MVA '96 IAPR Workshop on Machine Vision Applications, November. 12-14, 1996, Tokyo, Japan

Using Computer Vision in Real Applications: Two Success Stories

Gérard Medioni
Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA 90089-0273, USA
medioni@usc.edu

Abstract

We present two systems which are used today in industrial applications. While they are very different, and the two domains have no overlap, they share the properties that the problems they address were considered very hard to solve, and the requirements were very constraining, especially complete automation. We first present a system which automatically registers two sets of halftone color separations used to produce color pictures. The challenges involve accuracy, speed of processing, and consistency with human operators. The second system substitutes, in real-time, a given billboard in a video stream, with another, synthetically generated billboard. The challenges involve real-time performance and photo-realism. For each of the two systems, we provide some background, describe the requirements and the issues, then the implemented solution.

1 Registration of Color Separations

1.1 Background

In order to print color photographs such as those which appear in mass circulation magazines, or in other publications where quality is important, the original color image is broken down into four (or more) separate photographic images, which are then processed independently. Each one is a **halftone** picture, in which the appearance of continuous tones is obtained by using dots of varying sizes. Each color separation carries black and white tone information which reflects the corresponding intensity contents of the original color picture. It is crucial for the halftone color separations to be very accurately (within 25μ or better) registered, so that the printed image 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 pre-punched carrier sheet; the following films are visually aligned to the reference and taped onto their own pre-punched carrier sheet in their registered position.This manual registration suffers from some shortcomings:

- *accuracy:* strippers need to consistently register separation films within a fraction of the inter-dot spacing, and this is difficult to maintain, even for experienced strippers. In high quality printing, there are at least 5.9 dots per *mm* (150 dots per *inch*), so the center to center distance is less than 0.19mm (7 mils). With such tight constraints, human errors inevitably occur. Furthermore, if an unacceptable registration is not discovered until press time, the presses have to be held up, an inordinately expensive event.
- *cost:* manual registration is a tedious, slow, and highly labor intensive process. It typically takes a highly skilled (and therefore well paid) stripper between ten and twenty minutes to register four color separations. Consistency and reliability are hard to maintain as skill and vision vary from one stripper to the next, and each stripper is subject to fatigue.
- *magnification aids* have been proposed to increase accuracy. The magnified image of a small portion of a negative often appears as a rather random collection of dots, which do not correlate from one separation to the next. This problem is aggravated by the difficulty in accomplishing fine eye and hand controlled motion when viewing a magnified image. Magnification aids are therefore mostly used for the verification of registration.

1.2 Requirements and Challenges

The previous discussion sets the stage for the requirements of a successful vision-based system to perform the color registration automatically:

- **Speed**: the system should be at least as fast as a human stripper and therefore process about 10 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).
- **Accuracy**: the machine should consistently produce registration results of less than 25μ (1 mil) for halftone details and of less than 12μ (0.5 mils) for reverse type (letters or register marks).
- **Reliability**: the overall system, including mechanical parts, must be reliable because of the very tight deadlines imposed by magazines. (Overnight runs are more the norm than the exception.) Also, it is essential for the results to be consistent with the ones an operator would have manually selected. This means that there should be very few instances in which the machine aborts (refuses to punch because of inconsistencies) or punches films that would later be considered unacceptable for printing.
- Finally, there exist **cost** considerations which impact the design in terms of reasonable computer resources, and **acceptability** issues in a well established industry.

A solution to these problems is described in the text of two recent patents[1][2].

In machine vision terms, the problem is to match two images, typically 20 × 25 cm^2 (8"×10"), with 1/1000 in accuracy or better. These images, however, are halftone, which means that they are already coded on a grid, with each dot size proportional to the corresponding intensity.

From one color separation to the next, the dots are not overlapping because the grids are rotated between colors (to avoid moiré effects). In fact, corresponding dots in registration form a rosette pattern.

