Finger Tracking as an Input Device for Augmented Reality

James L. Crowley
IMAG-LIFIA
46 Ave Félix Viallet
38031 Grenoble, France

François Berard and Jöelle Coutaz
IMAG-LGI
B.P 53
38041 Grenoble CEDEX 9, France

Abstract

This paper concerns techniques for visual tracking of pointing devices. The first section introduces the motivation by describing the potential for applying real time computer vision to man machine interaction. The problem of tracking is then addressed as a problem of optimal signal detection. This approach provides a method in which the most probable location of the pointing device is determined by searching for the image position at which the sum of squared differences with a reference template is minimized. The problems of choosing the size of the reference template and the search region are addressed. A method is provided to detect when to initiate tracking as well as to determine when tracking has failed. The problem of updating the reference mask is also discussed. These techniques are illustrated with a visual tracking program called FingerPaint.

1. Computer Vision and Man Machine Interaction

One of the effects of the continued exponential growth in available computing power has been an exponential decrease in the cost of hardware for real time computer vision. This trend has been accelerated by the recent integration of image acquisition and processing hardware for multi-media applications in personal computers. Lowered cost has meant more wide-spread experimentation in real time computer vision, creating a rapid evolution in robustness and reliability and the development of architectures for integrated vision systems [Cro94].

Man-machine interaction provides a fertile applications domain for this technological evolution. The barrier between physical objects (paper, pencils, calculators) and their electronic counterparts limits both the integration of computing into human tasks, and the population willing to adapt to the required input devices. Computer vision, coupled with video projection using low cost devices, makes it possible for a human to use any convenient object, including fingers, as digital input devices. In such an "augmented reality" [Wel93a] information is projected onto ordinary objects and acquired by watching the way objects are manipulated. A simple example of augmented reality is provided by the "digital desk" [Wel93b].

In the digital desk, illustrated in figure 1, a computer screen is projected onto a physical desk using a video-projector, such as a liquid-crystal "data-show" working with standard overhead projector. A video-camera is set up to watch the work area such that the surface of the projected image and the surface of the imaged area coincide.

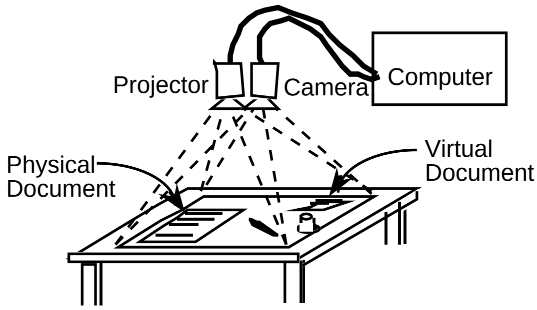

Figure 1 The Digital Desk (after [Wel93b]).

The projective transformation between the work-space (or screen) coordinates ${}^{s}P = (x, y, 1)$ and the image coordinates ${}^{i}P = (i, j, 1)$ is easily described as a reversible perspective transformation represented by a 3x3 homogeneous coordinate matrix:

$$\begin{bmatrix} w\,x \\ w\,y \\ w \end{bmatrix} = {}^{s}_{i}\mathbf{M} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix}$$

The letter "w" represents the deformation due to perspective. This notation permits the screen coordinates of ${}^{s}P$ to be recovered as a ratio of polynomials expressed in image coordinates ${}^{i}P$. That is, for a matrix ${}^{s}_{i}\mathbf{M}$ composed of 3 rows, ${}^{s}_{i}M_1$, ${}^{s}_{i}M_2$ and ${}^{s}_{i}M_3$:

$$x = \frac{w\,x}{w} = \frac{{}^{s}_{i}M_1 \cdot {}^{i}P}{{}^{s}_{i}M_3 \cdot {}^{i}P}$$

$$y = \frac{w\,y}{w} = \frac{{}^{s}_{i}M_2 \cdot {}^{i}P}{{}^{s}_{i}M_3 \cdot {}^{i}P}$$

If the viewpoint of the projector and camera are very close, the denominator of this projection can be approximated by a constant, s, giving an affine or "weak perspective" transformation from the image to the workspace. In this case:

$$x = \frac{1}{s}\left({}^{s}_{i}M_1 \cdot {}^{i}P\right)$$

$$y = \frac{1}{s}\left({}^{s}_{i}M_2 \cdot {}^{i}P\right)$$

The coefficients of this affine transformation, ${}^s_iM_1$ and ${}^s_iM_2$ and the scale factor, s, can be determined by observing the image position of the four corners of workspace [Cro93]

The visual processes required for the digital desk are relatively simple. The basic operation is tracking of a pointing device, such as a finger, a pencil or an eraser. Such tracking should be supported by methods to determine what device to track and to detect when tracking has failed. A methods is also required to detect the equivalent of a "mouse-down" event for selection.

