SEP 4 1997

SANDIA REPORT
SAND97-2091 • UC-906
Unlimited Release
Printed September 1997

RECEIVED
SEP 0 8 1997
OSTI

Extremely Secure Identification Documents

Keith M. Tolk, Michael Bell

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; distribution is unlimited.

Sandia National Laboratories

MASTER

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.
NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SAND97-2091
Unlimited Release
Printed September 1997

Distribution
Category UC-906

Extremely Secure Identification Documents*

Keith M. Tolk
Cooperative Monitoring Systems Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0656

Michael Bell
Engineering for Instrumentation Systems
Sandia National Laboratories
P.O. Box 969
Livermore, CA 94551-0969

Abstract

The technology developed in this project uses biometric information printed on the document and public key cryptography to ensure that an adversary cannot issue identification documents to unauthorized individuals or alter existing documents to allow their use by unauthorized individuals. This process can be used to produce many types of identification documents with much higher security than any currently in use. The system is demonstrated using a security badge as an example.

This project focused on the technologies requiring development in order to make the approach viable with existing badge printing and laminating technologies. By far the most difficult was the image processing required to verify that the picture on the badge had not been altered. Another area that required considerable work was the high density printed data storage required to get sufficient data on the badge for verification of the picture. The image processing process was successfully tested, and recommendations are included to refine the badge system to ensure high reliability. A two dimensional data array suitable for printing the required data on the badge was proposed, but testing of the readability of the array had to be abandoned due to reallocation of the budgeted funds by the LDRD office.

* The work reported here was funded under Sandia National Laboratories' Laboratory Directed Research and Development (LDRD) funding during the 1993and 1994 fiscal years under case number 3517.040. Prior work in the area by Dr. Tolk was funded by the Bureau of Engraving and Printing, the Department of Energy, and the Defense Nuclear Agency.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Acknowledgments
Several people contributed to the success of this project. Some that deserve special recognition include Trish Larson and Gary Simpson for their input in the system design and definition and their work on software development. Jack Bartberger designed and oversaw the construction of the special hardware required for the badge reader and special instrumentation. Victoria Hamilton, Tim Draelos, and Marjorie Jimenez provided the authentication software and consulted on data surety issues. The authors wish to thank each of these individuals for their help.

Contents

Executive Summary

Technology developed for the verification of nuclear arms control treaties can be used to protect security badges, credit cards, currency[1], and other documents from even the most determined professional counterfeiters. The price for this increased security is the need for a reader/authenticator device at every point that the authenticity of the documents must be proven. A considerable increase in the amount of data stored on the documents is also required. Whether the increased cost and complexity are justified in order to obtain the increased security must be determined for each potential application. As the cost of the electronics required for the reader/authenticator continues to decrease, this approach will be economically attractive for more applications. Also, when the amount of data that can be economically stored in smart cards increases and smart cards become more familiar to the general public, the data storage problem that is a limiting factor with the example presented here will be alleviated.

The cost for implementing the technology depends on the number of authentication units required. For relatively small quantities using commercial hardware available today, the cost of each of the units would be on the order of $1600. If several hundred of the units are built, the cost would be less than half that much. If several thousand units are built with custom hardware developed specifically for this purpose, the cost might get as low as two to three hundred dollars. The equipment cost for printing the badges and the cost of the badge itself would be essentially unchanged from today's costs.

A unique feature is added to or identified on each type of document to be protected. Data describing this feature and other identifying information such as the authorized user's name, account number, date and place that the document was issued, security related information, and possibly biometric data are recorded on the document. A data signature produced using public-key data authentication is also stored on the document. When the document is presented for use, the authentication signature is verified to be appropriate for the data file on the document using one of a set of public keys corresponding to the set of authorized issuing stations. This verifies that the data has not been tampered with and that the document was issued by an authorized issuing station. Then the pattern of the unique feature is read and compared to the data file to verify that the feature has not been altered. The authenticity of any other identifying data is also verified at this time. If all of these tests are passed, the document is known to be authentic and unaltered.

When biometric data is used as the unique feature in this application, it also serves to tie the credential to the authorized user. Information derived from the photograph or fingerprint printed on the document works well for this type of application. The photograph has the added advantage that a human can do a pretty good job of comparing the photograph to the individual without the need for additional equipment. If more security is required, other biometric information such as hand geometry or retinal scan can be included and verified using appropriate hardware.

The badge example used was a very difficult application of this technology, but one that should be generally acceptable to the general public. It required extensive development in image processing, data compression, and data storage technology. Unfortunately, the LDRD program reduced the funds allocated for this project before hardware required for testing of the data storage technology could be completed.

Introduction

One of the first steps in designing a counterfeit deterrence system should be to analyze the level of sophistication and the motivation of the expected adversaries. For example, if we are concerned about counterfeiting of hundred dollar bills, we only need to make the system secure enough to force a potential adversary to spend more than one hundred dollars to make each bill in order to make the enterprise economically unattractive. In fact, a cost of ten percent of the face value is probably adequate to deter most counterfeiters.

Many system designers tend to underestimate the sophistication and motivation of their adversaries. We at SNL tend to design counterfeit deterrence systems with the highly motivated, very sophisticated adversary in mind. It is often easier to make a system less secure and therefore less expensive than to add security to a basically flawed system. One reason that we approach the problem in this manner is that most of the applications that we have been involved with have been concerned with very determined, very well funded adversaries, such as foreign governments. These adversaries might be willing to spend millions of dollars on counterfeiting equipment. In one instance, it was estimated that the adversary might be willing to spend over a million dollars to make a single counterfeit. While most systems do not need to address such determined adversaries, many of the same principles can be applied to less demanding applications.

Don Bauder, who originated this work at SNL in the late '70's, developed the following four basic principles relating to duplication and counterfeiting of tags and similar items:

- Any pattern made to a specified design can be duplicated using identical technology. If an adversary is willing to buy the same equipment that was used to make the original, he can probably make an exact duplicate. The basic axiom is "What one man can make, another can copy."[2]
- Any surface feature can be duplicated. Although there is a limit to the level of detail that can be copied, this has been demonstrated to be true for all magnifications that are practical for field use.
- Any two dimensional pattern can be duplicated, no matter how complicated.
- The most difficult pattern to copy is a multidimensional pattern produced by random processes. Reflective particle tagging technology[3] is based on this principle and no practical process for copying a properly designed tag based on this technology has been demonstrated. Biometric data, including the photograph on a badge, can also be

thought of as multidimensional random patterns based on the individual's DNA information.

Random patterns are used for counterfeit deterrence by reading descriptive data from the pattern and comparing it to similar data taken earlier, usually at the time the document was issued. The following components are required for a random pattern system:

- A suitable random pattern
- A reader system for reading descriptive data from the pattern
- A means for storing the data for comparison to the pattern on subsequent readings
- A means for comparing the data sets to determine if the pattern is authentic.

In order for a pattern to be considered for this type of application it must satisfy the following criteria:

- Stability - It must not change over its useful life when exposed to the most severe of its expected environmental conditions.
- Readability - A reader system must be designed to read descriptive data from the pattern.
- Non Duplicability - The pattern must not be duplicable by any practical means.
- Uniqueness - All patterns generated by the process must be different enough from each other that a randomly produced copy cannot be confused with the original pattern.

Several different reader systems are possible, depending on the random pattern used. For optically readable patterns, film cameras, still video cameras, and video cameras are available. For magnetic patterns, read heads similar to those used in magnetic stripe readers are used.

Data storage can be accomplished using various forms of digital storage or the analog waveforms can be recorded. The data can be stored at a central site and transmitted or carried to the verification site, or the data can be stored on or with the item to be authenticated by using encryption or authentication techniques. The latter method is the most attractive for use on currency, credit cards, and identification cards.

Some form of correlation calculation is generally used for comparing the data sets to verify authenticity.[4]

The need for equipment to read the information and the need for a means of storing the data for comparison to prior readings are the main drawbacks to the use of random pattern technology. However, for long term security, no other technology has proven to provide the level of security that random patterns provide.

Random patterns that may be of interest include the shape and location of special fibers added to paper during the manufacturing process, the random timing errors that occur in recording magnetic stripe information[5], and biometric information. Biometric information is generated by an individual's genetic patterns and can also be thought of as random patterns. Biometric data printed on an identification document can be used not only to verify the authenticity of the document, but can also tie that document to the person authorized to use it. This is the basis of the application discussed in this paper.

The purpose and focus of the work reported here was to identify the technologies that need to be developed in order to implement a secure identification document system based on the concept, develop those technologies wherever possible, and illustrate the system with an example application. Some applications require very little technology development. If the identification system was based on a biometric device using very little data storage, such as hand geometry, and the identification document could store the required data plus the 20 bytes required for the public key authentication, the required technology to implement the system is available commercially. The only task required would be to engineer the system and to develop a key distribution system. A more ambitious example, that of a badge system based on existing printing technology and relying on a picture to identify the individual, was chosen. This required significantly more development. The most difficult task was the image processing required to reduce a photograph to a small enough data set to be printed on an identification document while retaining sufficient information to detect alteration of the image. Another of the major tasks was to develop the data storage technology and a reader capable of reliably reading the data stored.

The remainder of this report deals with the badge example and the technology development that was performed to make the badge system possible. First, there is a description of the badge system. Then each of the technologies is discussed in some detail. The report ends with a short discussion of the conclusions reached and recommendations for further development.

