The Registar$^{(R)}$ Machine: from Conception to Installation

G. Medioni and A. Huertas

Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, California 90089

M. Wilson

Opty-Copy, Inc.
A Division of Polychrome, Inc.
Lenexa, Kansas 66219

Abstract

We have developed a machine to perform the task of automatic registration of color separation films, a process manually performed by skilled professionals in the graphics arts printing industry. The development of such a machine requires overcoming significant challenges: designing a sound computer vision methodology while respecting hard timing constraints, transferring software across platforms and languages, validating the software, building the actual machine around the algorithms, testing the conformity to tolerances, educating operators on the use of such a machine, and having a system robust enough to operate around the clock with no technical supervision. We present a brief overview of the problem, followed by the answers we provided to the challenges above.

1 Introduction

This paper presents a case study in the transfer of machine vision technology from the laboratory to a small company, going from proof of concept to commercial distribution.

We first provide a brief overview of the problem to be solved, giving also some of the requirements for the task at hand. We then discuss the approach taken to solve the problem, which includes the choice of primitives, their implementation, and the matching algorithms. This technical part is given in detail in [6].

We then discuss some of the innovations and short cuts which were required in order to meet the running time constraints.

We also briefly mention the issues involved transferring software from Lisp on a Symbolics Lisp Machine to Fortran on a IBM-compatible personal computer (and machine code on the Mercury array processor).

Finally, we present the challenges posed by having a machine run by itself on hundreds of films, unattended, and operated by a user who looks at it as a black box. These include: lighting and camera focus, calibration between table and camera coordinates, design of punch mechanisms, self-test diagnostics, testing (in house and beta-test sites), and feedback to the operator (display, measure of quality).

2 Context of the Problem

In order to print color photographs such as those which appear in mass circulation magazines or other publications where quality is of utmost importance, the original color picture is broken down into four (or more) separate photographic images which are processed independently. Each one is a binary picture in which the appearance of continuous tones is obtained by using dots of varying sizes. It is crucial for the halftone color separations to be precisely (within 25μ (1 mil) or better) aligned so that the printed picture faithfully reproduces the original.

Currently, this registration is for the most part performed manually by highly trained personnel referred to in the printing industry as "strippers". The manual process involves taping the separation films to a larger, clear polyester film called a "carrier sheet". The first (reference) film is taped to a prepunched carrier sheet; the remaining films are visually aligned to the reference and taped onto their own prepunched carrier sheet in their registered position. Typically, this registration is carried out by aligning one detail of the images under a magnifying loupe (6x-20x), then another detail distant from the first (in order to correct for both rotation and translation).

While these trained personnel perform an outstanding job most of the time, it is important to bear in mind that an error may cause presses to be stopped, an extremely expensive operation.

Human errors usually come from fatigue and lead to inaccuracies. Consistency and reliability are hard to maintain as skill and vision vary from one stripper to the next. Also, manual registration is slow, taking 10 to 20 minutes to perform per set.

Other attempts at automation either rely on register marks (which may disappear during film trimming), or use correlation techniques [11] therefore limiting the applicability and precluding commercial viability.

From the observations above, it results that the requirements for a successful machine should include:

1. A *speed* of about ten sets per hour, including all human intervention (rough register, taping to carrier sheet, loading and unloading sheets and verification) and all processing steps (image acquisition and processing, matching, mechanical motion and hole punching).

2. The *accuracy* needed is 1 mil (25 μ) for true halftones and 0.5 mil (12 μ) for letters or register marks.

3. The machine should be *consistent* with a human stripper, both for false alarm (refusing to punch a film) or non detection (unacceptable for printing).

This makes for a challenging computer vision problem, for two reasons:

1. The separation images are actually different: The images are halftone, which means that from one color separation to the next, the dots are not overlapping because the rectangular grids are rotated between colors (to avoid moiré effects). In fact, corresponding dots in registration form a rosette pattern. This disqualifies pixel correlation.