As a result of this situation, methods based on the correlation of intensity values are limited: if the resolution is coarse enough, pixels will indeed correspond to each other, but the accuracy will not be sufficient; if the resolution is finer, then pixels do not correspond to each other.

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).

1.3 Implemented Solution

Three issues are involved in the process described above: representation, matching and verification. *Representation* refers to the selection of features and to the procedures used to extract these features from the input images. *Matching* refers to the procedures applied to each pair of images to obtain a measure of their relative mis-registration. *Verification* refers to procedures used to verify the relative mis-registration, to estimate rotation, and to the subsequent decision to punch registration holes to the films.

We illustrate the problems on a specific example. Figure 1(a) and (b) show the cyan and magenta images corresponding to the 7"x11" negatives. Following the strategy used by human strippers, an operator roughly aligns the films to the reference, and selects two detail areas on the reference, with a cursor.

The machine then takes over, automatically extracts features in the windows, matches them, and verifies the quality to either punch registration holes or reject the set of films. This is illustrated in Figure 2.

Feature Extraction

As noted earlier, windows in corresponding areas of two separations can appear very different, as shown in Figure 3. We therefore use as matching primitives the macro edges between regions of different dot density. They are extracted as subpixel zero-crossings of the convolution of the image with a large Laplacian of Gaussian mask[4], where the space constant is a function of the screen resolution. This computation is performed efficiently[3]. These contour points are then linked, thresholded using hysteresis, and approximated by linear segments.

Window Matching

By establishing correspondences between features extracted from a window, we obtain an estimate of translation (or mis-registration) in the two areas of the film, and then combine these results to estimate the rotation in the whole image.

The translation between two windows needs to be estimated robustly, as only subsets of features pair up. We use a Hough-like voting scheme in translation space, adapted for linear segments.

Good matches produce a sharp peak, whose position gives the estimate of translation. We post process the results to handle choke and spread effects, which produce crater-like peaks.

Verification

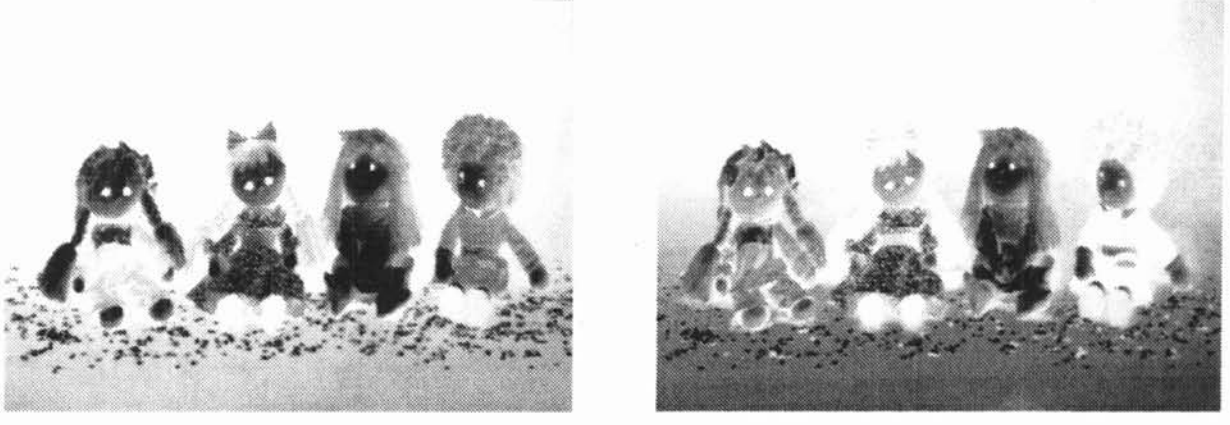
Once the two selected windows are matched, it is possible to estimate a rigid transform (rotation and translation) between the two negatives, by aligning the vectors joining the centers of the windows in each film. If these two vectors differ in length by more than 3 mils, the punching procedure is aborted.

The machine

The machine performs the registration of a four color set of films in about 5 minutes, including all operations. This corresponds to a cycle time of 12 seconds to match two windows. 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.