System Description

The application can best be illustrated by use of an example. Consider the security badges issued by Sandia and other DOE laboratories. The main features of the badge are a picture of the individual, his or her name, any special access information, and a printed substrate to identify the issuing agency. These items are laminated to prevent damage and to deter an adversary from making changes. In the proposed system, additional information would be included with the badge in authenticated form. This information would include the person's name, a control number, date and place of badge issue, access authorization information, and authentication information for the photograph on the badge. This process is shown in Figure 1. Figure 2 shows the corresponding verification process. When the individual presents the badge for access to a facility, it is placed into a reader that scans the photograph and reads the authenticated information.

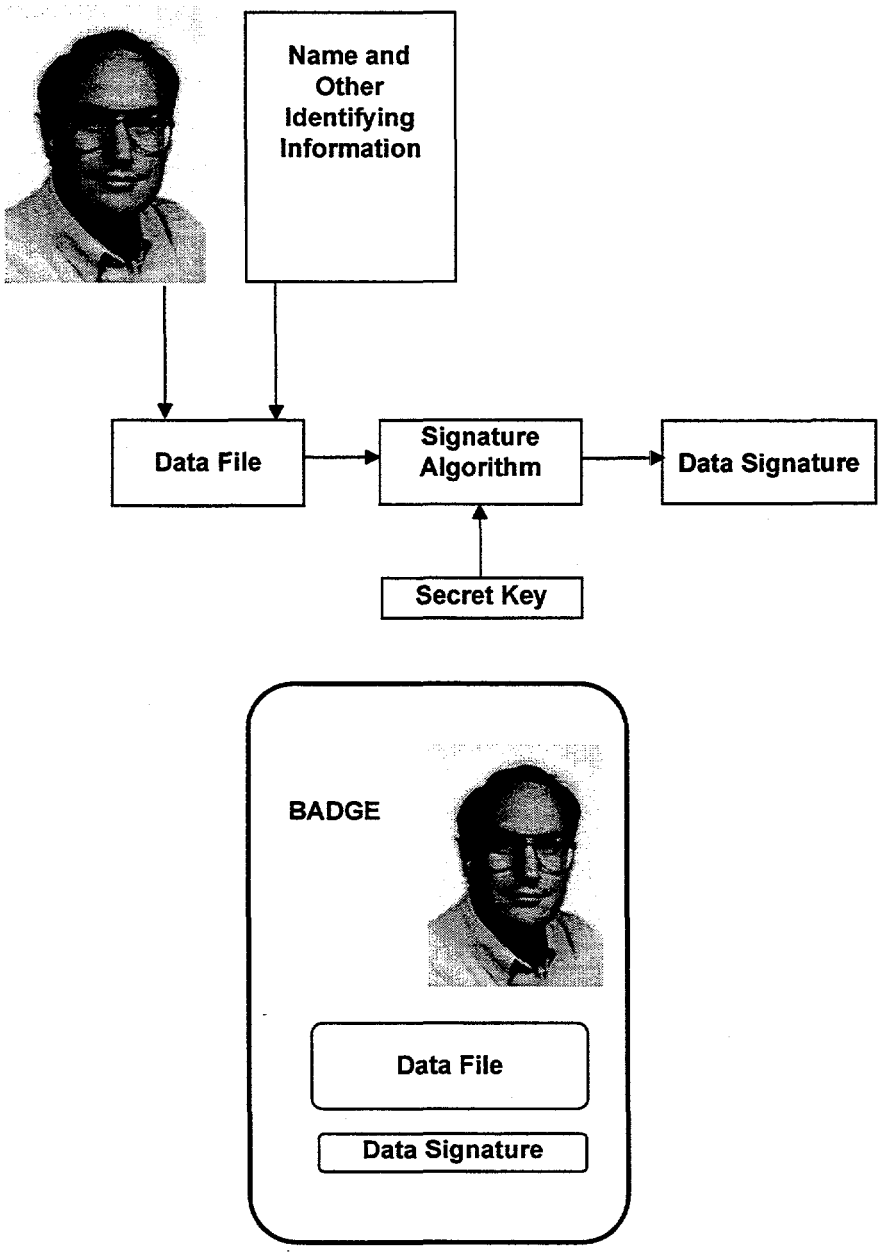


Figure 1
Badge Issuing Process

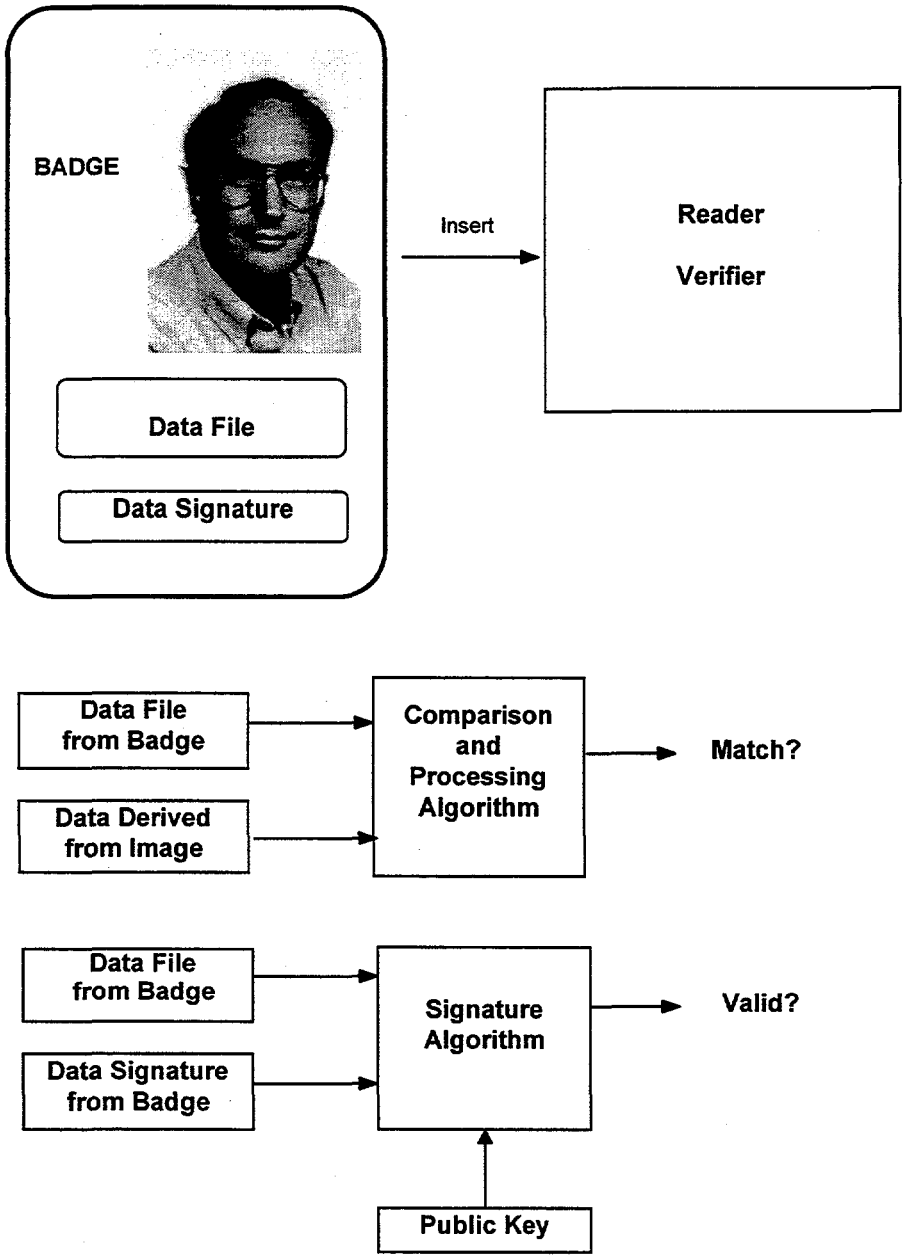


Figure 2
Badge Verification Process

The information is verified using one of a table of public keys. The authenticated information would be compared to the photograph, using algorithms discussed later in this report, to verify that this is the photograph that was on the badge when it was issued. The reader would also compare the control number to a list of lost, stolen, and revoked

badges. On the basis of this information, the reader would then make a decision on the authenticity of the badge. This authentication process will probably take on the order of five seconds using relatively inexpensive computing equipment. The reader would then display the person's name, his security clearance level, and whether or not he is currently authorized access to the facility. The guard or access control officer then decides if the individual is the person pictured on the badge and grants or denies access.

Note that this system does not preclude an adversary from making copies of an authorized badge. The copied badge will be difficult to use, however, unless an individual who strongly resembles the picture on the badge can be found to use it. If this is an unacceptable risk in the security system, other biometric information for the individual can be included in the data file and verified using appropriate technology.

The resulting system provides a robust, secure system that could include many locations without the need for a large database of issued badges. The only databases required are the control numbers of badges (or passports, etc.) that have been canceled by the issuing party and the public keys associated with authorized issuing stations.

Technical Details

Image Processing and Data Compression

Background

The principal way in which the *security* aspect of the E.S.I.D. card (Extremely *Secure* Identification Document) is accomplished, is by making an authenticated, digitally-encoded copy of the photograph that is on the front of the card, and then storing that copy on the back of the card using print media (a 2-dimensional form of "bar code"). During the identification process, the encoded data is decoded and compared to the photograph. If there is not a match, this indicates that the E.S.I.D. card has been damaged or intentionally altered.

The main limiting factor in this process is the density of the data that can be accurately stored and retrieved using the print media. The total amount of data that can be stored is constrained to approximately 3 Kbytes, by the dimensions of the card and by the spatial resolution of the imaging system that reads the encoded data. By contrast, the amount of information available from the photograph is on the order of 250 Kbytes, so only a fraction of that information can actually be stored on the back of the card. Since some parts of the photograph would be more useful for comparison than other parts, a strategy was devised to find the most useful data and store it in the most compact way possible.