These limitations are compounded by the fact that the information encoded by different colors, although globally similar, may be locally very different. An area can be dark in one color separation and light in another; text (which is easier to match) generally does not occur in all colors. It also may happen for the same pattern in different color separations to appear with different sizes (choke and spread effect).

2. Speed and accuracy: The accuracy needed is 1 mil, to be held on the entire negative (typically 8 $\times$ 10 in^2), but we cannot capture (much less process!) the entire image with adequate resolution.

3 Algorithmic Approach

In order to register any two color separations, we use a strategy analogous to the one used by human strippers. We pick two small detail areas on the reference film and register each of these two windows independently. (The choice of these detail areas is made by the operator using a cursor).

In figure 1 we show a flowchart with the sequence of steps in our system. In figure 2 we show the cyan and magenta component images used to illustrate our discussion throughout.

3.1 Feature Extraction

The halftone images differ locally at the pixel level, however, they contain very similar salient components at the macro level. The key is then to extract features representing these salient components.

We detect macro edges obtained by convolving the images with very large radially isotropic Laplacian-of-Gaussian (LoG) filters: $\nabla^2(r) = \frac{1}{2}\left(2 - \frac{r^2}{\sigma^2}\right) e^{-\frac{r^2}{2\sigma^2}}$ where σ is the space constant of the Gaussian, and $r = \sqrt{x^2 + y^2}$ is the radius of the operator.

We have chosen these filters because their properties are well understood in the literature [5, 9], and because the choice of the parameter σ associated with them has a natural interpretation in our case: it controls the amount of blurring desired, which depends on the known halftone screen resolution. The chosen value for $\sigma = 16.3$ in our implementation leads to a mask slightly larger than 100×100. We note in addition that an isotropic operator, as opposed to directional operators, requires only one convolution.

A salient feature, such as the contours of the doll's eye in the EYE images (figure 3), is actually the boundary between two regions of halftone dots of different densities (dark and light). The individual dots of each region fuse into a uniform region, thus preserving the boundary between them.

To extract the contours from the image we perform edgel detection and edgel linking. We used an efficient and accurate method to link edges obtained by finding the zero-crossings of the images convolved with LoG masks. Additional details can be found in [3, 4, 6]

The zero-crossings detected in our EYE example are shown in figures 4a and 4b. We compute the location of the step edges with subpixel accuracy. We also estimate the height d, and the slope difference Δk.

The RegiStar$^{(R)}$ uses the estimated height d of the edge to threshold the output of the operator. We apply a double thresholding mechanism similar to the hysteresis thresholding in [1].

The low threshold is obtained during calibration by processing a transparent film, for which all the obtained zero-crossings are spurious. The high threshold is obtained by statistical analysis of a large number of films.

This thresholding reduces the processing time by reducing the number of primitives.

We link and fit line segments to the zero-crossing contours in a manner similar to the technique described in [7]. The edge map is scanned to collect predecessor and successor information for each edge point using the orientation of each zero-crossing. The orientation is coded into 8 directions (45° intervals).

In the RegiStar$^{(R)}$ a fitting tolerance of 0.75 pixels is used. The fitting procedure involves first creating "2-segments" (segments connecting two neighboring edges) and then combining the 2-segments into longer segments until the fitting tolerance is exceeded. The result is a list of segment end point coordinates. The line segments in our EYE examples are shown in figures 5a and 5b.

3.2 Matching

3.2.1 Matching one window

Matching the features (line segments) computed from the images involves two major steps: Estimate the translation and estimate the rotation. We first obtain an estimate of translation (or misregistration) by independently matching two areas of the film, and then use these results to estimate rotation.

To estimate translation we first determine pairs of matching segments. Each pair of matching segments "votes" for a two-dimensional area in *translation* space. This accumulation mechanism in parameter space is a Hough-like transform [8] adapted to line segments. The amount of the contribution varies as a function of the lengths of the overlap between the contributing segments.

The resulting space should contain a peak, whose position gives the estimate of translation. Choke and spread features on half tone images may result in crater-like peaks. The final estimate of translation is then obtained by a least squares technique.

