コンピュータビジョンと
イメージメディア 103－16
（1997. 1. 24）

アクティブビジョンを応用した人物監視システム

グレゴリー ハウス†, 坂本 静生†, 高梨 伸彰‡, 田島 譲二†
†NEC 情報メディア研究所, ‡NEC 機能エレクトロニクス研究所

本論文では侵入者を探し出し、顔をズームアップして保存するアクティブ監視システムの開発について述べる。まずアクティブビジョンについての基本的な問題点について述べる。従来ゴール・オリエンテッドな画像処理・認識システムはリアルタイム処理に向き、多くの現実世界で良い性能を示してきたが、それらは各シーン特有の構造を成してしまうことが多い。そのためアクティブビジョンシステムを構築するための柔軟な枠組が望まれている。

ここでセンサー制御・画素レベル処理・領域解析・全体処理の四つのモジュールから構成される、階層的なアプローチを提案する。各モジュールの機能と新しく開発した処理について詳しく述べ、本アプローチがリアルタイム監視システム構築に大きく貢献することを示す。本枠組を人物監視システム開発に適用した。ハードウェアの概要を説明し、良好な結果を得られたことを示す。

The Human Monitoring System: A Useful Application of Active Vision

Gregory HOUSE†, Shizuo SAKAMOTO†, Nobuaki TAKANASHI‡ and Johji TAJIMA†
†NEC Information Technology Research Laboratories
‡NEC Functional Devices Research Laboratories

This paper details the research and development effort of the Human Monitoring System, which locates and tracks human subjects and obtains zoomed images of the faces. The fundamental issues in active vision research are discussed. Simple goal-oriented vision schemes have demonstrated good real-time performance in the many real-world problems. Since these solutions tend to be application specific, a flexible framework for realizing active vision systems is proposed.

The hierarchical approach consists of four modules: sensor control, pixel-level algorithms, regional analysis and global processing. Each stage including new developed analysis methods is described in detail and the novel contributions made toward the area of real-time surveillance are highlighted. The framework was applied to the development of the Human Monitoring System. A hardware realization is outlined and initial results are shown to be quite favorable.

1 Introduction

The area of "active vision" has been broadening in scope over the last several years. These are two principle components which has been making this an attractive research field. First, the recent availability of off-the-shelf cameras and motors has enabled the development of compact systems with the motion response similar to human vision. Consequently, many vision systems have been developed that incorporate the control of camera position and other sensing parameters[1]. Second, the computational complexity of traditional model-based computer vision approaches is currently impractical for many real-time applications [2]. Simple goal-oriented vision schemes which are enhanced by camera motion have demonstrated robust performance for many real-world problems [3].

There are four functions necessary in active vision systems: *signal selection, signal compression, signal representation* and *signal extraction. Signal selection* addresses the problem of sensing a 3D environment with a video camera which produces a sequence of 2D images. It chooses the relevant portion of the scene based on the requirements of the vision task. *Signal compression* removes redundant information inherent in the video sequences. It utilizes low level image processing algorithms to extract features relevant to the vision problem. *Signal representation* formulates a reduced model of the scene from the compressed signal. Limited scene understanding is essential to real-time sensor planning. Finally, *signal extraction* is depends on the requirements of the off-line processing or storage.

The organization of this paper is as follows. Section 2 outlines a four-stage general framework for an active vision. Sections 3, 4, 5, and 6 discuss each of these stages and highlights the new contributions made in these areas. Section 7 details the hardware and software implementation for the human monitoring system. Sections 8 and 9 are the performance evaluation and summary.

2 Hierarchical active vision

The necessary functions of active vision were realized in a four-stage hierarchy, which is detailed in Figure 1. *Sensor control* performs the low level control of camera parameters to regulate the signal selection of the scene. It is logically the first module in the hierarchy since the camera interfaces directly with scene. This is followed by *pixel-level algorithms* which perform computationally intensive computations to compress the video signal. The multiple color, space and time components are combined using image processing algorithms which operate on local pixel neighborhoods. The *regional analysis* module formulates a reduced scene representation from the pixel-level results. Object and scene models are vital to generating a scene representation which would useful for executing specific the vision tasks. The *global processing* stage performs higher level sensor planning using multiple time instances of the regional scene information. The location of the next observation is sent back to the sensor control module and relevant information is extracted for off-line processing. Vision modeling parameters are also feedback to the image processing modules to modify vision task objectives.