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 machine is now installed at several sites and performs the registration faster and more reliably than human operators.

The physical machine is shown in Figure 4.



(a) CYAN film

(b) MAGENTA film

Figure 1 Two Color Separation Films

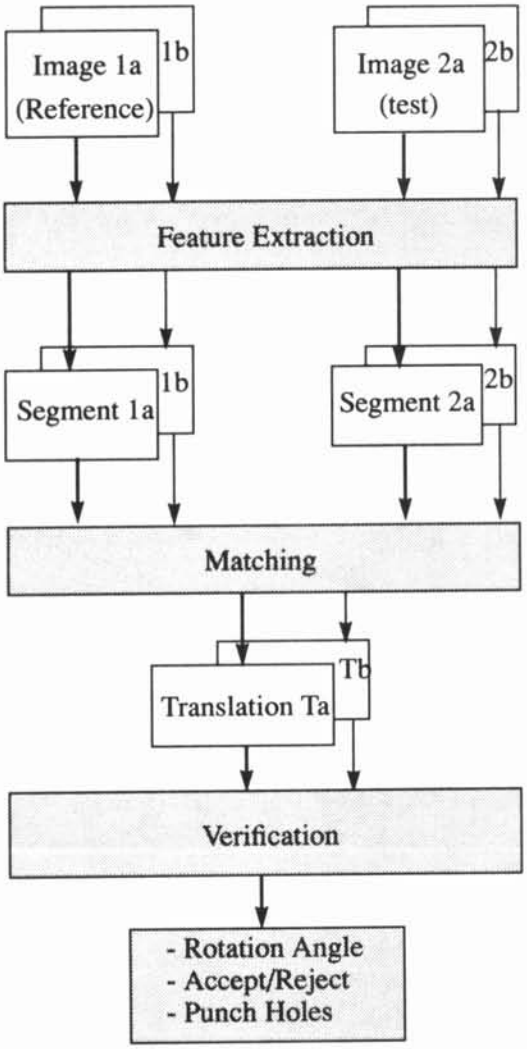
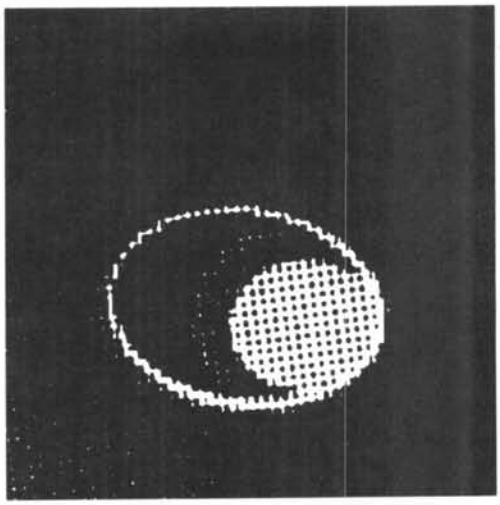
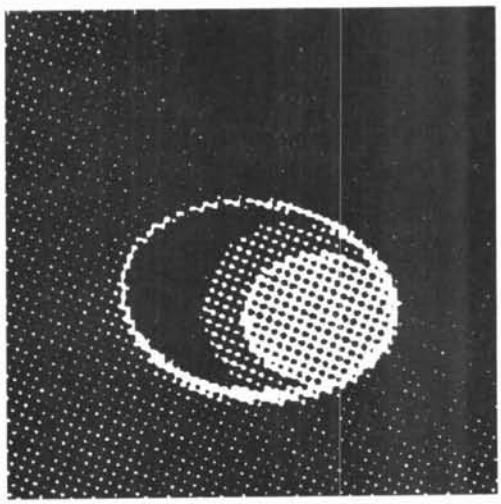