A 3-D plot of the raw accumulator array corresponding to the match between cyan and magenta is shown in figure 6. We smooth the array values with a Gaussian ($\sigma = 1.5$), and locate the two largest peaks in it. We require that the magnitude of the secondary peak be no larger than 80% of the magnitude of the largest peak to declare the match successful. The location of the main peak in parameter space gives the estimate of misregistration. The main peak in our EYE example is at $(+10, -15)$ and has a magnitude of 122.1. The second largest peak is at $(-27, +27)$ and has a magnitude of 47.7.

In figure 7a we show an overlay of the segments in their original position, and in figure 7b an overlay of the same segments after applying the predicted offsets.

3.2.2 Matching two windows

With the translation estimate for the two windows, it is now possible to compute the parameters for the full rigid transform relating the two films: Let $\vec{u}$ be the vector joining the centers of the two windows in the reference film, and $\vec{v}$ the corresponding one in the other film. The rigid transformation F which maximizes the overlap between $\vec{u}$ and $F(\vec{v})$ is the desired answer. In the ideal case, $\vec{u}$ and $\vec{v}$ should have the same length, so this condition provides a verification step in our algorithm: if the length difference is larger than an acceptable threshold (we use 75μ or 3 mils), then the matching procedure is aborted. There are multiple reasons for this to occur: the matching of individual windows may be wrong, or the physical film may have stretched in one direction, or the picture is a composite.

As a result of the segment based matching described above, we have a pair of offsets for two windows on the reference and test films. Using this information, we compute the parameters of the best rigid transform which should be applied to the test film in order to punch registration holes. The mathematical details are given in [6].

4 Additional Constraints

As the collaboration between the USC team and the Opti-Copy group got under way, it became clear that the "proof-of-concept" measure of success ("it worked once on one image!") often used in the computer vision community, was hardly sufficient in the eyes of the company.

We give three specific instances of such problems, and the way we solved them.

4.1 Convolution with a large LoG Mask

At USC, we considered the feature extraction "solved" when we showed that convolving the image with a LoG mask with a σ of 16.3 gave the desired result. This, unfortunately, leads to a filter size over 100×100, and a direct convolution, even with the decomposed version of the mask [4], takes hours on a VAX 750 or similar machine, an unacceptable run time for any viable implementation.

By being forced to confront this issue, we came up with an innovative and general solution, namely decimated convolution [2]:

Briefly, we take advantage of the spectral properties of the Gaussian and LoG filters: The LoG is a band pass filter, we can therefore fold the spectrum of the image (after low pass filtering) without loss of information, which is equivalent to reducing the resolution.

Consider g, a Gaussian function with space constant σ_g:

$g(r) = e^{-\frac{r^2}{2\sigma_g}}$ and $G(R) = Ce^{-\frac{R^2\sigma_g^2}{2}}$ its Fourier transform.

The LoG can be rewritten as:

$LoG_\sigma = G_{\sigma_1}(R) \cdot LoG_{\sigma_2}(R)$ where,

$\sigma_1 = \sqrt{1 - \frac{1}{k_\sigma^2}} \cdot \sigma$ and $\sigma_2 = \frac{\sigma}{k_\sigma}$

That is, the convolution operation is equivalent to a decimated Gaussian convolution followed by a LoG convolution with a very small mask. In our implementation the 640×640 convolution results in 80×80 output using a decimation factor of 8, taking 2.1s. This corresponds to approximately a speed up of 50 with respect to convolutions with separable LoGs. For details see [2].

4.2 Expansion to Original Resolution

The zero crossing detected on the smaller images (80×80) have a precision much too coarse to perform an accurate correspondence, so they need to be expanded back to their original resolution.

We had proposed what we thought was a "solution" in [4], by fitting a polynomial to a zero crossing pixel and its eight neighbors, then performing zero crossing detection again on a finer grid. This "solution" was deemed unacceptable because it is too slow, and it creates too many edgels for the next stage, linking.