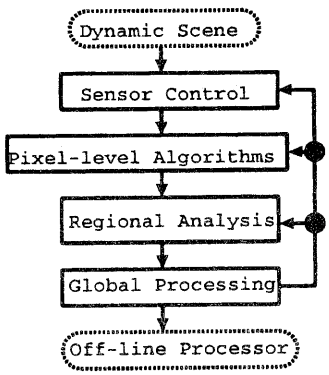

Figure 1: Hierarchical approach to active vision.

3 Sensor control

The most basic premise of an active vision is incorporating the ability to modify sensing parameters into the system design. The sensor control module performs low-level control of camera positioning parameters (pan, tilt, vergence, etc) and optical imaging parameters (aperture, focus, zoom). It translates the desired sensor settings feedback from the global processing stage into realizable steps.

Camera control routines need to balance between high resolution and a large field of view. We implemented a scheme which regulates pan velocity as a function of distance from the center of view, as shown in Figure 2. The camera response is greatest for targets near the peripheral since the possibility of losing the target is greatest, and decreases to the center where accurate tracking is primary concern (high resolution). The camera speeds are biased by target velocity (v_t) so velocities will match when the object is centered. A similar scheme was applied to tilt velocity control.

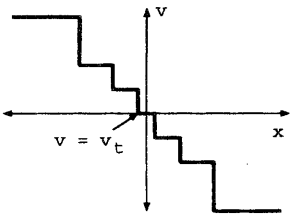

Figure 2: Pan velocity as a function of target deviation from center of view.

4 Pixel-level algorithms

An important aspect of any vision problem is reducing the magnitude of redundant information into an organized manner. We are particularly interested in algorithms with small time and space kernels to facilitate real-time implementation. For the human monitoring problem, two relevant features with straight forward implementation are skin color and motion detection. Skin color detection is useful since background scenes tend to have hues with a different color consistency [5]. Motion is also a reasonable choice since human subjects tend to move in a stationary environment.

4.1 Color detection algorithm

The simplest approach to color segmentation is directly thresholding the color space components [6]. The difficulty arises that global thresholds often yields object outlines with holes, as illustrated in the bottom of Figure 3A. We developed a technique that propagates binary values diagonally in downward and upward passes (Figures 3A and 3B). Holes are filled by reassigning zero value pixels to one if the values propagated from the diagonal directions are one (Figure 3B). The results of the two passes are "ANDED" together to prevent excessive bridging between adjacent objects. Efficient implementation of scan based algorithms on a Linear Processor Array (LPA) architecture is discussed in Section 5.

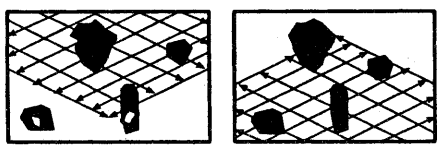

Figure 3: Scan technique for filling mask holes.

Another problem with using global segmentation thresholds is the tendency to to produce false positives in the background. We propose a two tier approach. First, a loose constraint is used used to segment clean outlines of object boundaries as in Figure 3B. Second, a tighter constraint is used to detect objects of interest such as a face or arm illustrated in Figure 4A. Segmented blobs (Figure 3B) which do not contain a minimum number of detected pixels (Figure 4A) are removed as seen in Figure 4B. The blobs that remain should be clean outlines of objects of interest.

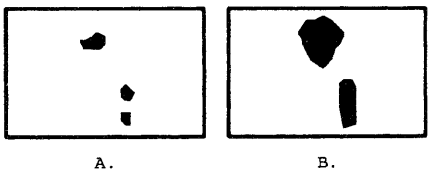

Figure 4: False object removal from mask.

4.2 Motion detection algorithm

Standard optical flow algorithms will detect the combined effect from camera and individual object motion. It is best to correct for camera motion before computing flow values since motion detection algorithms tend not to be quantitatively accurate. We implemented a heuristic approach for compensating the derivative based estimation for camera motion. Translational motion was eliminated by registering the time referenced images before computing time derivatives. Other images distortions were corrected by using the minimum time derivative in the neighborhood around each pixel. The simplified algorithm computes optical on the IMAP-VISION board is less than 1.8 ms.

5 Regional analysis

The regional analysis module is to develop a full scene representation from the pixel-level processed data. A robust method for real-time tracking humans is blob detection with statistical moment calculations [8]. This can be applied to the skin color detection output to find faces, arms and hands or to motion or background detection to locate silhouette of people. However, multiple object blob perception can be difficult to implement in real-time because computationally intensive labeling techniques are generally needed. Several orders of central moments increase the complexity further [9]. We developed a comprehensive scheme which simultaneously computes several orders of central moments for multiple targets without the need for labeling.