The tracking problem can be expressed as: "Given an observation of an object at time t, determine the most likely position of the same object at time t+ΔT". If different objects can be used as a pointing device, then the system must include some form of "trigger" which includes presentation of the pointing device to the system. The observation of the pointing device gives a small neighbourhood, w(n,m), of an image p(i, j). This neighbourhood will serves as a "reference template". The tracking problem can then be expressed as, given the position of the pointing device in the k^{th} image, determine the most likely position of the pointing devise in the $k+1^{th}$ image.

For implementation reasons, we have chosen to use a square neighbourhood of size N by N. The origin of this neighbourhood is the upper left corner. A point at (0, N/2) is designated as the "hot-spot". The size of the tracked neighbourhood must be determined such that the neighbourhood includes a sufficiently large portion of the object to be tracked with a minimum of the background.

The image at time (k+1)ΔT to be searched will be noted as $p_{k+1}(i, j)$. The search process can generally be accelerated by restricting the search to a region of this image, denoted s(i,j), and called a "Region of Interest" or ROI. Our system uses a square search region of size M by M, whose center is denoted as (i_o, j_o). The center corresponds to the location where the reference template was detected in the previous image, although using Kalman filter for tracking we could easily predict the next position based on an estimate of the current velocity [Cro89].

We have experimented with two different approaches to tracking pointing objects: correlation tracking and active contours (or snakes)[Ber94]. The active contour model [Kas87] presented problems which we believe can be resolved, but which will require additional experiments. Because of this, plus space limitations, in this paper we present only techniques for correlation tracking.

2 Tracking by Correlation

The tracking problem can be expressed as a problem of optimal signal detection [Woz65]. In the simplest such formulation, the pixels serve as basis vectors, and the decision rule for matching is based on minimizing a sum of squared differences. This approach can be shown to provide a minimum probability of error for a signal corrupted by additive white noise. The the proof of optimality is only valid for additive white noise, experience shows that the technique is quite robust in the presence of other common noise sources.

The optimum receiver formulation leads to tracking by correlating a reference template with a region of the image. However, this model leaves a number of implementation details to be determined. These implementation details depend on the application domain and thus require experimentation.

Correlation has been occasionally used in computer vision since the 1960's. However, its use has generally been rejected because it does not provide a general solution for view-point invariant object recognition. In addition, the hardware to support real time implementation of correlation has only recently become sufficiently low cost to be of general use.

Tracking of pointing devices for the digital desk provides a number of simplifications that make the use of correlation well suited. For example, the illumination of the workspace is controlled and generally uniform. The device to be tracked remains close to a 2D surface and thus its appearance changes little. Change in view point is limited to (slow) rotation of the template within the 2D workshop. Under these conditions, correlation tracking provides an easy implementation for real time operations and can be accelerated by special purpose hardware.

2.1 Correlation and SSD

In the signal detection formulation for tracking, a reference template, w(i,j), is compared to all neighbourhoods within the search region, s(i, j) of a received signal $p_k(i, j)$ centred on a pixel (i_o, j_o), as shown in figure 2. The pixel (i_o, j_o) represents the position at which the tracked object is expected to be found, based on previous observations.

The optimum receiver requires that the received image and reference signals be expressed in an orthogonal basis set. The pixels which make up an image provide just such a basis set. A well known result from signal detection theory shows that for additive white noise, the probability of error can be minimized by minimizing the sum of squared difference between the reference and received signal expressed in the chosen basis space. In terms of searching for the new position of the object, this can be expressed mathematically as determining the position (i_m, j_m) within the search region s(i,j) which minimizes the sum of squared difference, as shown in equation 1.