The chosen compromise was to indeed expand the zero-crossings to their original (or higher) resolution using bilinear or spline interpolation of the decimated convolution. (We use a 17×17 window centered at each zero-crossing.)

Moreover, we only expand each edge pixel at low resolution to one pixel at high resolution, by choosing the value closest to zero in the expanded window. This guarantees that the chosen pixel belongs to the contour formed by expanded edges. Note that this procedure is equivalent to making a linear approximation between expanded edgels. Since we perform a linear approximation afterwards to obtain line segments, the loss of resolution here is not critical.

4.3 More Edge Accuracy Problems

We reached a stage of development where the registration with real images for crosshairs gave acceptable accuracy consistently, but was too sloppy (2 mils) for actual halftone images.

Tracking the source of the misregistration proved quite difficult, as it appeared on certain films only.

We finally attributed it to the different profiles an edge can present in different images leading to a different bias in the location of corresponding edges.

We therefore analytically studied the effects of the edge profile on bias, and proposed a method to correct it, as described in [10].

We briefly summarize the results:

Consider a 1-D step edge. Let d be the height of the edge and k_1 and k_2 the slope of the profiles on both sides of the step:

$$I(x) = \begin{cases} k_1(x-t) & if\ x \leq t \\ k_2(x-t)+d & otherwise \end{cases}$$

The result of the convolution with a LoG operator is:

$(I * G'')(x) = [\frac{d(x-t)}{\sigma^2} - \Delta k]G(x-t)$

where $\Delta k = (k_2 - k_1)$ and $G(x)$ is the Gaussian function defined as (neglecting the normalizing constant $\frac{1}{\sqrt{2\pi}\sigma}$): $G(x) = e^{\frac{-x^2}{2\sigma^2}}$

The zero-crossing of $(I * G'')(x)$ above is located at $x_z = t + [\frac{\Delta k}{d}]\sigma^2$

Therefore the bias is $x_z - t = [\frac{\Delta k}{d}]\sigma^2$

Next we determine t to be (see [10] for details):

$t = x_z + \frac{\sigma^2}{2l} log\left|\frac{m_1}{m_2}\right|$

where m_1 and m_2 are samples of the convolution output, at a distance l on both sides of the zero-crossing. In the RegiStar$^{(R)}$ implementation we use $l = \sigma$ with very good results.

4.4 Software Transfer

The software at USC was written mostly in Lisp by Andres Huertas on a Symbolics Lisp Machine, whereas Monti Wilson wrote in Fortran. The excellent editing, graphics, debugging and overall software environment on the Lisp machine allowed for rapid programming and testing of the various techniques.

Early attempts at exchanging code proved to be the wrong approach (who wants to debug someone else's code in a different language?). Rather, the emphasis was in delivering regularly clear descriptions of the algorithms and techniques, well supported by a comprehensive set of test results. On the Symbolics, the entire process was developed as modules of a software system that could easily be studied and tested separately and/or together.

We believe that it is not realistic to supply code from an academic laboratory to an industrial one and expect it to behave like a commercial system, even if no language differences exist. The role of the academic lab was to come up with, and to demonstrate, theoretically sound techniques. The role of the industrial lab was to extensively test these on many examples, adjust the parameters, and to eventually point out

Process	time
Convolution	2.1s
Edge detection	0.5s
Edge linking	0.8s
Interpolation	2.6s
Segment fit	0.9s
PSM matching	3.3s
Least squares	1.0s
Total	11.2s

Table 1: Processing times

deficiencies. Small differences in results were due to different accuracy and rounding effects.

5 The Machine around the Software

The actual machine [1], shown in figure 8, carries out all mechanical motions required, captures the images, punches the registration holes and provides an interface to the user. It performs the registration of a four color set of films in about 5 minutes, including all operations. This corresponds to a cycle time of 12s. to match two windows, broken down as shown in table 1.

This speed is achieved by using a Mercury array processor to perform the vector operations. The code is written in Fortran, with some critical inner loops microcoded for efficiency.