5.1 Parallel moment blob analysis

The standard moment equation for discrete images has the following form in cartesian coordinates [9],

$$M_{pq} = \sum_{x=1}^{X} \sum_{y=1}^{Y} x^p y^q f[x,y], \quad (1)$$

where $f[x,y]$ is the image, p and q denote moment order, and X and Y indicated image size. It is convenient to decompose (1) into one dimensional data scans along the directions of the coordinates [9],

$$M_q[x] = \sum_{y=1}^{Y} y^q f[x,y], \quad (2)$$

and

$$M_{pq} = \sum_{x=1}^{X} x^p M_q[x]. \quad (3)$$

A serial processor could implement moments by summing down each column of the image in (2) and summing up the column totals in (3). Single moment scans can be rewritten in terms of central moments M_m with respect to a reference coordinate n,

$$M_m[n] = \sum_{r=1}^{N} (r-n)^m f[r], \quad (4)$$

where r is the directional coordinate, m is the moment order and $f[r]$ is the input image in (2) or (3). The result can be decomposed into two more data scans and a correction factor,

$$M_m[n] = \sum_{r=1}^{n} (r-n)^m f[r] + \sum_{r=n}^{N} (r-n)^m f[r] - (r-n)^m f[n]; \quad (5)$$

which can be rewritten as

$$M_m[n] = M'_m[n] + (-1)^m M'_m[N-n] - (r-n)^m f[n], \quad (6)$$

where

$$M'_m[n] = \sum_{r=1}^{n} (r-n)^m f[r]. \quad (7)$$

The correction factor applies only to the zeroth order since $(r-n)^m f[n]$ is nonzero only when $m = 0$. (6) is a convenient form since it allows the moment to be computed in reference to each point on the line segment by summing the results of two opposing scans of (7). The central moments for the line segment are found by sampling the pixel center moments stored at the pixel corresponding to the center of mass. To simplify the calculations, the central moments scans of higher orders in (7) can be solved in terms of previously computed moments. For example, the first order moment scan can be decomposed as

$$M'_1[n] = \sum_{r=1}^{n-1} (r-n+1) f[r] - \sum_{r=1}^{n-1} f[r], \quad (8)$$

yielding

$$M'_1[n] = M'_1[n-1] - M'_0[n-1]. \quad (9)$$

Similar decompositions can be derived for the higher orders in terms of previously computed moments.

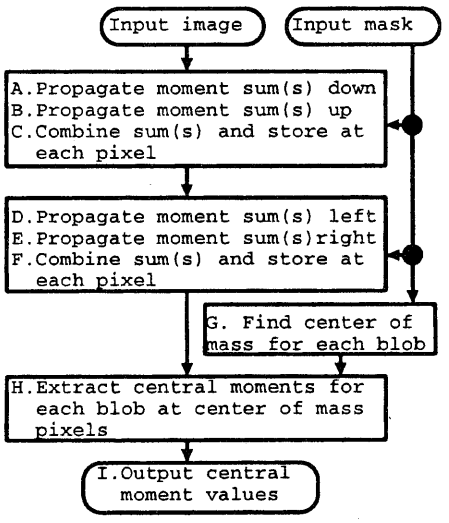

Figure 5: Fast technique for computing moment statistics for multiple blob objects in a scene.

The method for finding central moments for multiple objects in a single image is detailed in Figure 5. It utilizes the input image and a mask image generated by pixel-level algorithms (Section 4). Moment values of form (9) are first computed and are stored at each pixel by scanning the input image in downward direction (Figure 5A). Moment sums are reset to zero whenever a zero pixel is encountered in the mask image so that information does not propagate between blobs. The input image is similarly scanned in the upward direction using the mask image (Figure 5B). The vertical moments are thus computed for each pixel by summing corresponding moments (Figure 5C) as in (6). Next, the vertical moments results and the mask images are used to compute the horizontal moments through scans in the left and right directions (Figures 5D and 5E). The central moments are computed for each pixel (Figure 5F) by summing corresponding moments as in (6). Finally, the moment values are sampled at the center of mass locations (Figure 5G) to find the

central moments for each blob in the scene (Figure 5H and 5I). This is possible since the moments values computed for each pixel in the scene are normalized to the location. Note, a scan based technique can be devised to find the center of mass for each blob using distance transforms.