Another major factor for consideration was the variations inherent in the card reading process. The largest effects would be due to deterioration of the card caused by smudges, delamination, scratches, etc. The digitally-encoded data would be resistant to these effects

up to a point, and then degrade quickly, resulting in severe problems for the identification algorithms. It was decided that error-correction coding would be necessary to avoid these problems. On the other hand, the quality of the photograph would deteriorate linearly, with no threshold, and the effect on the identification algorithms would be far less severe.

Other variations in the card reading process included lighting and card-position in the reader (offset and rotation). Lighting was fairly easy to correct for, but tests were performed to determine how much effect position would have, and to decide whether or not it should be corrected for.

The Badge-Making Process

Reducing the Photographic Data

In an initial attempt to reduce the amount of photographic data from 250 Kbytes to just 3 Kbytes, it was decided that the most distinguishing characteristics of a face were in the *shapes* of the features ... head, eyes, nose and mouth. By turning the photograph into something like a line-drawing, there would be only black-or-white (1-bit) pixels instead of 256-level (8-bit) gray pixels, which reduces the data by a factor of 8. Further data reduction could be achieved by *segmenting* the image ... that is, by storing black-and-white sub-images of the areas containing just the left eye, right eye, nose, etc.

To transform the gray-scale image into a black-and-white line-drawing, an image processing technique called "edge detection" was used. This technique basically scans an image horizontally and vertically calculating changes in light intensity. The borders of facial features are identified by marking the larger changes in shading, as determined by some threshold value. This process has the virtue of being very fast. The comparison of two edge-detected images is also very fast, since it can be done by logical operations, instead of multiplications.

The drawback to this method was in its sensitivity to the thresholding operation. To begin with, the edge-detection process never yielded repeatable results, even with manual adjustment of the threshold level, due to variations in lighting and position. Based on that, we were pessimistic about finding an appropriate algorithm for automatically calculating a threshold level. Then, because we always got a slightly different edge-detected image, the property of this process that had been seen as an advantage, became a disadvantage; the black-and-white comparisons indicated a poorer match, for images that should match, than gray-scale comparisons would. For these reasons, this method was abandoned.

A second method, which proved to be effective and was eventually adopted, used segmentation and image compression to reduce the photographic data to 3 Kbytes. Segmentation, as described earlier, was used first to reduce the data to 46 Kbytes ... two

eye sub-images of 80-by-100 pixels, a nose sub-image of 140-by-100 pixels, and a mouth sub-image of 80-by-200 pixels. (Selecting the sub-images must be done manually, since automating that process would require a rather significant image processing effort.)

Next, the sub-image data was compressed 16:1 with a well-known technique, the Discrete Cosine Transform (DCT). Each sub-image, which can be thought of as simply a matrix of spatially-sampled *integer* data, was transformed by the DCT into the frequency domain. The result is another matrix of the same dimensions, but with *real* data representing the coefficients of spatial-frequency components. The inverse transform operation on this matrix would reproduce the original sub-image data exactly, so a comparison could be performed with perfect results. However, since the transform data matrix is the same size as the original image matrix, nothing has been gained yet. The compression is accomplished by storing only one-sixteenth of the transform matrix data. A property of the DCT is that (for highly correlated data - as is the case for most images) nearly all of the information is concentrated in the low-frequency coefficients. (Of the more than 50 images tested, every one had more than 99% of its energy in the low-frequency coefficients.) So, if the high-frequency coefficients are removed, the inverse transform will produce an image that is very similar to the original image. The two can then be compared using an algorithm to be described later, with extremely good results - in our tests of 28 images, every image correlated to a compressed version of itself with a value higher than 98.4%.

Details of the Compression Algorithm

The actual compression/decompression algorithms finally used start with the manual selection of the four sub-images, as described above. Each of the sub-images is then transformed and compressed in a single step, using matrix multiplication, to reduce the amount of computation.

To understand how this is done, let's start with a square N-by-N image, and define the standard equations for a 2-dimensional discrete cosine transform and inverse transform, as shown below.

Forward transform:
$$y(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n) \cdot z(k,l), \quad \text{for } (0,0) \le (k,l) \le (N-1,N-1)$$

Inverse transform:
$$x(m,n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} y(k,l) \cdot z(m,n), \quad \text{for } (0,0) \le (m,n) \le (N-1,N-1)$$

where
$$z(i,j) = \begin{cases} \sqrt{1/N}, & \text{for } i = 0 \text{ and } 0 \le j \le N-1 \\ \sqrt{2/N} \cdot \cos\left[\frac{\pi (2 \cdot j + 1) i}{2 \cdot N}\right], & \text{for } 1 \le i \le N-1 \text{ and } 0 \le j \le N-1 \end{cases}$$

X is the original image, Y is the Discrete Cosine Transform and Z is a set of cosine basis functions. To compute each element in the transform takes NxN operations, where an operation is defined as one multiply and one accumulation. Since there are NxN elements in the transform, it takes NxNxNxN operations to calculate the image transform. For large N, this becomes a very long process even for the fastest computers, but it can reduced by about a factor of N if matrix multiplication is used.

The equations shown above can be rewritten as:

Forward transform: $\mathbf{Y} = \mathbf{Z} \cdot \mathbf{X} \cdot \mathbf{Z}^T$

Inverse transform: $\mathbf{X} = \mathbf{Z}^T \cdot \mathbf{Y} \cdot \mathbf{Z}$

where **Z** is unitary, meaning that: $\mathbf{Z} \cdot \mathbf{Z}^T = \mathbf{I}$ or $\mathbf{Z}^T = \mathbf{Z}^{-1}$

X is the original image *matrix*, **Y** is the DCT transformed *matrix* and **Z** is the cosine transform *matrix* of basis functions. (Z is pre-calculated, using the above formula for z(i,j), as stored in memory.) Calculating the transform or inverse transform this way takes only 2xNxNxN operations. The computational saving results from properties of the transform matrix [1].

Next, to achieve 16:1 compression, the transformed matrix **Y** is "decimated"; that is, all but one-sixteenth of the DCT coefficients are thrown away. As mentioned earlier, when it is the low-frequency coefficients that are kept, very little information is lost. The new decimated matrix $\underline{\mathbf{Y}}$ is illustrated in Figure 3, with zeros shown as place-holders for the high-frequency data that was thrown away. Compression is achieved by storing only the non-zero coefficients (denoted as the sub-matrix $\underline{Y}$). The matrix $\underline{\mathbf{Y}}$ can easily be reconstructed from $\underline{Y}$ by appending the zero place-holders.

$\underline{\mathbf{Y}} =$

$\underline{Y}$	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Figure 3
Matrix Decimation

The transform matrix $\underline{\mathbf{Y}}$ can then be inverse-transformed to create a new image matrix $\underline{\mathbf{X}}$. When viewed, the new image is very similar to the original image, except for some softening of the edges. This is the effect of the high-frequency filtering. Comparing

element-by-element, the *values* in the two image matrices are also very similar; within two percent of each other, on average.

Even more computational savings can be achieved by simply eliminating the matrix calculations that produce the elements which are going to be thrown away. For this "reduced matrix" method, the forward transform equation is rewritten as follows:

$$\text{Forward transform:} \qquad \mathbf{Y} = \mathbf{Z} \bullet \mathbf{X} \bullet \mathbf{Z}^{T}$$

$$\text{New forward transform:} \qquad \underline{Y} = \underline{Z} \bullet \mathbf{X} \bullet \underline{Z}^{T}$$

Where $\underline{Z}$ has dimensions (N/4)-by-N, $\mathbf{X}$ has dimensions N-by-N, and the transpose of $\underline{Z}$ has dimensions N-by-(N/4), so that $\underline{Y}$ has dimensions (N/4)-by-(N/4). The DCT matrix is now calculated and decimated in a single step, and the number of operations is 0.3125xNxNxN, which is a 6.4 reduction factor.

Another way to reduce the number of calculations is to use the Fast Discrete Cosine Transform (which is similar to the Fast Fourier Transform). It requires about 2xNxNxLog2(N) operations; where Log2(N) is the base-2 logarithm of N. Theoretically, for N-by-N images with 16:1 decimation, the reduced matrix method computes faster for N up to 32; above that, the fast transform is quicker. However, when Log2(N) is not an integer, the fast transform cannot be used, unless the image dimensions are forced to be a power of two, by zero-padding. Also, most so-called Fast DCT routines are really just variations of the FFT [2] that work by expanding the size of the input matrix, and run about 6 times slower than a Fast DCT should! True Fast DCTs are hard to find, understand and implement efficiently [3]. For this application, the reduced matrix method was used because it is quite effective and easy.

After segmenting the four sub-images, Cosine Transforming and decimating each sub-image 64:1, the result is four DCT matrices; two 20-by-25 matrices for the eyes, one 35-by-25 matrix for the nose and one 20-by-50 matrix for the mouth. So, the original 46 Kbytes of image data was reduced to less than 3000 DCT coefficients.

Storing the Compressed DCT Data

Unfortunately, the DCT coefficients are not in the same numeric form as the image data. Instead of positive integers in the 0 to 255 range, they are positive and negative floating-point numbers in an extremely wide range. To simply quantize these values into 256 levels would have a serious adverse effect when the inverse transform is performed. The quantization process is non-linear and irreversible, and due to the wide range of the coefficients, quantization makes very gross approximations to the original values.