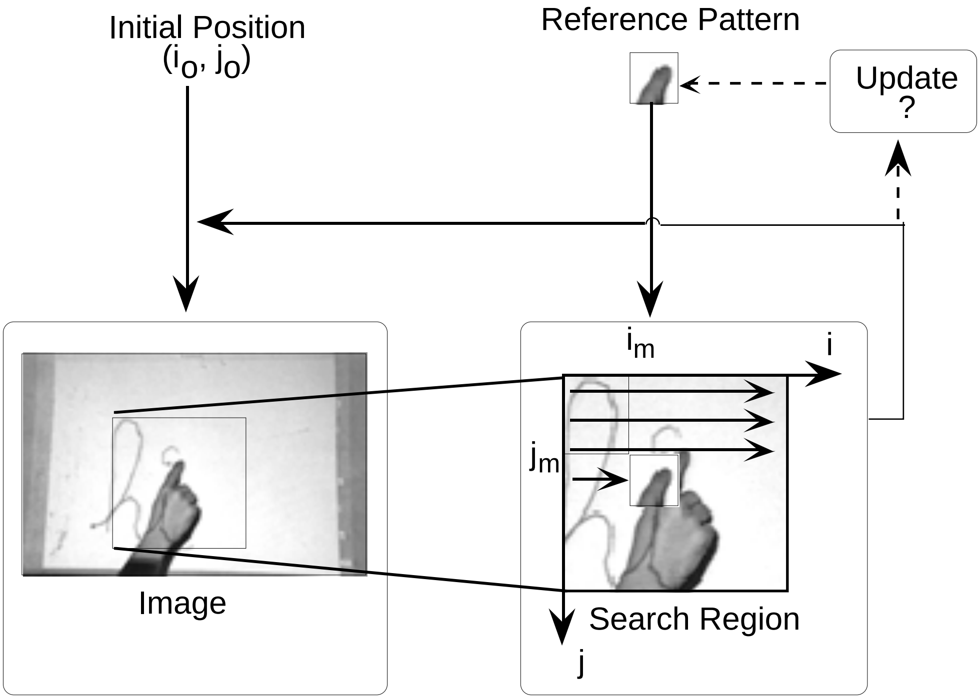

Figure 2. The components of a finger tracking system based on SSD correlation.

$$(i_m, j_m) = \underset{(i,j)}{\mathrm{Min}} \left\{ \sum_{m=0}^{N} \sum_{n=0}^{N} (p_k(i+m,j+n) - w(m, n))^2 \right\} \tag{1}$$

$$(i_m, j_m) = \underset{(i,j)}{\mathrm{Min}} \left\{ \sum_{m=0}^{N} \sum_{n=0}^{N} (p_k(i+m,j+n)^2 - 2\, p_k(i+m,j+n)\, w(m,n) + w(m, n)^2) \right\} \tag{2}$$

$$p(m,n) \otimes w(m,n) = \frac{\sum_{m=0}^{N} \sum_{n=0}^{N} p_k(i+m,j+n)\, w(m,n)}{\sqrt{\sum_{m=0}^{N} \sum_{n=0}^{N} p_k(i+m,j+n)^2 \sum_{m=0}^{N} \sum_{n=0}^{N} w(m,n)^2}} \tag{3}$$

Matching image neighbourhoods by sum of squared differences has come to be known in the vision community as SSD [Ana89]. This technique provides a simple and robust method for motion measurement and stereo correspondence matching. The relation between SSD and cross-correlation can be seen by rewriting the expression as show in equation 2. If the terms $p_k(i+m,j+n)$ and $w(m,n)$ are suitably normalised, then minimizing the sum of squared differences is equivalent to finding the position (i_m, j_m) which maximizes the inner product $<p_k(i+m,j+n), w(m,n)>$.

The summation terms $w(m,n)^2$ and $p_k(i+m,j+n)^2$ express the energy contained in the reference pattern and the neighbourhood beginning at (i, j). The signal processing literature contains many possible normalisation techniques [Asc92] for this energy. In a related project [Mar94] we compared a number of possible normalisation techniques. Experimental results showed that the most robust results were obtained by dividing the correlation by the energies of the neighbourhood and the reference signal, as shown by, as shown in equation 3. This robustness was most evident in scenes in which the ambient illumination varied. This form is in fact mathematically equivalent to SSD. For the finger tracking presented in this paper, we used the SSD formulation.

The results presented below were obtained using a personal computer (Apple Quadra AV 840) equipped with a built in frame grabber. In the last few years, a group at the University of Tokyo [Ino92], have used an M-PEG image coding chip to build a very simple video-rate cross-correlation device. A commercial version of this circuit has been announced at the time of writing of this paper.

Implementing correlation by SSD required solving practical problems concerning the size of the reference template, the size of the search region and when to when to initialise the reference template. These are described in the following sections.