Figure 2 Flowchart of the Method



(a) CYAN film



(b) MAGENTA film

Figure 3 Detail Windows

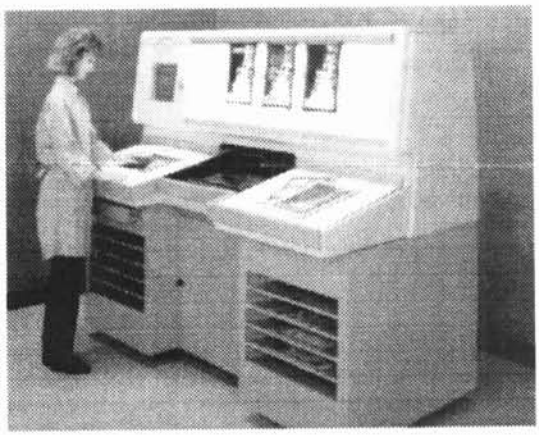


Figure 4 The Physical Machine

2 Billboard Replacer

2.1 Background

It is a common practice to place billboards advertising various products and services during sports events. These billboards target not only the spectators at the stadium, but also (and mostly) the viewers of the TV broadcast of the event. This fixed advertising is therefore limited, as the billboards might be advertising products out of context for the TV audience, especially for international events.

We are presenting a system to automatically substitute, in real-time, one billboard by another, synthetically created, billboard. It aims at replacing the billboards in the scene in such a way that it should be transparent to the viewer. It allows a local TV station to plant its own advertisement billboards regardless of the original billboard, thus increasing the overall effectiveness of the advertising.

The process by which we accomplish this goal is therefore the composition of a video stream and a still image, to create a new, smoothly blended and photo-realistic video stream.

Editing of images or image streams is fast becoming a normal part of the production process[5]. Many recent movies, such as Terminator 2, Forrest Gump, Casper, ID4 seamlessly blend live images with Computer Generated Imagery. The mixing of multiple elements is performed primarily by *screen matting*, in which the background is of almost constant color, generally blue or green. This approach requires a very controlled studio environment and operator intervention for optimal results.

Our system must instead function without active cooperation, in real-time (therefore automatically, without operator intervention), and in a non controlled environment. Furthermore, it must also adapt the model to fit the observed billboard. It involves the "intelligent," automatic manipulation of images and image streams, *based on their contents*.

The system receives as input a TV broadcast signal, must identify a given billboard in the image flow, track it precisely, and replace it with another pattern (fixed or animated), broadcasting the replaced signal, in real-time, with only a short, constant delay. Figure 5 presents an example frame of billboard replacement.

2.2 Requirements and challenges

The fundamental requirement that the system perform *on-line* in *real-time*, imposes major constraints on the design and implementation. These are:

- No human intervention is possible.
- No on-screen errors are permitted. The system has to include self quality control mechanisms to detect problems and revert to the original signal when they occur.
- Complex high level algorithms are limited due to the need for implementation in real time.
- No cooperation from the field is expected, in order to allow the system to operate independently from the imaging process (e.g. at the down link).

The challenges stem from the requirements listed above: real-time operation, simple algorithms, and extreme reliability. This last point requires a redundant design in which failure of an individual module must be compensated.

The contribution of such a system, for which a patent was issued[6], thus resides both in the design and implementation of the individual modules (finder, tracker, replacer), and in the management of failure and uncertainty for each of these modules, at the system level, resulting in reliable replacement.

2.3 Implemented Solution

Overall System Design

The task of the system is to locate a planar, rectangular target billboard in the scene, detect camera switches, track the billboard throughout the sequence (between camera switches), and replace it with a new billboard. The direct naive approach would be to inspect the incoming frames, search for the billboard and replace it. Unfortunately, this approach is not sufficient, as it may be impossible to locate the billboard in the current frame: This may be due to large focus or motion blur, or to the billboard being occluded, or to the fact that only a small part of it may be in the field of view. The billboard may therefore be found only in a later frame of the sequence, and it is not advisable to start replacing then, as this would be offensive to the viewer. Instead, replacement should be performed on the whole sequence to avoid billboard switches on screen.