To minimize the effects of quantization, the coefficients are first multiplied by an equalization matrix. This is similar to an audio equalization filter that boosts the high frequencies in music before transmitting or recording. We take advantage of the same generalization that we used to perform compression: the fact that the low-frequency components are typically stronger than the high-frequency components. To compress the data, we threw away the highest-frequency components. To reduce quantization effects, we boost the remaining high-frequency components so that they will be in the same general range as the low-frequency components. After quantization, the boosted components are then attenuated by the same amount. The effect is that the quantization is not as coarse for the high-frequency components as they are for the low-frequency components.

To find the appropriate equalization filter, a model for each of the DCT matrices was created. By observation, it was determined that a general model for the DCT matrix was that of an exponentially decreasing 2-dimensional function of the form:

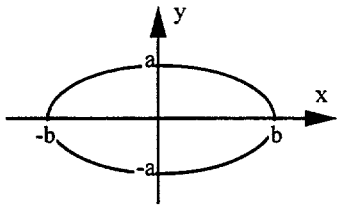
$$V(x,y) = k + p \cdot \exp(-q \cdot z/\text{scaler})$$

where x = 1 to M, and y = 1 to N

$\text{scaler} = \text{sqrt}(M^2 + N^2)$

$z = \text{sqrt}(r^2 x^2 + y^2)$

r = b/a, the ratio of the length-to-width



The elliptical function, z, allows the 2D exponential to decay at different rates along the x and y axes. The function has two variables (x and y), and essentially four parameters, which are:

p - scales the range of the function,
q - defines the overall rate of decay (or curvature) of the function,
r - defines the ratio of the rate of decay between the two axes, and
k - provides an offset, that raises or lowers the entire curve.

The only other parameter is the "scaler", which is simply used to provide a curve of the same shape for matrices of different sizes. Essentially, the curve is normalized to have a domain from 0 to 1, even though x and y are integers that get much larger. So, matrices with more rows and columns only provide more resolution.

This curve was used to create a model for each of the four sub-image DCTs. To find the best parameters for each model, the sub-image DCTs were first normalized and then ensemble-averaged. For example, from the 28 images on hand, 28 left-eye sub-images were captured and cosine transformed. The 28 DCTs were normalized (so that the total signal energy was unity) and then averaged point-by-point. (Actually, the *magnitudes* of each point were averaged.) This creates a single matrix that is representative of all 28 matrices. A gradient search method was then used to find the four parameters that would give an exponential curve which was a best-fit to the ensemble-averaged DCT.

These four curves were used as the equalization filters. Each point in a DCT is *divided* by each point in its respective equalization filter, to boost the high-frequency components before quantization. After quantization, each point in the DCT is *multiplied* by each point in its respective equalization filter, to attenuate the high-frequency components back to their original values (except for some quantization error). The goal is to flatten the DCT before quantization; the ideal *equalization* filter would make all coefficients *equal* (typically a value of one). However, DCTs have positive and negative coefficients, so the model doesn't actually represent the DCT - instead, it represents the *envelope* of the DCT. So, in this case, the ideal equalization filter would make all coefficients either negative one or positive one.

After a DCT is divided by its equalization filter, it is almost ready for quantization. But first, there is one small detail to take care of; the coefficient in the first row and first column is set to zero. The reason for doing this is that this coefficient is by far the largest one in the matrix, and by setting it to zero we can significantly decrease the coarseness of the quantization. Surprisingly, it does not affect the badge identification process. The comparison method (cross-correlation) will be described in detail later, but to put it simply, that method ignores the overall light level of two images when comparing them. If it didn't do that, a picture under one light source would fail a comparison test with the same picture under a darker or a brighter light source. By definition of the DCT, the coefficient in the first row and first column is a direct measure of the overall ("dc") light level in the image. (In frequency domain terms, it defines the zero-frequency content of the image.) So, by throwing away something which isn't used in the comparison process, we don't lose anything; in fact, we gain quantization resolution.

The quantization process starts by expanding or compressing the range of the equalized DCT, so that it does not exceed the range from -127 to +127. This is done by finding the largest-magnitude coefficient (positive or negative), dividing every coefficient by the *magnitude* of that coefficient *(so the signs are not changed)* and then multiplying every coefficient by 127. The resulting values are then rounded to the nearest integer. Changing the range of the DCT does not affect the badge identification process. Once again, the comparison method is immune to this change. One reason is that the cosine transform is a "linear" transformation, so scaling changes in one domain simply results in scaling changes in the other domain. However, another feature of linear transformations is that there is no way to compensate in one domain when a constant is added in the other domain. That is why this particular algorithm was used for the quantization process; to assure that no offset is added to the DCT coefficients.

Finally, the data is digitized, which at this point is simply a form of encoding. First, the constant 128 is added to the DCT coefficients, so that they will be integers in the range from 1 to 255. (This is doing exactly what was just described as detrimental, but in this case, since the constant is a known value, it is simply subtracted from the DCT before the inverse transform is performed.) Then, the integer values are simply encoded in the usual 8-bit binary numerical format.

Fiducials and Additional Parameters

Before getting to the comparison algorithm, it must be noted that there is a small amount of additional data that must be calculated and stored along with the DCT coefficient data. When the four sub-images are manually segmented, parameters identifying their locations within the whole image are also stored. These parameters are used by the comparison algorithm to determine if there is a match to the sub-image only in the area of the image where it is expected. This way the algorithm is not looking for a match all over the entire image, which would take a lot of time, and increase the probability that some part of the image might accidentally produce a fairly high match value. It also makes the badge-image geometry an important part of the badge identification process; i.e., if a facial feature is not where it is expected, then the image has probably been changed somehow. It would also typically preclude two different pictures of the same person from matching.

The problem, however, is the absence of a reference point from which to measure the locations of the sub-images. The badge reader hardware cannot be used as an absolute reference point, due to the tolerances of the card holder, and variations from one reader to another. So, relative reference points, or "fiducials", were added to the badge images themselves. A requirement of the fiducials is that they must be uniquely identifiable from anything else in the image area, using image processing techniques. The fiducials used in this case were not the most ideal fiducials, but they were readily available. Two narrow vertical lines are printed on the badges to help align the photographs, which are positioned by hand. Also, a single broad horizontal line is printed on the badge. The four points where the lines intersect were to be used as the fiducials (see Figure 4). Unfortunately, when the photographs are laminated onto the badge, one of the two vertical lines is invariably covered up (see Figure 5). Also, with dark backgrounds, the vertical edges of the photographs are similar in appearance to the vertical lines. It turned out that the latter problem was used to help solve the former problem, but the computer processing would be much more efficient if the badge markings could be changed. With the current markings, the fiducials can only be approximately located, so the comparison algorithm must do several correlations (which take up most of the badge identification processing time) in the area to look for a match. When better fiducials are used, only a few correlations will be necessary.

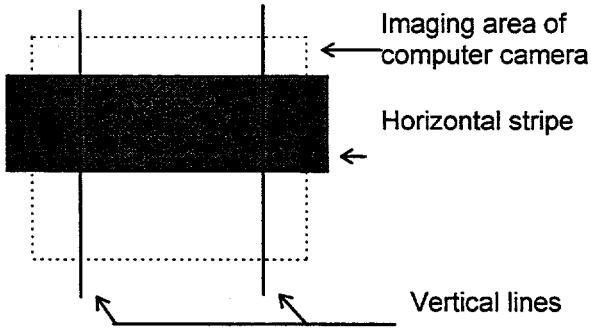


Figure 4
Badge Fiducials

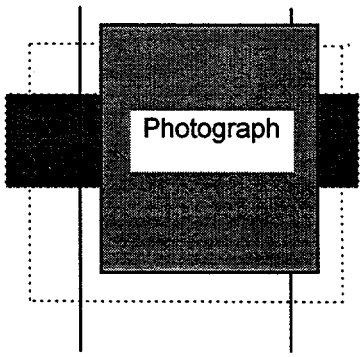


Figure 5
Photograph Mounting

The algorithm for finding the fiducials looks for sharp changes in intensity as the imaging area is scanned vertically on the left side (within set bounds), and then on the right side. This works quite well, locating the top and bottom of the horizontal stripe for each side of the badge. (If the badge is rotated, the top of the stripe is at different elevations on the two sides, as well as the bottom of the stripe.) Next, the imaging area is scanned horizontally in the areas just identified at the top of the stripe, and then in the areas at the bottom of the stripe. The results of these horizontal scans are not as easily interpreted as the results from the vertical scans. Either or both vertical lines could be obscured by the photograph, or if the alignment is poor, the upper half of one line could be obscured, while the lower half of the other line is obscured. For simplicity, the current algorithm takes the sharpest intensity change to be the fiducial point, which could occur at one of three places: the left edge of the vertical line, the right edge of the vertical line, or the edge of the photograph. This algorithm could be improved to select one specific edge, regardless of the intensity of the edges, but it would be even better if more appropriate fiducials were used.

At this point, the horizontal and vertical coordinates of the four fiducials have been identified. Only the upper-left fiducial is actually used as a reference point (the "primary fiducial"), but all four fiducials are used to calculate the rotation of the badge, which is another stored parameter that is necessary for making proper comparisons. Only two fiducials would ordinarily be required to calculate rotation, so the redundant information is averaged in an attempt to make up for inaccuracies in the location of any one fiducial. The formula (shown below in Figure 6) uses two horizontal lines formed by the fiducials, and averages their endpoint locations to construct a third line. The angle of this line is then calculated and used as the amount of rotation for the whole badge. (Horizontal lines are used instead of vertical lines because it is known in this application that they are longer; and since their dimensions are discrete, the extra length provides more resolution to the calculated angle.)