2.2 The size of the reference mask

The size of the correlation template depends on the image size of the object to be tracked. If the template window is too large, correlation can be corrupted by the background. On the other hand, if the template covers only the interior of the pointing device, then the template will be relatively uniform, and a high correlation peak will be obtained with any uniform region of the image, including other parts of the pointing device. For a precision position estimate, the reference template size should be just large enough to include the boundary of the pointing device, which contains the information which is used for detection and localisation.

Our workspace is of size 40 cm by 32 cm. This surface is mapped onto an image of 192 x 144 pixels, giving pixel sizes of 2 mm by 2.2 mm. At this resolution a finger gives a correlation template of size 8 by 8 pixels or 16mm by 18mm, as shown in figure 3.

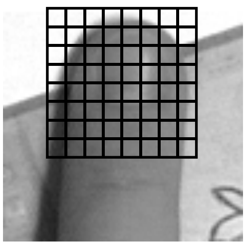

Figure 3 reference template for a finger.

2.3 The size of the search region.

Given an image processing cycle time of ΔT seconds per cycle, and a maximum pointer speed of V_m pixels/sec, the pointing device will be found within a radius of $M = \Delta T\ V_m$ pixels of its position in the previous frame. Fitts law [Car83] permits us to place an upper limit on the movement of the pointing device. However, this limit is based on assumptions which are best verified experimentally.

For images of 192 x 144 pixels, our built-in digitizer permits us to register images at a maximum frame rate of 24 frames per second, giving a cycle time of ΔT_{max} = 41.7 msec. This represents an upper limit on image acquisition speed which is attainable only if image tracking were to take no computation time. Considerations based on Fitts law indicated expected tracking speeds of up to 1800 mm/sec, or roughly 3600 pixels/sec. To verify this, we performed an experiment in which a finger was filmed making typical pointing movements in our workspace. The maximum speeds and accelerations observed in this experiment were V_m=1390 mm/sec or 695 pixels/sec with accelerations of A_m=17 660 mm/sec^2 or 8 830 pixels/sec^2.

The computational cost of correlation is directly proportional to the number of pixels in the search region. Thus reducing the size of the search region will decrease the time for each cycle. This, in turn, increases the number of times that correlation can be operated within a unit time, further decreasing the region over which the search must be performed. This positive feedback relation will continue until reaching a lower limit given by the image acquisition time. This relation can be expressed analytically.

An SSD correlation is composed of m sum-of-squared difference comparisons between the reference template and an image neighbourhood, one for each pixel of the (2M+1) by (2M+1) search region, such that $m = (2M+1)^2$. Each inner product costs n multiplies and n adds, where n is the number of pixels in the N by N reference template, such that $n = N^2$. Thus the cycle time for an SSD correlation is proportional to m and n, where k is the factor of proportionality (determined by the time to fetch, add and multiply pixels). In addition, there is a constant time delay, e, determined by the image acquisition time.

$$\Delta T = k\, m\, n + e = k\,(2M+1)^2 \cdot N^2 + e.$$

Computing an SSD correlation every ΔT seconds, permits a maximum speed of

$$V_m = \frac{M}{\Delta T} = \frac{M}{k \cdot (2M+1)^2 \cdot N^2 + e}$$

When $k \cdot (2M+1)^2 \cdot N^2 > e$, the relation is dominated by the time to compute a scan of the entire search region and the maximum speed is approximated by

$$V_m \approx \frac{1}{2kMN^2}$$

That is, there is an inverse relation between the width of the search region, M, and the maximum tracking speed, V_m. The smaller the search region, the faster the finger movement that can be tracked. On the other hand, when the search region is small, and $k \cdot (2M+1)^2 \cdot N^2 < e$, the maximum tracking speed will grow with the size of the search region. The two curves will meet at some value to produce a maximum displacement speed which can be measured by correlation. This is confirmed by experiments.

To determine the optimum trade-off between M and V_m, we systematically varied the size of the search region from M = 10 to 46 pixels and measured the cycle time that was obtained. Figure 4 shows the maximum displacement speed V_m in pixels/sec plotted for different size search regions. The maximum speed of 126 pixels/sec (252 mm/sec) is obtained with M=13 pixels.