Our system relies on modular design, and on a pipeline architecture, in which the search and track

modules propagate their symbolic, low-bandwidth results throughout the pipe, and the replacement is performed at the exit of the pipe only, therefore relying on accumulated information. This allows the system to make replacement decisions based on complete sequences, thus avoiding mid-sequence on-screen billboard changes.

The *Finder* module searches for the target billboard in the entering frames and passes its results to the *Updater*, which propagates them throughout the buffer.

The *Global Motion Tracker (GMT)* module estimates the motion between the previous and current frames, *regardless of whether the billboard was found or not.* This is used as a mechanism for predicting the billboard location in the frames in which it was not found. The prediction is necessary to ensure continuity of replacement, since we do not want the billboards to switch back and forth between the original and the new one in front of the viewer. The GMT also performs the task of camera switch detector.

The *Replacer* performs the graphic insertion of the new billboard, taking into account variations from the model due to lighting, blur and motion.

The *Updater* handles communication within the buffer and also manages the *Measure Of Belief (MOB)* associated with the information passed along, due to the MOB of each of the modules, and a decay related to the length of the propagation. The information about scene changes is also used so that the *Updater* does not propagate the predictions beyond the scene change markers.

Figure 6 presents the overall system architecture. As the frame at time *t* comes in from the video source on the right, the Finder searches for the billboard. At the same time, the Global Motion Tracker (GMT) computes the camera motion between the previous and current frames, and stores it in an attribute record. If the billboard is found, its four corners are recorded in the attributes record, and the Updater unit predicts the location of the billboard in all the (previous) frames from the first frame of the sequence to frame *t-1*, based on the computed motion, and updates the attribute records accordingly. As the frame is about to be displayed, the Replacer performs the insertion.

Let us consider the difficult case where the billboard is slowly entering into view, as a result of a pan or zoom. In this case, the billboard cannot be found initially by the Finder. As the frames continue to come in, the Global Motion Tracker computes the camera motion between frames, regardless of whether the billboard was found or not. The camera motion parameters found are stored in the frame attribute record to be accessed by the Updater. When the billboard is reliably found in some frame, *t*, of the sequence, the Updater module uses the motion parameters computed earlier, to predict the location of the billboard in all the frames from the first frame of the current sequence up to frame *t-1*. Since this is a very simple computation (not image based), involving low bandwidth communication, it can be performed for the whole buffer in one frame time. As the images reach the end of the buffer, we know the location of the billboard, either directly form the Finder, if it was found in this frame initially, or via a prediction from the Updater, using the motion information.

The combined use of the Global Motion Tracker, the delay buffer and the Updater mechanism, allow the system to, in essence, go back in time without having to process the image again, and to use information from the current frame to locate the billboard in earlier frames. This enables the system to perform well under varied conditions, such as occlusion and entering billboards. The system is also very robust to failure of specific modules, as it can overcome failure in some frames by using information from the other frames of the sequence. It is important to note that each image is processed once only, and that each module works at frame rate, thus the system works in real-time, introducing only a constant delay.

This design can guarantee that no offensive substitution will take place, as long as a whole sequence fits in the buffer. Otherwise, in case of a problem occurring after replacement is started, a smooth fade back to the original billboard is used. In practice, a buffer of the order of 3 seconds (180 fields in NTSC), covers a large percentage of sequences in which the billboard is present.

Components

Finder: The task of the Finder is to examine the incoming frames and find the position of the target billboard if it is present in the scene. It first extracts "interesting" points (corners, or other "busy" formations) in the image, then selects the interest points

which are most likely to come from the target billboard based on color information. It then finds a set of corresponding points between model points and image points, and uses these correspondences to find the precise (to a sub-pixel resolution) location of the billboard.

While most of these steps are fairly intuitive, the point matcher module deserves a lengthier explanation. Its output is the transformation between the model and the image and, therefore, the position of the billboard in the image. The direct exhaustive attempt to match every point of the model with every point of the image leads to unacceptable complexity. Instead, we use an affine-invariant matching technique proposed by Lamdan and Wolfson[9]. We search for the best affine transform which maps the model points to the image points. Any three points define a *base*, relative to which all other points have some relative coordinates. A set of any three matching pairs of points (matching bases) uniquely determines the six parameter affine transform between the model and the image.