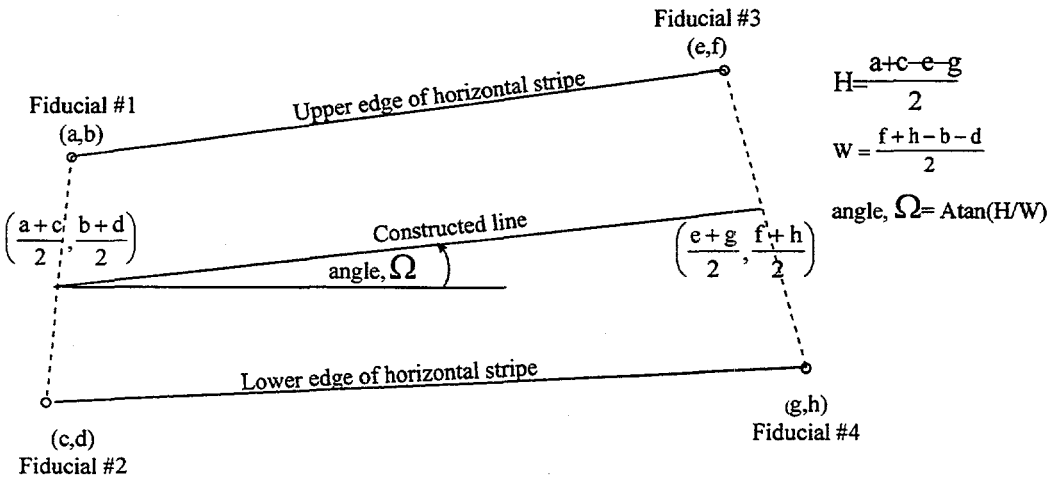


Figure 6

Fiducial Information Averaging

The comparison algorithm (correlation) is not insensitive to rotation; i.e., if the badge is in one orientation during the badge making process, and in another orientation during the badge identification process, the sub-images that were coded and stored will not correlate well with the new image in the badge reader. On top of that, the sub-images will not be in the exact locations that they are expected. The effects of rotation must be compensated for during both the badge making process and the badge identification process to ensure proper identification.

Compensating for Badge Orientation

When the sub-images are manually segmented during the badge making process, the operator guides cross hairs to the area where each feature can be found. The cross hairs mark the center of the sub-image that will be segmented and stored. The location of the cross hairs is also stored as a pair of coordinates indicating the vertical and horizontal offset relative to the primary fiducial. At this point, however, it is already possible that the badge is misaligned, rotating the image slightly. (It has been determined that a maximum of ±2° rotation is possible, based on physical tolerances of the badge readers.) As illustrated in Figure 7, rotation means that the coordinate system of the badge is different from that of the imaging system. The offset coordinates that were found manually use the imagers' coordinate system, so they need to be converted to the badges' coordinate system before being stored; i.e., coordinates (M,N) need to be transformed into coordinates (P,Q). This is done to the coordinate pairs for all four sub-images. Figure 7 also shows that the feature area actually segmented (enclosed by the solid lines) is based on the imagers' coordinate system, and is a rotated version of the image that would

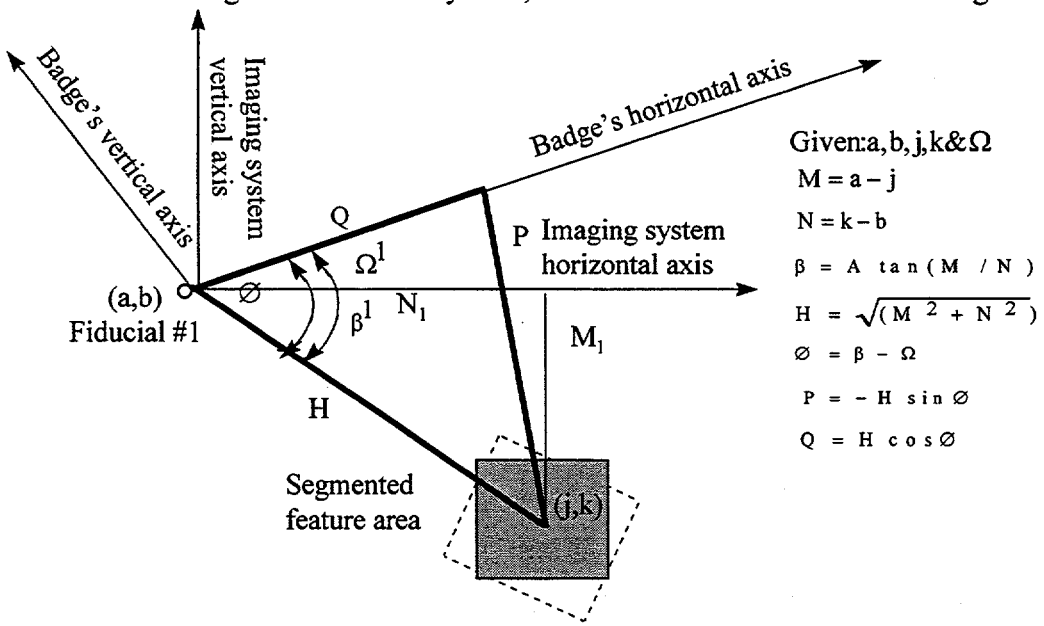


Figure 7
Badge and Imaging System Coordinates

have been segmented (enclosed by the broken lines) if the badge had been aligned properly. Since the amount of rotation is known, it might also be prudent to transform the coordinate system of the sub-images before storing that data. However, it would just have to be rotated again to compensate for rotation during the badge identification process. Just one rotation of the sub-images is adequate, based only on the difference in rotation from the badge-making to the badge-reading processes. That image-rotation compensation must be done later during the badge-reading process. So, the DCTs of the un-rotated sub-images, along with their offset coordinates and the amount of rotation at the badge-making time, are optically encoded and stored on the back of the badge.

The Badge Identification Process

Fiducials and Badge Orientation

Locating the four fiducials and calculating badge orientation, during the badge identification process, is performed using the exact same algorithm that was used in the badge making process. However, when compensating for badge orientation, this time the offset parameters (P,Q) are known, and must be transformed to find parameters (M,N) using the equations found in Figure 8. (P & Q are read from the coded data on the back of the badge; and a, b & ½ come from the fiducial locating algorithm.) Assuming that the fiducials are properly located, and that the angle is accurately calculated, offsets P and Q should provide the imaging system with coordinates (j,k) which do indeed locate the center of the feature area, regardless of rotation during the badge making or badge reading processes. Once again, however, the feature sub-image that is segmented (automatically, this time) will be rotated with respect to the badge's coordinate system, if it is not perfectly aligned. But perfect alignment is not really desirable, because only if the rotation during badge reading is the same as the rotation was during badge making, will the new sub-image exactly match the sub-image from the badge making operation. In Figure 8, the segmented feature area enclosed by a broken line indicates how the feature would have been segmented if the badge had been aligned properly with no rotation. The area enclosed by the solid line indicates the sub-image that will be used for comparison in the badge identification process. In both Figures 7 and 8, the areas enclosed by the broken line are purely academic. What gets compared are the sub-images that are enclosed by the solid lines.

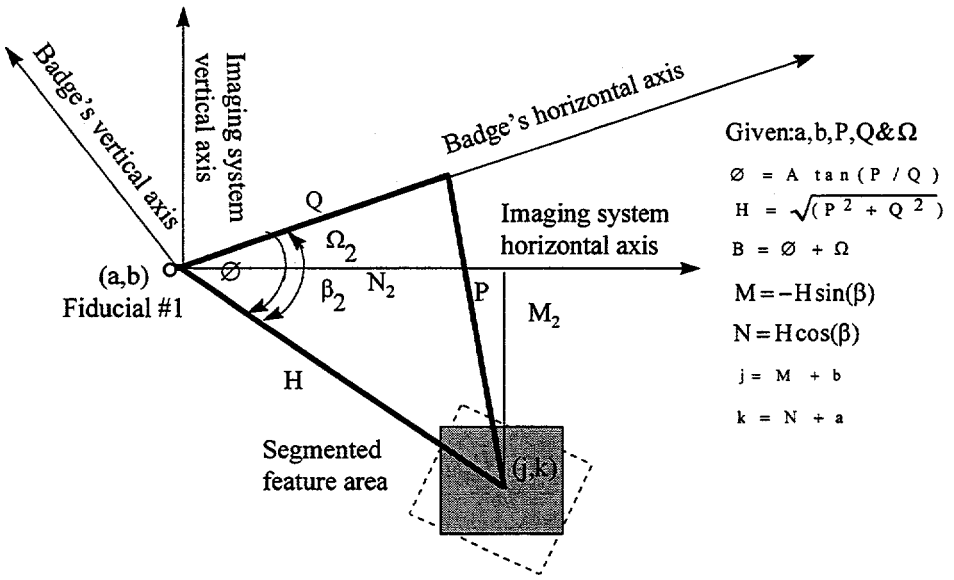


Figure 8
Segmented Feature Area with Alignment Difference

These two sub-images are illustrated in Figure 9, using the badge's coordinate system. The enlargement on the right shows that the two areas are fairly similar (in this case), except for a small rotation relative to each other. The shaded area represents the image captured during the badge making process.