5.2 Special hardware implementation

The use of data scans has implementation advantages for Linear Processor Array (LPA) such as the IMAP-VISION[4]. We adapted the moment based calculations so that scan lines in a single direction can be computed simultaneously. This was done by rotating the scan direction 45° relative to the coordinates of the input and mask images (as shown in Figure 6). Forward diagonal scans (Figure 6A) are implemented by propagating data to the right as the LPA processes data downward or propagating data left as LPA processes data upward. The reverse directions are used for the reverse diagonal scans (Figure 6B). The summation of different moments scans can be likewise be implemented in parallel. This approach was used to compute six moments orders (M_{00}, M_{01}, M_{10}, M_{11}, M_{02}, M_{20}) for up to 10 objects in 5.2 ms.

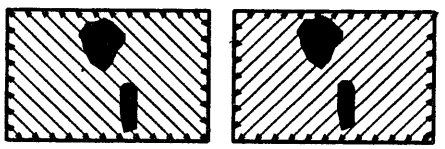

Figure 6: Diagonal scan moment implementation.

6 Global processing

The global processing stage conducts high level sensor planning, which is the most important aspect of any active vision system. It defines the sequence of behaviors that the camera head will exhibit and hence it is the most influenced by the vision goals out of all the stages Figure 1. The four relevant system behaviors to human monitoring are: *object detection, object selection, object tracking* and *object extraction. Object detection* involves active searching of the surveillance region until a possible object of interest is found. *Object selection* involves choosing one (of possibly many objects) to follow. *Object tracking* entails systematic estimation of the objects position so it can be tracked. Finally, *Object extraction* retains images and detailed feature information for the off-line processing stage.

6.1 Object detection

Active vision implies a conscious effort to obtain information through camera motion. The object detection mode integrates a limited amount of priori information about human behavior and scene layout. This includes scanning at a constant but quick pace since humans are known to move erratically and pausing at doors were subjects enter and exit. This approach yields a search pattern similar to standard moving surveillance camera. As seen in Figure 7, the camera sweep behavior continues until the moment analysis module locates an object of a minimum size (i.e., 64 pixels).

6.2 Object selection

The object selection mode uses the moment calculations from regional analysis stage to choose objects for tracking. Specifically, it selects the objects whose zeroth (M_{00}) and first (M_{11}) order moments minimized the cost function,

$$C = s(M_{00} - m_0)^2 + (1 - s)(M_{11} - m_1)^2 \quad (10)$$

where s is a scalar ($0 \leq s \leq 1$), m_0 and m_1 are reference values (i.e., 3000 for a medium ranged object). The scales s, m_0 and m_1 varied with time to change the size (i.e. depth) of the object as well as the relative shape. *Object selection* is implemented for 1 second to give time for the camera to home in on a target.

6.3 Object tracking

Object tracking techniques tend to sensitive to object scale and rotation changes. The *object tracking* mode attempts to overcome this problems by only comparing the moment criteria between successive time iterations. It selects the object which minimizes

$$C = s(M_{00} - M'_{00})^2 + (1 - s)(M_{11} - M'_{11})^2 + r(COM_x - 128)^2 + r(COM_y - 120)^2, \quad (11)$$

where s and r are scalars, 128 and 120 is the center of view in the x and y directions, and $'$ denotes the moment values from the tracked object in the previous iteration. A balance between the shape and position scalars (s and r) allow the cameras to track objects as it passes in front of others. As seen in Figure 7, The systems provides for limited time period to extract data about the object before returning to the object selection module. This ensures the camera switches between subjects when more than one person is in view. The control sequence returns to *object detection* mode (Figure 7)

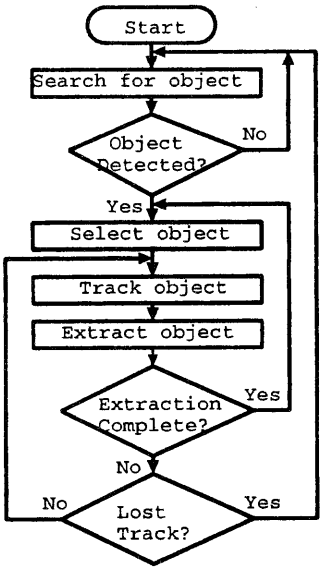

Figure 7: Control sequence of the active system's global processing module.

when the track is lost completely such as when a person exits the surveillance area.