The algorithm consists of two components. A pre-processing of the model points, which is time consuming but performed off-line, and an efficient on-line process. During the pre-processing step, the coordinates of all the model points with respect to all possible bases (triplet of model points) are computed and stored in a hash table. The coordinates are used as the index into the hash table, and the base as the value.

During the on-line processing, the feature points of the current image are considered. A base (triplet of points) is selected at random, and for all feature points in the image, their coordinates are computed with respect to the selected base. The coordinates are used to index into the hash table and vote for the model bases in that entry. If any one of the model bases receives a sufficient number of votes, it is considered a possible match. The two bases (the one selected from the image and the one receiving most votes) define a transformation between the model and image. This transformation is applied to all model points, and, if a sufficient number of the transformed points indeed find a match, then the process ends, otherwise, a new base is selected.

We do not process all possible bases during the on-line processing. If a match is not found after a fixed number of random trails, then the matching fails for the current frame.

Global Motion Tracker: The Global Motion Tracker computes estimates of the global camera motion parameters between consecutive frames. Since we are interested in the motion of the camera and not in a per pixel motion, we take a global approach, and use an iterative least squares technique on all pixels of the image[7]. This results in an efficient and robust method which produces accurate results.

The images are first smoothed and the spatial and temporal derivatives are then computed. Using this information, estimates of the motion parameters are computed. Using these estimates, Frame $t+1$ is warped towards Frame t, and the process is repeated. Since Frame $t+1$ gets closer to Frame t at every iteration, the motion parameters should converge. The accumulated parameters are then reported to the Updater. If, however, the motion parameters do not converge after a fixed, predetermined number of iterations, the process is stopped with a report of zero reliability.

We have implemented the algorithm at multiple levels of resolution. A Gaussian pyramid is created from each frame[8]. At the beginning of a sequence, the algorithm is applied to the lowest resolution level. The results from this level are propagated as initial estimates for the next level up, up to the highest level. This allows for recovery of large motions.

An improvement to the global motion algorithm allows for accurate and stable results, even in the presence of independently moving obstacles in the scene. This is achieved by scaling the coefficients of the motion equations inversely proportional to the temporal derivatives. Moving obstacles do not match when the images are warped according to the camera motion. Therefore, pixels corresponding to obstacles produce high temporal derivatives, and consequently contribute less. The improved results allow for long propagation of estimates along the sequence.

The complete algorithm has been applied to a very large set of sequences. The parameters recovered are accurate enough to allow for the accurate propagation of the billboard location for a few seconds, in the absence of confirmation by the Finder. The algorithm fails when the motion is drastically different from the

modeled motion, or when the image is uniform (e.g. an image of the sky).

Updater: The Updater's task is to collect data from all the other modules, and to correct missing or inaccurate information within a processed sequence. We can visually think of the system as a circular buffer, holding a frame and a frame attribute in each of its cell. The Updater manipulates these attribute records only, which are composed of a small number of parameters, and processes *all* attribute records in the buffer in one frame time.

Input from other modules are read in, and a hypothesis from the last to current frame is computed and compared with the current hypothesis. If it is larger (in terms of reliability measures), it replaces it. Then the propagation backward in time is started. The propagation process is stopped if a blocking flag is encountered in the process, due to a scene change, a high enough existing measure of belief, or a low enough measure of belief for the current hypothesis.

For example, when the Finder fails to find the billboard in a certain frame, it assigns a zero certainty to the data it hands the Updater. The Updater then interpolates the position of the billboard in this frame using data from the Global Motion module and/or previous frames, and/or future frames. The Updater changes the Measure of belief accordingly.