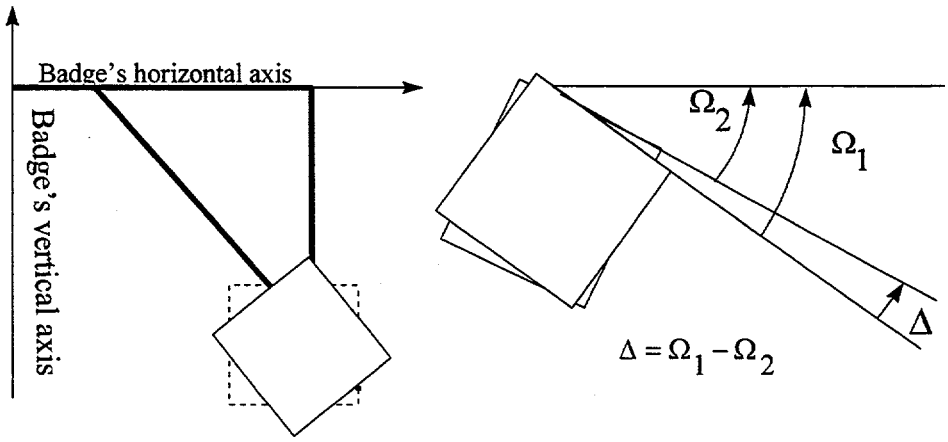


Figure 9

Sub-Image Rotation

For the stored sub-image to be compared properly, it must be rotated by the amount Δ, as shown above. Or, the new sub-image must be rotated by negative Δ, before comparison to the stored sub-image. This rotation is performed numerically, not physically. Since the amount of relative rotation could only be as much as four degrees, compensating for rotation was initially questioned as a necessary step. The effects of compensation are

included as a factor for consideration in the Summary Table at the end of this report. It shows that compensating for rotation improves performance considerably, by increasing the correlation value for images that should match, and by decreasing the correlation value for images that should not match.

Details of the Rotation Algorithm

There are several methods that could be used to perform rotation numerically. The image could be transformed from the rectangular coordinate system to the polar coordinate system, rotated and then transformed back again. Or a 2-dimensional interpolation method could be used, like bilinear or bicubic interpolation. A simple 2D linear interpolation, with automatic image "cropping", was used. The sub-image's center of rotation is the "pixel" (picture element, or matrix element) located at coordinate (j,k), as in Figure 8. However, since the sub-images all have an even number of rows and columns (m and n, respectively), the pixel at the midpoint was arbitrarily defined as being at coordinates (m/2 + 1, n/2 + 1), as illustrated below. Further, since the pixel has height and width, the point of rotation is actually in the middle of that pixel.

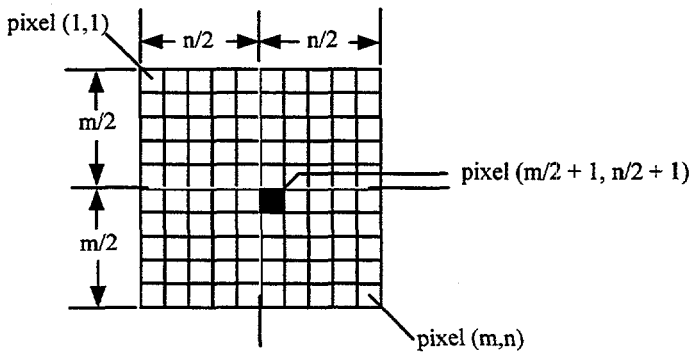


Figure 10
Image Midpoint Definition

Since a pixel must be placed at discrete locations, it is not wise to consider where a pixel goes when the image gets rotated, because the new location will almost always be at a position that is not discrete. Instead, it is better to consider a new rotated image of the same size, and try to find out where each element of that image came from. Figure 11 illustrates an example in which a new pixel, at coordinates (-3,5), in the rotated image is formed by a combination of pixels from the un-rotated image at coordinates (-2,5), (-2,6), (-1,5) and (-1,6). But how much does each of these four pixels contribute to the new pixel? Visually, it seems obvious that the contribution of the pixel at (-2,5) is the ratio of the overlapping area labeled A, to the area of the entire pixel; and likewise for the other three pixels. Since the area of a pixel is one, the ratio of areas reduces to just the area of overlap. If the intensity of the pixel at coordinates (-2,5) is represented as I(-2,5), then the intensity of the new pixel at coordinates (-3,5) is:

$$I(-3,5) = A \bullet I(-2,5) + B \bullet I(-2,6) + C \bullet I(-1,5) + D \bullet I(-1,6)$$

To find the values for A, B, C and D, simply consider the non-discrete situation. If non-discrete pixel locations were possible, it could be determined by simple geometry that the new pixel came from x,y-coordinates (-1.109, 5.725), which is nearest to pixel (-1,6). The distance to pixel (-1,6) is 0.109 in the x-direction, and 0.275 in the y-direction; so the overlap is 0.891 by 0.725, for an area of 0.646, which is the value for D. The overlap in area A is 0.109 by 0.275, for an area of 0.03; area B is 0.109 by 0.725, for an area of 0.079; and area C is 0.891 by 0.275, for an area of 0.245. The total of the four areas is, of course, one.

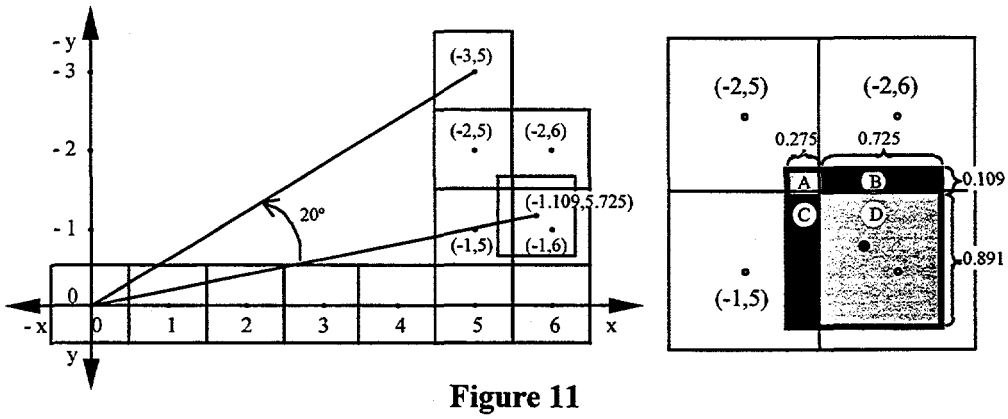


Figure 11
Sub-Image Rotation and Resampling

This set of calculations should be performed for each pixel in the new rotated image, but some problems arise. Since image data is treated as a matrix, the new rotated image must be square or rectangular. As illustrated in Figure 12, there are only three different ways to rotate an image and end up with a square or rectangular result. One way is to use the vertical maximum of the rotated image as the top, the vertical minimum as the bottom, the horizontal maximum as the right limit and the horizontal minimum as the left limit of the new image rectangle (at the left of Figure 12). In this case, the new image is larger than the original, but more importantly, there are areas of the new image for which no source data may be available. Another way is to partially crop the image so that it is the same size as the original. This time, some of the original image data is unused, and again there are areas of the new image for which no source data may be available. (This data is unavailable only if the stored image from the badge making process is rotated.) The third way is to completely crop the rotated image so that more of the original image data is unused, but so that there is no part of the new image that requires source data outside the area of the original image. Notice, at the right of Figure 12, that all the shaded areas are unused parts of the original image. The borders of the new image are delineated by two vertical and two horizontal lines which, for simplicity, are found using the points where the borders of the rotated and un-rotated images intersect.

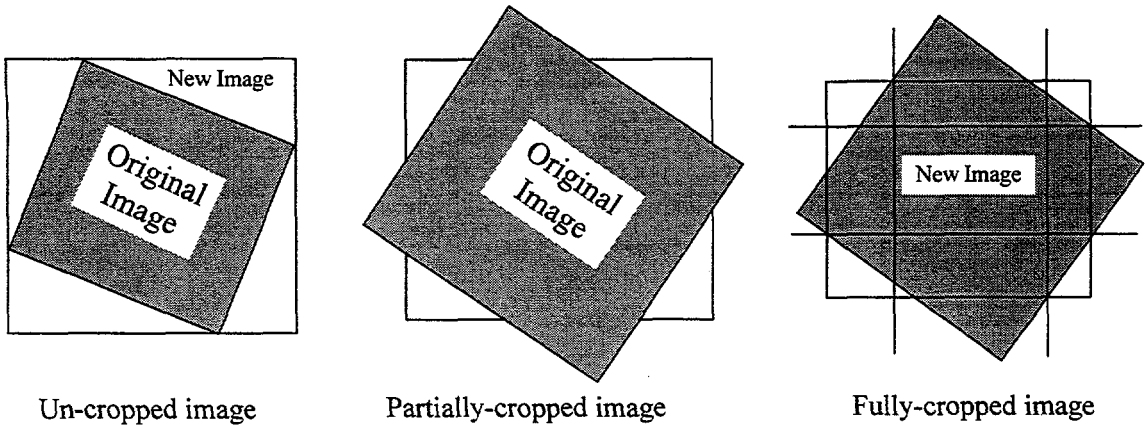


Figure 12

Image Cropping to Resolve Alignment Differences

Although the fully-cropped new image in Figure 12 appears much smaller that the original image, in practice there is very little difference due to the limited relative rotation possible. If the maximum rotation possible is ± 2°, then the maximum relative rotation possible (from badge making to badge reading) is ± 4°. A 100-by-100 image would only be cropped to a 92-by-92 image, with 4° relative rotation, using this method.