The image acquisition device is a line scan CCD camera with a resolution of 640 square pixels. A stepping motor controlled by a microprocessor allows it to scan the area in order to produce a 640×640 image.

The windows are selected by an operator with a cursor whose position is read on a digitizing tablet.

The films to be processed are affixed to a backlit transparent plate and held by vacuum. The plate is part of a chase carried on an X-Y table with independent linear motions operated by high precision drive screws. The repeatability is better than $\simeq 6\mu$ (0.25 mils.)

Finally, punching of registration holes on the film is performed by a mechanism including a pneumatic cylinder.

We now describe issues involving the overall machine rather than the algorithm itself, and the testing procedure to validate the machine.

[1] U.S. patent 4,849,914 issued to Medioni *et al.* in July 1989

Lighting and Camera Focus: The light source for RegiStar$^{(R)}$ is a 24 volt dc projection lamp with a rheostat control. The light projects about a 2 inch diameter spot on a diffuser plate between light and lens. The 55mm Micro Nikkor lens is operated at f/5.6. With the light near full brightness, the camera is vernier focused by maximizing a computed edge sharpness gradient. The light uniformity is adjusted by inspection of a 2D gray level map over 64 image blocks, until computed uniformity is optimized. Small changes in the light intensity can be made to optimize computed halftone tint values (percentage of black pixels to total pixels) to agree with known (via densitometer) tints. A delay is present in software so that the lamp has been on for a least 2 seconds before an exposure is made.

Recently, both the Nikkor 55mm lens and the DataCopy digital camera were discontinued. The camera had 13μ square pixels and utilized a line scan array. We windowed a 640×640 pixel image out of 1024×1024. The 55mm lens was (accurately) operated at 1:1 with no calibration required. The new camera is the Videk Megaplus coupled with the new 60mm Micro Nikkor lens. The Videk camera is a 2D CCD array device with 6.8μ square pixels. To avoid quadrupling the throughput time, we decided to keep the same number of pixels over the same image dimension as before. This requires optical reduction of the original image onto the CCD. The lens is set to nominal reduction ratio $R = 1.9$ and the precise value is determined by counting pixels over a precision target width of 125 mil $\pm$ 0.1 mil. The calibrated value is closer to 1.8. The only change to the algorithms is that the LOG filter size is scaled by factor $6.8 * \frac{R}{13}$, since the effective pixel size over the unreduced image dimension is $R * 6.8\mu$, still close to the old 13μ figure. The loss in view field is quite small. Using the original light source at full brightness for proper color temperature and uniformity, the new system operates at f/8 using a neutral density filter of optical density 2 (the Videk camera is quite sensitive, and operates best at 50 ms exposure time with gain of -6db). It was required for the Videk system that we design a frame grabber which streams digital pixels through windowing hardware into RAM. The RAM can then be interrogated for both PC display and data transfer into the array processor memories.

Calibration: Proper machine calibration is crucial to the registration accuracy. The primary calibration concerns the coefficients of the translation and rotation transformation between the digitizer tablet and its image coordinate system on the chase. The procedure is based on two special targets: 1) a crosshair bearing known dimensions to the rough register pin system and 2) a grid of parallel dots. The crosshair is centered in the image grid by successive centroid computations plus table moves. This fixes the coarse translation between digitizer and chase. The dot grid target is parallel to the line of pins and allows the rotation component of the calibration to be determined, as well as fine translation adjustments, again through dot centroid computations. A well-calibrated machine will display an image centered to within $5 - 10\frac{1}{2}$ mil pixels. A 40 pixel error, acting through rotation would begin to cause unacceptable additional errors in the punched registration on a random basis. The entire calibration has been reduced to two simple, automatic procedures.