The decision of whether to transmit a given sequence is made as late as possible. The latest is when the first frame of the current sequence is about to exit the buffer. The Updater then evaluates the global "goodness" of the sequence, and if it passes a fixed threshold, a decision is made to transmit-with-replacement.

Replacer: Given the coordinates of the billboard corners in the current image, the Replacer module replaces the image contents within these corners (the billboard) with the new desired contents (usually a new billboard). This is a standard application of computer graphics. Also, because the human eye is quite sensitive to sharp changes in colors, we correct the gain and offset of the replaced billboard to make it appear close to the average intensity of the image. Note that we currently assume that the original billboard is un-occluded. Mechanisms which allow for detection of obstacles in front of the billboard are currently under development with promising results.

The Machine

A design somewhat simpler than the one described here has been made operational by Matra CAP Systèmes using off-the-shelf components, and used by Symah Vision for live broadcasts. The accompanying video illustrates the resulting performance of the system.

This successful aggregation of computer vision and computer graphics techniques should open up a wide avenue for other applications, which are either performed manually currently, or simply abandoned as too difficult.

On a different note, it is interesting to note that such a system also casts some doubts as to the authenticity of video documents, as predicted in fiction such as Rising Sun. It shows that digital video documents can be edited, just like audio and photo documents.

3 Conclusions

We have presented two challenging applications critically relying on computer vision, and which resulted in functioning physical machines. A number of lessons can be learned from these examples:

- the field of computer vision has matured, and many techniques are producing consistent results.
- while the techniques are understood, we still do not have a bag of tools which we could use as plug-in components of a solution.
- instead, each application requires a large amount of effort to customize algorithms.
- integration of the machine vision components into a machine also presents challenges, as standard interfaces do not exist.

References

[1] G. Medioni, A. Huertas, and M. Wilson, *Method and apparatus for registering color separation films*, U.S. patent # 4,849,914, July 1989.

[2] M. Wilson, and G. Medioni, *Method and apparatus for registering color film separations*, U.S. patent # 5,023,815, June 1991.

[3] J. Chen, A. Huertas, and G. Medioni, *Fast Convolution with Laplacian-of-Gaussian Masks*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, No. 4, pp. 584-590, July 1987.

[4] A. Huertas, and G. Medioni, *Detection of Intensity Changes with Subpixel Accuracy using Laplacian-Gaussian Masks*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 5, Sep. 1986, pp. 651-664.

[5] R. Fielding, *The Technique of Special Effects Cinematography*, Focal/Hastings House, London, 3rd edition, 1972, pp. 220-243

[6] G. Medioni, G. Guy, and H. Rom, *Video processing system for modifying a zone in successive images*, U.S. Patent # 5,436,672, July 1995.

[7] J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg, *A Three-Frame Algorithm for Estimating Two-Component Image Motion*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 9, pp. 886-896, Sep. 1992.

[8] P.J. Burt, *Fast Filter Transforms for Image Processing*, Computer Graphics and Image Processing, Vol. 16, pp. 20-51, 1981.

[9] Y. Lamdan, J. Schwartz, and H. Wolfson, *Affine Invariant Model-Based Object Recognition*, Robotics and Automation(6), 1990, pp. 578-589.