By fully cropping the rotated image, every pixel in the new image is formed by a combination of pixels from the original image. If that were not the case, a value would have to be arbitrarily assigned to some pixels, as in the first two methods - typically, zero is chosen. This works fine in some cases, such as simply displaying the rotated image, but it is quite detrimental when the data is to be used for comparison. As mentioned earlier, valid data could be calculated if rotation was performed on the new images, instead of rotating the stored images from the badge making process. However, as described in the next section, the matching algorithm makes several comparisons in the area where the match is expected, so rotating the new images would greatly increase the amount of calculations. (Either one large sub-image area would have to be rotated, or a normally-sized sub-image would be rotated for every offset position tested by the correlation algorithm.) Since the amount of data loss is so small, to reduce calculation time, it is the stored images that are rotated and cropped, with no significant effect to the comparison algorithm. The Summary Table at the end of the report has already factored the effects of cropping into the performance analysis on rotation.

One last note about the subject of rotation: it is probably obvious from Figure 11 that the area of the conceptual pixel at coordinates (-1.109, 5.725) should be rotated, and not aligned to the x,y-axis, as shown. The linear interpolation method used is indeed an approximation, and not as accurate as bicubic interpolation, for example. However, for simplicity and speed, the approximation is certainly an acceptable compromise.

Badge Identification Procedures

To initiate the badge identification process, a badge is placed in the ESID badge reader, and the system acquires images from the front and back of the badge, and then stores those images in memory. The data that was optically encoded on the back of the badge is decoded and provides the identification algorithm with 9 sets of data: 4 DCT matrices, four sets of location coordinates, and an angle that specifies the orientation of the badge during the badge making process. The identification algorithm first decodes (or un-digitizes) the DCT coefficients by subtracting the constant 128, and then multiplies each DCT matrix by it's respective equalization filter (as before, this is an element-by-element vector product; not a true matrix multiplication).

Then, the DCT matrices are simultaneously inverse-transformed and interpolated to regenerate images the same size as they were when originally captured. Just as the "reduced matrix method" forward-transformed and decimated in a single step with fewer calculations, an "interpolating matrix method" performs the opposite function while reducing the amount of calculations, as compared to traditional methods. The badge reader now has images in memory that are very similar to the sub-images that were manually segmented during the badge making process: a left eye, right eye, nose and mouth. They have the same dimensions as the original sub-images, and they "look" the same, except for differences in scale and offset.

These sub-images are almost ready for comparison with the badge photograph, but first they need to be adjusted for the current badge orientation. The system searches the image on the front of the badge for four fiducials, and calculates the orientation angle. If the magnitude of the angle is greater than a predetermined threshold (somewhere from 2° to 5°), the system rejects the badge immediately. Otherwise, the new orientation and old orientation are subtracted to determine the relative rotation, which is then used to rotate and crop each of the reconstructed sub-images. Finally, before the comparisons can begin, the angle and the location of the primary fiducial are combined with each of the feature location coordinates to determine the new coordinates where each feature is expected to be found in the photograph image.

Details of the Comparison Algorithm

Of the few methods available for comparing two images, "phase correlation" was chosen for the E.S.I.D. project. By treating each image as a vector, and multiplying the two vectors, the length of the resulting vector is a measure of their similarity (see Figure 13). For images, the value of each element in one image is multiplied by each element in the other image; the results are accumulated and the square root is taken. This is similar to finding the length of a vector, except that, when finding length, the vector isn't multiplied by another vector - it's multiplied by itself. When two vectors are multiplied (such as A•B), the length of the resulting vector (P) is a maximum if the two vectors are pointing in the same direction, or have the same "phase" - thus the term phase correlation.

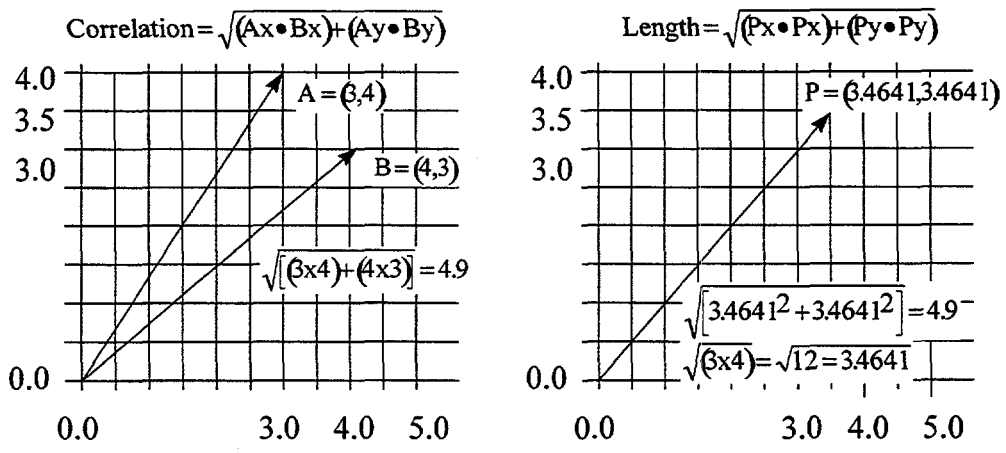


Figure 13
Phase Correlation Illustration

This similarity measure fails, however, when the lengths of the vectors are not taken into account. For example, assume there is a vector C (in Figure 13), that is in the same direction as vector B, but has twice the length. Correlating vectors A and C will produce twice the result as correlating vectors A and B. In some processes, this is advantageous; but if only the phase similarity is desired, the vectors must be normalized as part of the correlation procedure. This also ensures that maximum correlation (when a vector is correlated with itself) is one.

Normalizing the "image vectors" provides insensitivity to variations in lighting, as mentioned previously. If a badge reader has a light source that is stronger than the light source in the badge maker, then the image will have added gain and background light, leading to increased scale and dc-offset (respectively) in the image data. The two vectors shown in Figure 14 have this same problem; vector S is raised above vector T by some dc-offset, and vector T has twice the range of vector S. But both have the same shape, and by the definition of our imaging problem, should compare favorably. If the dc-offset is first removed from each 27-point vector (by subtracting the average value), the resulting vectors will have the same phase (in 27-dimensional space), but one will be twice as long

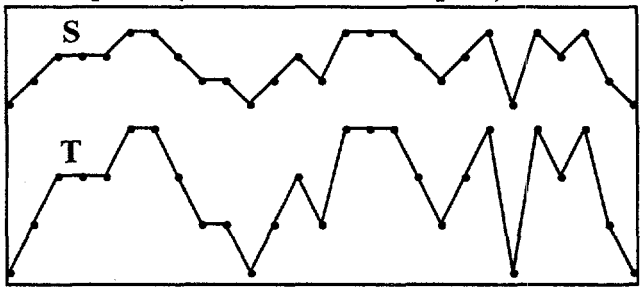


Figure 14
Waveforms with DC Offset and Range Difference

as the other. After normalization, they will have the same length and same phase, and therefore will correlate perfectly.

Image data is phase correlated in the same way as the vectors in Figure 14. A 64-by-64 image with 4096 elements, for example, is treated as a vector in 4096-dimensional space. The 2D dimensionality of the data arrangement is irrelevant, since each element is considered independently of the other data. Any two images being compared first have their dc-offsets removed, and then their vector lengths are normalized to unity. These dc-normalized matrices are then cross correlated. The following equations formally define the dc-normalization procedure for any matrix A, of dimensions M-by-N. Below that are the equations for finding the phase correlation of any two dc-normalized matrices Á and Ú. (Note that taking the square root of the double summation is eliminated for simplicity.)

$$\text{mean} = \frac{1}{M \bullet N} \sum_{i=1}^{M} \sum_{j=1}^{N} A_{ij} \qquad \hat{A}_{ij} = A_{ij} - \text{mean}$$

$$\text{length} = \left[\sum_{i=1}^{M} \sum_{j=1}^{N} (\hat{A}_{ij})^2 \right]^{0.5} \qquad A'_{ij} = \hat{A}_{ij} + \text{length}$$

for i = 1 to M, and j = 1 to N

$$\text{correlation} = \sum_{i=1}^{M} \sum_{j=1}^{N} A'_{ij} \bullet U'_{ij}$$

Correlation Search Area

The encoded coordinate and original rotation data are combined with the new rotation and fiducials information to predict where a feature is expected to be found in the new image, during the badge identification process. The error in this prediction should be no more than ±2 pixels. To be on the safe side, a search area of ± 4 pixels vertically and ±4 pixels horizontally was used, covering a 9-by-9 block of 81 pixel locations. The search can be visualized as taking a transparency with a grid of data on it, and overlaying it on a larger grid of data. Check the overlay to see how well it fits the grid underneath it, using the correlation method. Then slide the overlay to the left by one unit of the grid and correlate again. Slide the overlay left or right up to 4 units, then up or down up to 4 units, over all possible combinations. The result is a 9-by-9 matrix, whose elements indicate the degree of correlation at each possible offset location. The position of the largest element in the correlation matrix indicates the offset that gives the best fit between the two images. To preserve information about the relative positions of the facial features, all four features are correlated as a group - that is, all are shifted by the same offset and a combined correlation is calculated, producing a single correlation matrix, instead of four. That way if two people have the same eyes, but one person's eyes are farther apart, they won't correlate as well as if the features were correlated independently.

Unfortunately, a 9-by-9 search area means that the above correlation calculations must be performed 82 times. Since the stored reference image doesn't change, it only needs to be dc-normalized once, but at every one of the 81 test locations, the new image must be dc-normalized. (Ú is the dc-normalized reference image, and Á is one of the 81 dc-normalized images being tested for similarity to the reference image). This is a lot of number crunching and slows the badge identification process. Calculating the vector length, adjusting for length and calculating the correlation, each require M•N multiply/divide operations. And all but the length adjustment require M•N add/subtract operations. The four features cover 46,000 pixels, so just calculating the cross correlation once requires effectively 138,000 multiplies and 184,000 additions. This must be done once for the reference image and once for each of the 81 locations to be tested, for a total of 11.3 million multiplies and 15.1 million additions.

