer time without receiving the influence of the constantly changing input patterns.



■The upper (yellow region) and lower (blue region) ranks model the long- and short-term storage, respectively.

■Expressing the memorization strength of the associative memory's reference data with a rank.

Recognition and Learning Algorithm

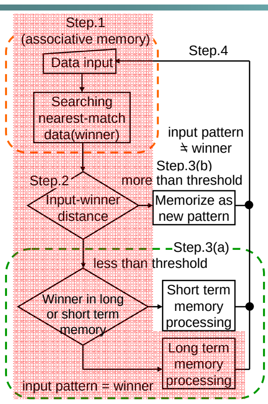


Rank

		Rank	
N_L	Long term storage	1	Ref 1
		2	Ref 2
		⋮	⋮
		N_L	Ref N_L
N_S	Short term storage	N_L+1	Ref N_L+1
		N_L+2	Ref N_L+2
		⋮	⋮
		N_L+N_S	Ref N_L+N_S

*Ref : Reference pattern

Input Pattern = Winner (in Long Term Storage)



Input-winner distance ≦ Threshold

⬇

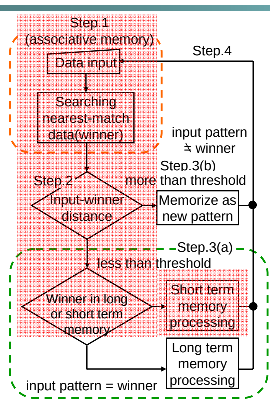
Input pattern = winner

Winner in long term storage

rank (before)		rank (after)	
1	Ref 1	1	Ref 1
⋮	⋮	⋮	⋮
34	Ref 34	34	Ref 39
35	Ref 35	35	Ref 34
36	Ref 36	36	Ref 35
37	Ref 37	37	Ref 36
38	Ref 38	38	Ref 37
39	Ref 39	39	Ref 38
40	Ref 40	40	Ref 40
41	Ref 41	41	Ref 41
⋮	⋮	⋮	⋮
64	Ref 64	64	Ref 64

J_L=5 up

Input Pattern = Winner (in Short Term Storage)



Input-winner distance ≦ Threshold

⬇

Input pattern = winner

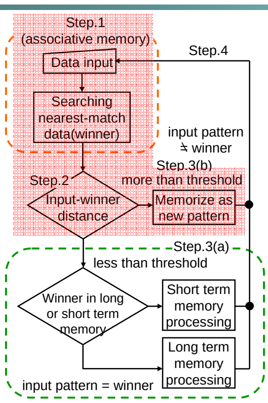
Winner in short term storage

rank (before)		rank (after)	
1	Ref 1	1	Ref 1
⋮	⋮	⋮	⋮
40	Ref 40	40	Ref 40
41	Ref 41	41	Ref 41
⋮	⋮	⋮	⋮
60	Ref 60	60	Ref 63
61	Ref 61	61	Ref 60
62	Ref 62	62	Ref 61
63	Ref 63	63	Ref 62
64	Ref 64	64	Ref 64

J_S=3 up

J_L > J_S

Input Pattern ≠ Winner (Learning New Pattern)



Input-winner distance > Threshold

⬇

Input pattern ≒ winner

rank (before)		rank (after)	
1	Ref 1	1	Ref 1
⋮	⋮	⋮	⋮
40	Ref 40	40	Ref 40
41	Ref 41	41	Input
42	Ref 42	42	Ref 41
43	Ref 43	43	Ref 42
⋮	⋮	⋮	⋮
62	Ref 62	62	Ref 61
63	Ref 63	63	Ref 62
64	Ref 64	64	Ref 63

erase

Chip Architecture and VLSI Implementation of Reference-Pattern Learning and Optimization Algorithm

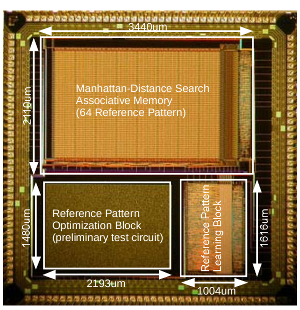
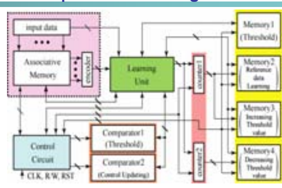
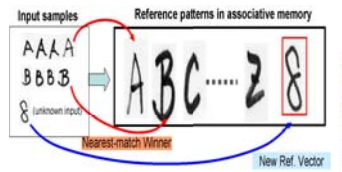


Photo of the test chip for associative memory with high-speed matching, learning and optimization of reference patterns.（0.35um 3-metal CMOS）