6.4 Object extraction

The final mode of behavior, *object extraction*, is concerned with the processing of detailed images from the zoom camera. The image with the best scaled and centered face was selected using moment criteria by minimizing the cost function,

$$C = s(M_{00} - m_0)^2 + (1-s)(M_{11} - m_1)^2 + r(COM_x - 128)^2 + r(COM_y - 120)^2, \quad (12)$$

where symbols are the same as those in (10) and (11). Furthermore, size of the image was digitally scale using (10) so face images would have approximately the same size regardless of depth.

7 System realization

The center piece of any active vision development system is the multiple camera robotic head. The active vision head we used consists of three Sony XC-999 video cameras with two wide angle lens mounts (6 mm focal length) and zoom lens (10 mm to 100 mm focal length) [10]. The degrees of freedom include zoom, focus, tilt, vergence and pan. We note that the tests primarily use tilt and pan motors during the demonstration discussed in the next section.

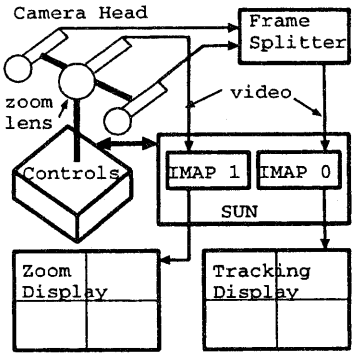

Figure 8: Active camera system hardware.

The hierarchical technique discussed in Figure 1 was realized by the camera head and support hardware detailed in Figure 8. The camera head control functions were implemented by a controller interfaced to a Sun Workstation. Low level control functions were directed by the workstation through c libraries. Pixel level and regional analysis image processing functions were coded in assembler on two IMAP-VISION boards mounted inside the Sun [4]. One board performed tracking for the two external cameras using a video splitter and the other directly processed and extracts zoom lens images. The output signal were displayed on tracking and zoom monitor displays. Sensor planning control routines were implemented on the workstation in c and accessed IMAP-VISION boards image processing results through c libraries.

8 Performance evaluation

The evaluation of an active vision system in itself is an challenging problem. The image processing and control functions can not be easily separated. All the modules have to be evaluated using a real environment since the performance of each algorithm ultimately affects the video sequence obtained by the camera. Furthermore, active vision system is designed to exercise a complex sequence of behaviors which are difficult to measure objectively. The precise manner that the scene changes can drastically effect the response, requiring an extensive amount of testing to validate the performance.

Surveillance tests of the active camera system were conducted using a 7 square meter room. People were allowed to enter in irregular intervals and numbers and were encouraged to actively evade the camera system. The wide angle cameras viewed only half the room at any given time requiring ac-

tive sensing to detect subjects entering or leaving. Furthermore, the system was geared for face extraction at about 170 centimeters height with a range from 1 to 3 meters.

8.1 Sensor control performance

It is worthwhile to evaluate the systems performance in terms of the four main components described in Figure 1. As discussed in Section 2, sensor control module is responsible for the low level control the cameras position. In the system tests, the motor control was reasonably smooth for transitions over a wide range of velocities. The camera head slowed when centering the subjects of interest were centered so clear images with minimal blur were obtained. This is particularly important for the zoom lens camera which is particularly sensitive to the slightest blurring. The system also responded quickly to subjects exiting the field of view so that a minimal number of tracks were lost. Subjects were able to occasionally elude the active camera because the communication delay and maximum motor acceleration limited the response system response.

8.2 Pixel-level performance

A typical frame of the split screen tracking display is illustrated in Figure 9. The upper left and right squares illustrate the view from the camera head and of the camera head. The lower left square displays the color detection output. Saturated regions denote detected skin color where as black areas denote suppressed background blue color. The skin color detection performance overall was fairly consistent for a large number of subjects covering a range of skin tones. The lower right hand square displays motion detection where 128 represents no motion. As seen in Figure 9, only the left subject was detected since he was the only one in motion.

8.3 Regional analysis performance

The regional analysis module simultaneously located the center of mass and first and second order moment orders of blobs in skin color detection mask. The center of mass locations for the five largest objects were denoted by bright spots. The system consistently identified the locations of multiple face and arms regions as seen in the upper left square of Figure 9. The higher order moments of the ten largest blobs were successfully used to track targets over multiple time frames. Noise from the color detection algorithm did result in noisy moment computations, but the inclusion of several orders made the tracking algorithm reasonably robust.