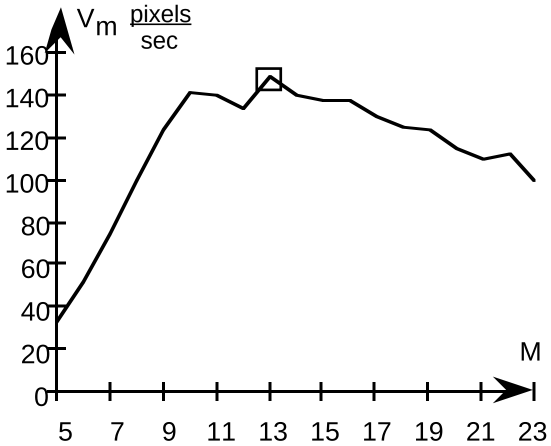

Figure 4 Maximum speed of trackable movement V_m as a function of the search region width.

2.4 Triggering and breaking tracking

When tracking is not active, the system monitors an 8 by 8 pixel "tracking trigger", $T_k(i,j)$, located in the lower right corner of the workspace. As each image is acquired at time k, the contents of this tracking trigger are subtracted from the from the contents at the previous image, k-1. This creates a difference image as shown in figure 5. The energy of the different image is computed as

$$E_k = \sum_{m=0}^{7} \sum_{n=0}^{7} (T_k(m,n) - T_{k-1}(m,n))^2$$

When a pointing device enters the tracking trigger, the energy rises above a threshold. In order to assure that the tracking device is adequately positioned, the system waits until the difference energy drops back below the threshold before acquiring the reference template. At that point, the contents of the tracking trigger, $T_k(m, n)$ are saved as a reference image, and the tracking process is initiated.

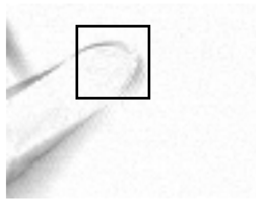

Figure 5 Temporal different of images in the reference square.

Tracking continues as long as the minimum value of SSD remains below a relatively high threshold. However, it can happen that the tracker locks on to a pattern on the digital desk (for example a photo of the pointing device!). To cover this eventuality, if the tracked location of the pointer stops moving for more than a few seconds (say 10), the system begins again to observe the difference energy in the tracking trigger. If the trigger energy rises above threshold, the tracker will break break the previous track and re-initialise the reference pattern with the new contents of the tracking trigger.

2.5 Updating the reference mask

As the user moves the pointing device around the workspace, there is a natural tendency for the device to rotate, as shown in figure 6. This, in turn, will decrease the minimum SSD and can even cause loss of tracking. In order to avoid loss of tracking, the value of each SSD is compared to threshold. If the value rises above a relatively high threshold, then the reference template is updated using the contents of the image at time k-1 at the detected position.

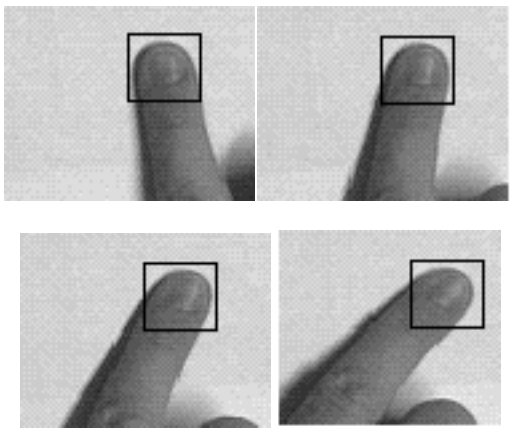

Figure 6 Change in reference template as a function of finger orientation.

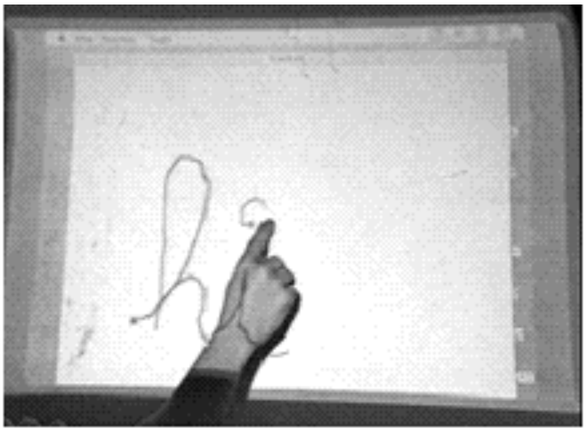

Figure 7. Drawing with "FingerPaint".