Punch Mechanism: In the early stages of RegiStar$^{(R)}$, it was thought that proper punch alignment of punch to die was a difficult problem. For this reason, soft metal punches and dies were used on the theory that they would "peen" into alignment. Alas, the soft metal punching was a failure. The punches frequently stuck and wore out too quickly. Electric punches were tried, but frequently did not have enough "oomph" to punch properly. Ultimately, we settled on house compressed air punching. A radically different approach yielded the current highly satisfactory punching system. The punches are hard steel with a 0.2 mil radius to the cutting edge. They punch into die buttons that are "floated" into alignment by virtue of an adhesive matrix that allows precise positioning and subsequent hardening into fixed locations. This system was first verified on a test stand that punched more than a million punch holes on long rolls of tape before it was turned off. The final holes were still accurate. Typically, punching will contribute less than or equal to 0.1 mil to misregister, although the conservative figure is $\frac{1}{4}$ mil ($\frac{1}{4}$ punch error plus $\frac{1}{4}$ algorithm error = tolerable worst case machine error of $\frac{1}{2}$ mil).

Self-diagnostic: Diagnostics are used to verify proper machine function and to troubleshoot machine failures. The key diagnostic is the self-test. The simplest self-test registers the same set of marks to itself and records the registration offsets for many registered films (we look for mostly zero offsets with a small scattering of random $\frac{1}{4}$ mil and even smaller scattering of random $\frac{1}{2}$ mil errors). The punches are turned off during self test and the repeated film is always under vacuum. Varieties of self-test are 1) the zero offset test, 2) the known input random offset test, 3)the stationary self-test at a fixed point and 4) the one-dimensional movement self-test. Test 3) is useful for eliminating the motion table as a variable. Test 4) allows one-dimensional table repeatability to be tested. Test 2) is the primary test, since it simulates out of register (random offsets) images put back into register by the computed registration offset. Another useful diagnostic is the display of images pulled out of array processor memory. This verifies the image integrity after image capture and transfer into memory.

Testing: Testing for Registar$^{(R)}$ is divided into two categories: 1)measured marks and 2) halftones. The measured marks are simple crosshairs 7 mil thick and about $\frac{1}{4}in^2$ located diagonally at the corners of each page size to be tested. In one quadrant of each mark is a reference black dot 1 mil in diameter. Mark sets are registered on the machine and punched. Register is quantified by the movement of the reference dot as focus varies from bottom film to top film of the two films laid up on register pins. Each selected mark dot is noted visually with a 100x microscope as a displacement estimated to be either $0, \pm\frac{1}{4}, \pm\frac{1}{2}, \pm\frac{3}{4}$ or greater in mils as the focus shifts between top and bottom dots of the overlaid images. Statistics are run for at least 80 registered films, and the 3σ figure of merit is derived. A machine passes if no measured error exceeded $\frac{1}{2}$ mil and the 3σ merit figure is reasonably close to $\frac{1}{2}$ mil. Halftone register is checked visually (somewhat subjectively) by the trained eye of a stripping professional aided by about a 6-20x loupe (6x is most common). Perhaps 10 or so different halftones are registered to verify any changes made in the software. In the early days of development, before the algorithms stabilized, about 150 different halftones were used in the test suite. These halftones varied in screen ruling from 120 to 200 lines/inch. Recently it was verified that the simple scaling of the LOG filter size by the factor 150/RULING extends to at least 300 line screen. Only in recent years has interest in high screen rulings revived due to new plate materials that can now hold the quality improvement. Once machines were established in the field, proper machine function was confirmed at the rate of about 300 films per 8 hour shift. Beta testing in the field basically took over the role of extensive in-house testing with the large film suite. Another aspect of RegiStar$^{(R)}$ that involved running about 100 different halftone pictures was the empirical selection of pass-fail thresholds for the heuristic algorithm used to assess match quality. This algorithm has about a 95% success rate in identifying problem images that failed to register closely enough. The technique used was to minimize false labeling of pictures as "bad" when the offset analysis said otherwise.

Measure of Reliability and Interface: In the presence of a registration error or a physical imperfection on the films (the most common occurrence is some amount of directional stretching occurring on a graphics art scanner drum), the lengths of the two vectors associated with the two windows differ by $2\Delta d$. Given our tight registration tolerances, this error becomes unacceptable when $\Delta d \geq 37\mu$ (1.5 mils). In such situations, the registration procedure is aborted, and it is up to the operator to reject the set of films, or to try to register the films again with a different selection of windows.