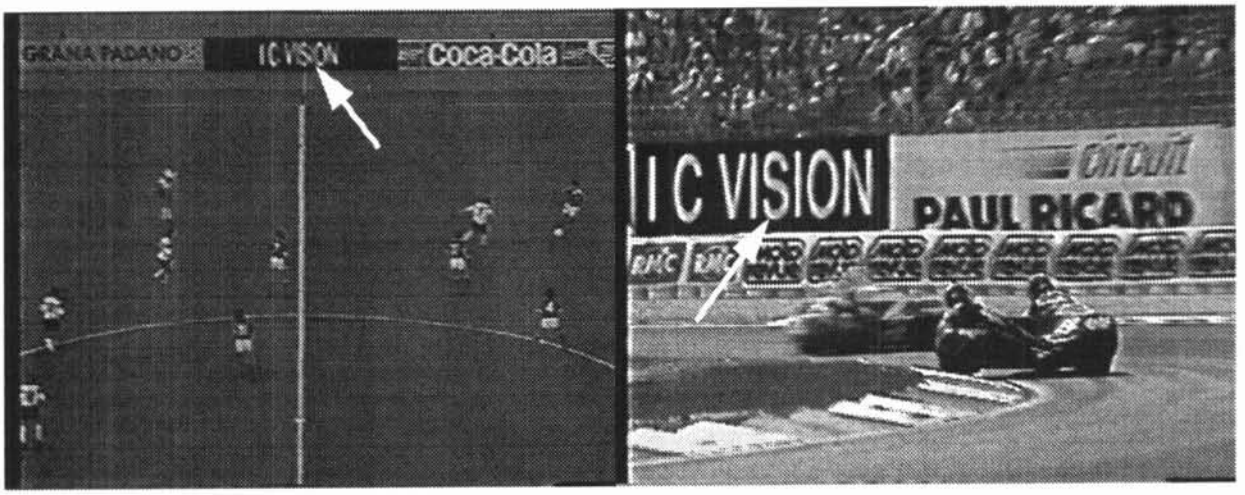


Figure 5 Two examples of billboard replacement in a Video Sequence

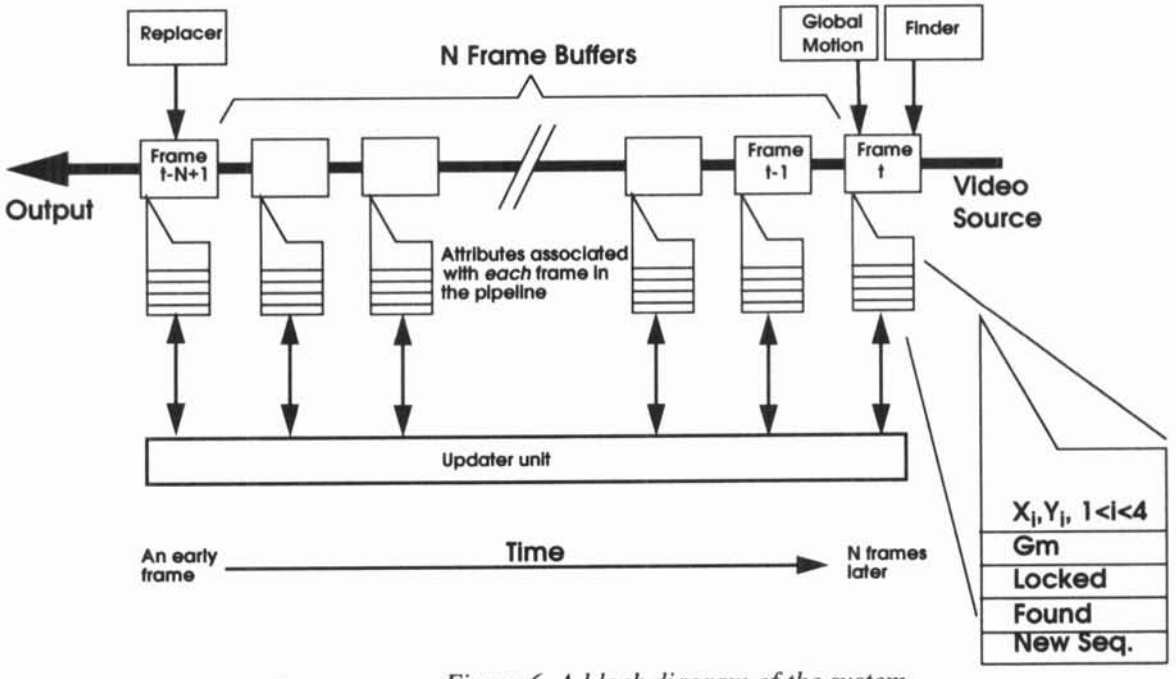


Figure 6 A block diagram of the system.