3 FingerPaint: a simple demonstration

As a simple demonstration, our finger tracking system was used to build a demonstration program called "FingerPaint". FingerPaint uses a work-space projected with an overhead projector using a liquid-crystal display "data-show". A CCD camera with an 18mm lens observes this workspace and provides visual input. MouseDown detection was simulated using the space bar of the keyboard. The user can use tracking to draw pictures, as shown in figure 7.

SSD correlation provides a simple means to track pointing devices for a digital desk. The widespread availability image acquisition and processing hardware adequate for real time correlation makes it very likely that real-world physical world objects will come to replace the mouse and keyboard as the communication device in computer systems.

Bibliography

[Ana87] P. Anandan. *"Measuring Visual Motion From Image Sequence"*. Phd dissertation and COINS Technical Report 87-21, University of Massachusetts, Amherst, 1987.

[Asc92] P. Aschwanden and W. Guggenbühl. *"Experimental Results from a Comparative Study on Correlation-Type Registration Algorithms"*. in "Robust Computer Vision" pp. 268-289, Förstner and Ruwiedel, Wichmann Publisher, 1992.

[Asc88] P. Aschwanden, *"Real-time Tracker with Signal Processor"*. Signal Processing IV : Theories and Applications. Elsevier Science Publishers, 1988.

[Bal93] S. Balbo, J. Coutaz and D. Salber. *"Towards Automatic Evaluation of Multimodal User Interfaces"*. International Workshop on Intelligent User Interfaces, Orlando, USA, Jan., 1993.

[Bla94] A. Blake and M. Isard, "*3D Position, attitude and shape input using video tracking of hands and lips*", ACM - SIGGRAPH Annual Conference on Computer Graphics, 1994.

[Ber94] F. Berard, "*Vision par Ordinateur pour la Réalité Augmenté: Application au Bureau Numrérique*", Mémoire du D.E.A. en Informatique, Univeristé Joseph Fourier, Juin 94.

[Cou90] J. Coutaz, "*Interfaces hommes-ordinateur Conception et réalisation.*" Dunod Informatique, 1994.

[Cro89] J. L. Crowley, P. Stelmaszyk and C. Discours, "*Measuring Image Flow by Tracking Edge Lines*", ICCV-2, Tasrpon Stpings, Fla, Dec 1989.

[Cro93] J. L. Crowley, P. Bobet and C. Schmid , "*Auto-Calibration of Cameras by Direct Observation of Objects*", Image and Vision Computing, Vol 11, no. 2, March 1993.

[Cro94] J. L. Crowley and H. Christensen, Vision as Process, Springer Verlag Basic Reserch Series, Heidelberg, 1994.

[Har92] C. Harris, *"Tracking with Rigid Models"*. in "Active Vision", The MIT Press, 1992.

[Ino92] H. Inoue, T. Tashikawa and M. Inaba, "*Robot vision sytem with a correlation chip for real time tracking, optical flow, and depth map generation*", 1992 IEEE Conference on Robotics and Automation, Nice, April 1992.

[Kas87] M. Kass, A. Witkin and D. Terzopoulos. "*Snakes : Active Contour Models*". Proc. 1st International Conf. on Computer Vision, pp. 259-268, 1987.

[Mae94] P. Maes, T. Darrel, B. Blumberg and S. Pentland. *"The ALIVE system : Full-Body Interaction with Animated Autonomous Agent"*. M.I.T. Media Laboratory Perceptual Computing Technical Report No. 257, Jan. 1994.

[Mor80] H. P. Moravec. *"Obstacle Avoidance and Navigation in the Real World by a Seing Robot Rover"*. Phd Thesis, Stanford University, 1980.

[New92] W. Newman, P. Wellner. *"A Desk Supporting Computer-based Interaction with Paper Documents"*. Proceedings CHI'92, pp. 587-592.

[Reh93] J. M. Rehg and T, Kanade. *"DigitEyes : Vision-Based Human Hand Tracking"*. Carnegie Mellon University Technical Report CMU-CS-93-220, December 1993.

[Wel93a] P. Wellner, Wendy Mackay and Rich Gold. *"Computer-augmented environments : back to the real world"*. Special Issue of Communications of the ACM, Vol.36 No.7, July 1993.

[Wel93b] P. Wellner, *"Interacting with paper on the DigitalDesk"*. Communications of the ACM, Vol.36 No.7, July 1993.

[Woz65] Wozencraft J. M. and Jacobs I. M. Principles of Communication Engineering, John Wiley and Sons, 1965