Distance Measure	Manhattan Distance (5bit x 16)
Reference Patterns Short Term Storage N_S Long Term Storage N_L Jump in Long Term J_L Jump in Short Term J_S	64 24 (Programmable) 40 (Programmable) 10 (Programmable) 5 (Programmable)
Nearest-Match Range	0 to 496
Technology	0.35 μm, 2-poly 3-metal, CMOS
Design Area Associative Memory Automatic Learning Circuit	11.04mm² 6.4mm² 4.84mm²
Automatic Learning Algorithm Processing Time	< 290nsec (search time 250nsec)
Automatic Learning Circuit Max Operation Frequency	166MHz (gate level simulation)

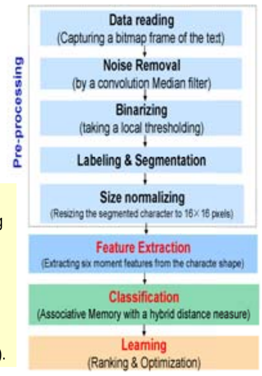
Handwritten & Printed Character Recognition with Learning Capabilities based on Associative Memory



◆To enhance the pattern matching speed, the classification process is designed on the basis of using the fully parallel associative memory.

◆The two types of distance measures (Hamming and Euclidian distances) are used for pattern matching.

◆The application oriented feature extraction is used (extracting moment features from the character shape).



Six Moment Features of Character Shape

For each segmented character following moment-based features are selected so that they have the minimum dependency on size and variation of data.

- ***Total mass*** *(number of pixels in a binarized character)*
- ***Centroid*** *(determined by an averaging on horizontal and vertical projection of character)*
- ***Elliptical parameters***
  - o ***Eccentricity*** *(ratio of major to minor axis)*
  - o ***Orientation*** *(angle of major axis)*
- ***Skewness***

In principle skewness is defined as the third standardized moment of a distribution as

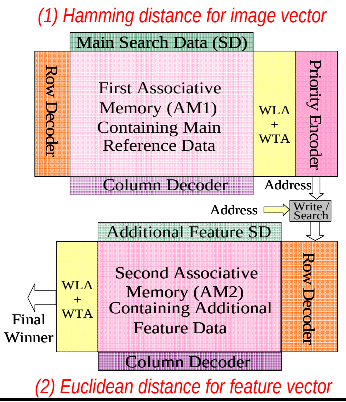
$$\gamma = \mu_3 / \sigma^3$$

but to simplify the calculations we take a simpler measure of Karl Pearson defined as

γ = 3 (mean – median) / standard deviation

and calculate horizontal and vertical skewness, separately.

Fully Parallel Associative Memory in Two-Stage Pattern Matching in Different Distance Measures

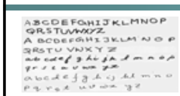


✦Hybrid distance measures like Hamming or Manhattan distance in the first associative memory and Euclidean distance in the second.

✦After searching the first *k* nearest-matched reference patterns within the first associative memory and finding the nearest-match addresses, the corresponding lines in the additional feature memory will be activated for the final winner search.

✦The second-stage winner search in the additional feature memory will be done among the *k* nearest-matched rows searched by the first associative memory.

Experimental Results: Classification and Learning



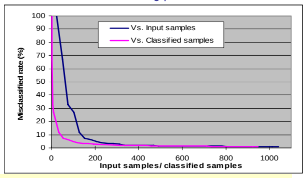
•The system performance was evaluated by using real data samples.

•A number of 35 datasets (832 samples) of English characters written by four different writers were used for the experiments.

Classification results for two test datasets after a period of learning.

Writer	Test Set 1 (26 samples)		Test Set 2 (26 samples)	
	Mis classify	New Ref. added	Mis classify	New Ref. added
1	0	0	2	5
2	1	0	1	6
3	1	1	3	7
4	1	0	2	4
Total (%)	2.8	0.9	7.4	20.4

Changes in the misclassification rate during the learning process.



●The number of patterns added as new references as well as the misclassification rate is high in the beginning of the process.

●When the learning goes on and system adjusts continuously the reference pattern memory and distance thresholds, the misclassification rate reduces dramatically.

Conclusions

◆Memory/processing-unified sub-systems for example segmentation, pattern matching, feature extraction and pattern generation as well as recognition and learning.

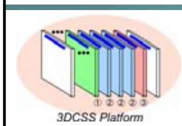
◆Real-time multi-object tracking architecture based on image segmentation and object matching and a prototype system for FPGA/ASIC implementation.

◆Learning model based on a short/long-term memory and an optimization algorithm for constantly adjusting the reference patterns and the simulation results showed an acceptable performance of classification and learning.

◆The unification approach can lead to high performance, high density solutions and is demonstrated in the picture segmentation and associative-memory applications.

◆The efficient pattern-matching performance with an associative memory leads to elegant solutions for the realization of intelligent functions like object tracking, recognition and learning in integrated systems.

Architecture Realization by 3D Custom Stack System Platform



- ➢ Large capacity bank-type associative memory
- ➢ Winner detection and recognition decision chip
- ➢ Automatic reference pattern learning and optimization chip
- ➢ Pipeline processing over multiple chips