6 Conclusion

We have presented the issues involved in building a complete system to perform the automatic registration of color separation films. Besides the algorithmic development a significant amount of effort was also devoted to the mechanical part of the machine in order to calibrate the optical apparatus, move the images in x and y with 6μ accuracy or better, and reliably punch clean round holes. The code was written in Fortran with some critical portions microcoded for speed. The machine is now installed at many sites both in the United States and in Europe, and performs the registration faster and more reliably than human operators.

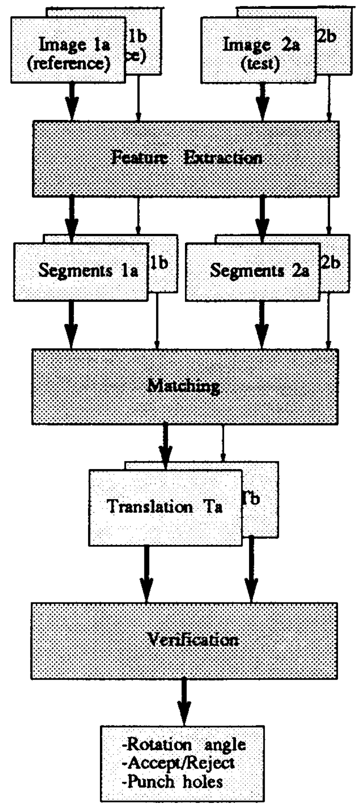

Figure 1: Major steps in Registration process

References

[1] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[2] J.-S. Chen, G. Medioni, and A. Huertas. Fast convolution with Laplacian-of-Gaussian masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):584–590, July 1987.

[3] S. Gazit and G. Medioni. Accurate detection and linking of zero-crossings. Technical Report USC-IRIS 252, University of Southern California, Loas Angeles, California, April 1989.

[4] A. Huertas and G. Medioni. Detection of intensity changes with subpixel accuracy using Laplacian-of-Gaussian masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):651–664, September 1986.

[5] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London*, B(207):187–217, 1980.

[6] G. Medioni, A. Huertas, and M. Wilson. Automatic registration of color separation films. *International Journal of Machine Vision and Applications*, 4:33–51, April 1991.

[7] R. Nevatia and K. R. Babu. Linear feature extraction and description. *Computer Graphics and Image Processing*, 13(3):257–269, July 1980.

[8] Hough P.V.C. *Method and Means for Recognizing Complex Patterns.* U.S. Patent 3069654, December 18, 1962.

[9] V. Torre and T. A. Poggio. On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):147–163, March 1986.

[10] F. Ulupınar and G. Medioni. Refining edges detected by a LoG operator. *Computer Vision, Graphics, and Image Processing*, 51(2):275–298, August 1990.

[11] M. Wilson. Machine vision in the graphic arts: Digital registration of four color halftone separations. In *Proceedings of the Technical Association of the Graphics Arts*, pages 525–536, 1986.