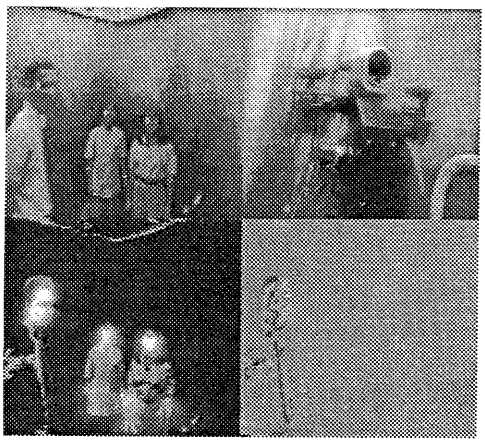

Figure 9: Tracking display with view from/of the camera head (upper left/right) and image color/motion detection outputs (lower left/right).

8.4 Global processing performance

The global processing tracking component exhibited a wide range of desirable surveillance behaviors: searched for prospective subjects; tracked candidate subjects around the surveillance region, and periodically switched between subjects when multiple persons were present. The system performed well when the scene contained up to four subjects. When large crowds were present, the system merged faces in the front and back rows as well as missed faces on the the peripheral.

The global face extraction mode also exhibited reasonable performance characteristics. Moments were used to analyze the 30 frames obtained each second and select one according to moment centering and size criteria. The image was normalized to fit on a quarter of the split screen display. The image was outputed to one of the zoom display squares if it contained a face. The location for writing the image to the monitor changed in a clockwise manner if moment parameters significantly varied from the previous iteration. Consequently, successive images of the same face would overwrite the same location and a new face would appear in a new location. This is demonstrated in Figure 10 where four different faces of similar size are displayed simultaneously.

9 Summary

This paper introduced a flexible framework for active vision consisting of four modules: sensor control, pixel-level algorithms, regional analysis and

Figure 10: Zoom lens display with four different close facial images.

global processing. Each stage is described in detail and the novel contributions made toward the area of real-time surveillance were highlighted. The *sensor control* module utilizes a trade-off compromise between quick response to capture objects leaving the view and smooth response to accurately track objects in view. Real-time implementation issues for color and motion detection *pixel-level algorithms* were outlined. A *regional analysis* technique for simultaneously computing the moments of several blobs in a color thresholded image is presented. Finally, the *global processing* module which uses the moment information to track objects and extract images was described.

The framework was applied to the development of a human monitoring system. The report outlines the hardware realization and discusses the successful during initial tests. The control module implemented global processing commands in a smooth but quick manner. The pixel-level image processing techniques successfully identified the face and body regions of person in the scene. The moment based module located the position of multiple subjects simultaneously to develop a reduced representation of the scene. The global processing module used the object information to perform target tracking for the control module, and image/data extraction for off-line processing or storage. All of these tasks were realized in real-time.

Acknowledgement The authors would like to thank Yoshihiro Fujita and Sholin Kyo of NEC for their assistance and support.

References

[1] N. Kita. Active vision systems using human vision as inspiration. *Journal of Information Processing Society of Japan*, 36(3):264–272, March 1995.

[2] T.S. Huang. Computer vision: evolution and promise. In *Imaging science and technology, Evolution and promise*, pages 13–20. 5th International conference on high techology, September 1996.

[3] J. Aloimonos. Purposive and qualitative active vision. In *Image Understanding Workshop*, pages 816–828, 1990.

[4] Y. Fujita, N. Yamashita, and S. Okazaki. Imap-vision: An SIMD processor with high-speed on-chip memory and large capacity external memory. In *IAPR Workshop of Machine Vision Applications*, 1996.

[5] D. Sanger, N. Tsumura, H. Haneishi, and Y. Miyake. Face extraction using lip detection algorithm. In *Imaging science and technology, Evolution and vpromise*, pages 221–229. 5th International conference on high techology, September 1996.

[6] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Yen. A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, 41:233–260, 1988.

[7] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[8] A. S. Pentland. Machine undertanding of human action. In *Human Communication Technologies for Substantiating Information Infrastructure*, volume 7, pages 3.1–3.12, November 1995.

[9] R. Prokop and A. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP Graphical Models and Image Processing*, 54(5):438–460, September 1992.

[10] S. Sakamoto, N. Takanashi, and J. Tajima. Active stereo camera with fast variable baseline mechanism and real-time object detection/tracking based on object inherent color. In *IEICE Proceedings*, volume D, page 387, March 1996.