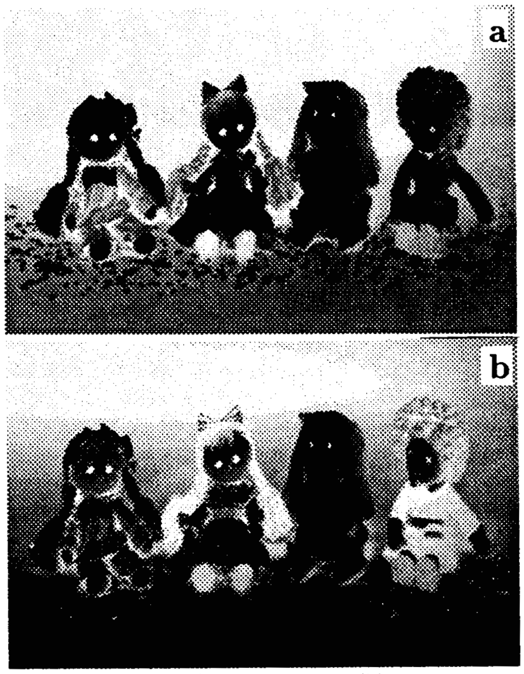

Figure 2: Cyan (a) and magenta (b) components

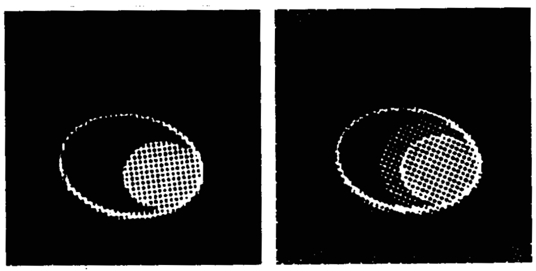

Figure 3: 640x640 details. Cyan (a), magenta (b)

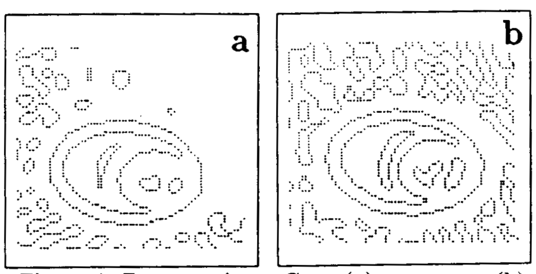

Figure 4: Zero-crossings. Cyan (a), magenata (b)

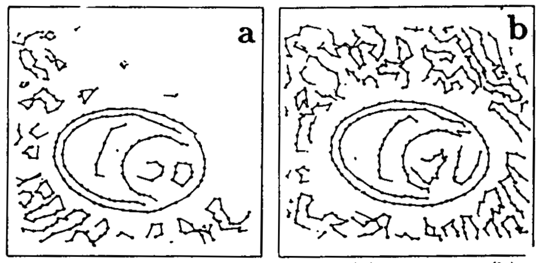

Figure 5: Line segments. Cyan (a), magenta (b)

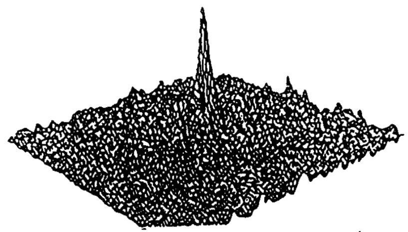

Figure 6: Plot of accumulator array

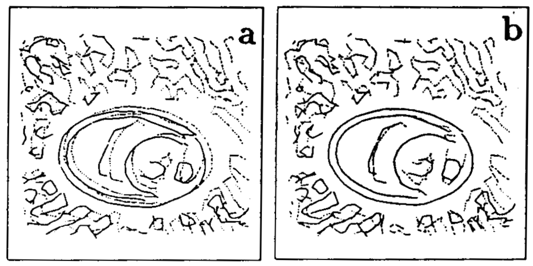

Figure 7: Original placement of segments (a), and placement of segments after applying estimated translation (b)

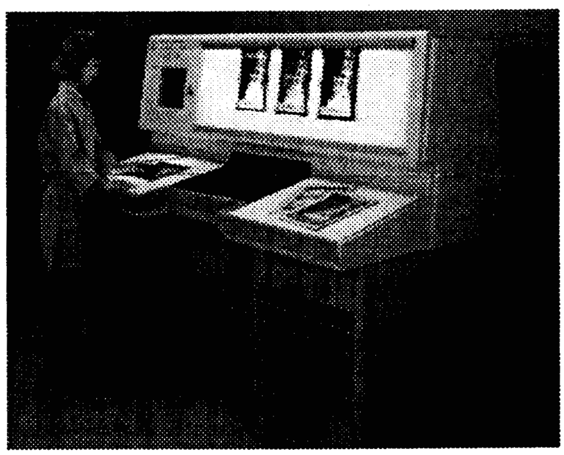

Figure 8: RegiStar$^{(R)}$ machine