This is obviously the bottleneck in system performance. It could be improved by reducing the search area, but the cost would be degraded performance - most likely, a legitimate E.S.I.D. badge would fail the comparison test. But if the fiducials were improved (as suggested earlier), reducing the search area would be recommended, and the badge identification process would speed up. As it stands, however, the 9-by-9 search area is considered to be optimal, allowing for some sloppiness in the feature locating process, yet limiting the number of calculations from becoming unreasonable.

Some improvement in performance was achieved using a two-step correlation algorithm. This method takes advantage of the fact that the 9-by-9 correlation matrix is always a "smooth" surface - that is, there is very little difference between adjacent elements. This is a consequence of the fact that the images themselves are smooth; that is why low-pass filtering the images (decimating the Discrete Cosine Transform) has almost no effect on them. If the correlation matrix is calculated only at every other vertical and horizontal position (as illustrated by the shaded areas at the left of Figure 15), a "course resolution" matrix is generated after only 25 correlations. The second step of the algorithm fills in the correlation matrix only near the largest of those first 25 correlations (shown as a black

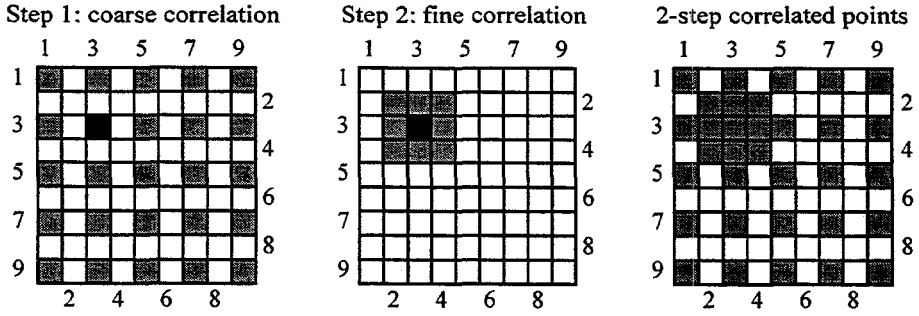


Figure 15
Iterative Correlation Procedure

square). This adds 8 more correlations to the process, for a total of 33 correlations. This runs almost 2-and-a-half times faster than calculating all 81 correlations. It also effectively increases the search area to 11-by-11, since it can look for a match up to 5 pixels away from the expected location, depending on the outcome of the coarse search.

The Decision Method

The largest of the 33 correlations is assumed to be the largest value in the 9-by-9 correlation matrix. At this point, only the value itself is important - not it's location. It might seem logical to use the distance from the expected location of the feature set to the peak correlation point as a "weighting factor" to be used in the pass/fail decision. However, this inherently asserts that a strong match to a feature set that is farther away should be as valid as a poorer match that is right where expected. But to the contrary, this should actually be considered evidence of tampering with the E.S.I.D. The point of the identification process isn't to determine if a specific person's face is pictured on the badge; it's to determine if the photograph of that person has changed since the badge was made. So the position of each of the features is extremely important, and moving should not be tolerated.

A badge passes or fails based solely on the peak correlation value. An acceptance level of 0.85 was determined empirically from cross-correlations of 28 images. This is about halfway between the lowest "low match" (0.9047) and the highest "high fail" (0.7377), in the Summary Table (bottom section). A peak correlation above 0.85 passes the identification process, and one below that fails.

Summary

Assessment of the Process

A set of 28 images was provided for testing the identification algorithms. There were three images of each of 10 subjects (two images were ruined in the data transfer process). The images of each subject were taken from the same photograph, but had changes in position, rotation and lighting - to mimic the effects of using various badge readers. A feature set was extracted from each image, and all 28 feature sets were cross-correlated with all 28 images. The Cross-Correlation Table below illustrates how the images should correlate. The black squares indicate auto-correlation (self correlation), the shaded squares indicate the images that should cross-correlate well, and the white squares indicate the images that should not cross-correlate well.

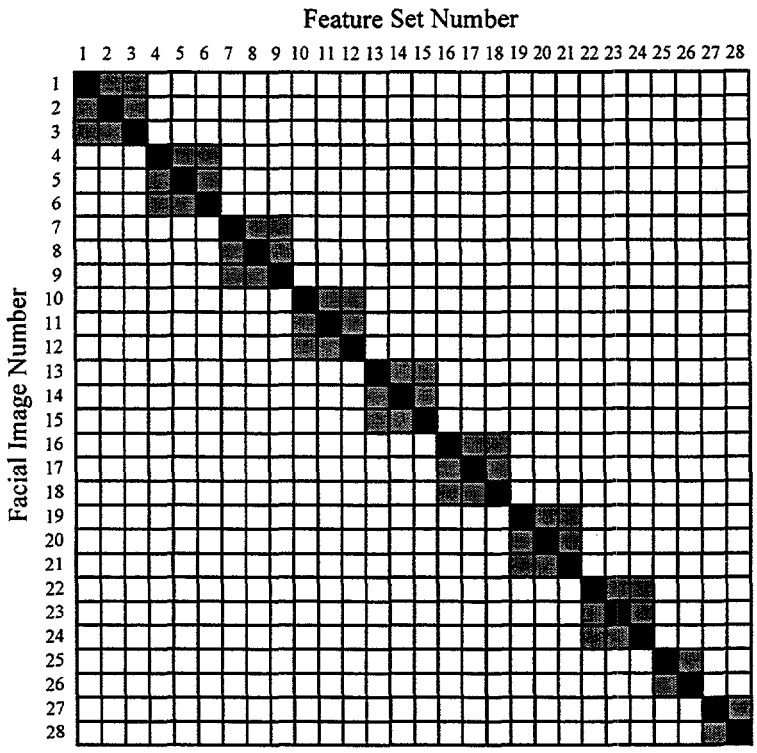


Figure 16
Cross Correlation Table

Rather than fill 18 Cross Correlation Tables with data, for each of the 18 test cases, the Summary Table lists only the important data. Of the 18 auto-correlations, only the lowest value is of real interest; it is called "Low Auto" in the Summary Table. Likewise, of the 52 cross-correlations that should match, the lowest value called "Low Match" is given. Conversely, of the 704 cross-correlations that should fail to match, only the highest value (called "High Fail") is of interest. Finally, the "Headroom" values give the distance between High Fail and Low Match. The more headroom there is, the less chance there is for a false match or a false fail. This test includes only 28 images, but in a larger test set it is very likely that the Low Match value would get lower, and even more likely that the High Fail value would get higher. So for this test set it is important for the Headroom to be as large as possible.

The Summary Table includes 18 variations of four parameters, so that the effects of those parameters can be understood individually. The compression parameter C indicates whether the feature set sub-images were compressed or not. Similarly, the quantization parameter Q indicates if the feature set sub-images were quantized (quantization includes equalization). And parameter A indicates whether the feature set sub-images were adjusted for the effects of rotation (rotating and cropping each sub-image). Only six of the eight possible permutations of those three parameters were tested; it doesn't make sense to have quantization without compression, so that permutation is omitted. The last

Summary Table

Parameter codes: C = Compression; 0 = no, 1 = yes
Q = Quantization; 0 = no, 1 = yes
A = Adjustment for rotation; 0 = no, 1 = yes
R = Rotation; 0 = no, 1 = +4°, 2 = - 4°

Group #	Line #	Parameters C	Q	A	R	Low Auto >>	Low Match >>	High Fail <<	Headroom >>
1	1	0	0	0	0	1.0000	0.9042	0.7368	0.1674
	2	0	0	0	1	0.8793	0.8340	0.7187	0.1153
	3	0	0	0	2	0.8874	0.8350	0.7363	0.0987
2	4	0	0	1	0	1.0000	0.8997	0.7320	0.1677
	5	0	0	1	1	0.9598	0.9040	0.6904	0.2136
	6	0	0	1	2	0.9682	0.9026	0.7114	0.1912
3	7	1	0	0	0	0.9842	0.9086	0.7425	0.1661
	8	1	0	0	1	0.8934	0.8466	0.7235	0.1231
	9	1	0	0	2	0.9018	0.8462	0.7420	0.1042
4	10	1	0	1	0	0.9842	0.9047	0.7376	0.1671
	11	1	0	1	1	0.9633	0.9091	0.6969	0.2122
	12	1	0	1	2	0.9690	0.9076	0.7171	0.1905
5	13	1	1	0	0	0.9841	0.9087	0.7427	0.1660
	14	1	1	0	1	0.8931	0.8459	0.7235	0.1224
	15	1	1	0	2	0.9019	0.8460	0.7421	0.1039
6	16	1	1	1	0	0.9841	0.9047	0.7377	0.1670
	17	1	1	1	1	0.9634	0.9091	0.6972	0.2119
	18	1	1	1	2	0.9689	0.9078	0.7174	0.1904

>> means this number should be as big as possible, but the smallest is given.
<< means this number should be as small as possible, but the biggest is given.

parameter R (for rotation) has three different cases, which are possible for each of the other six permutations, for a total of 18 permutations. The table is arranged in groups of three, in which only the rotation parameter changes.

The rotation parameter is a little different from the other parameters. The small differences in rotation between the three images of the same person were not enough to fully test the effects of rotation on identification, so a special test was performed in which the 28 facial images were rotated +4° (and also -4°), and the complete set of correlations were run. (The feature set sub-images were not rotated, nor was a new feature set collected.) The effect on the Low Auto value is particularly interesting, because this shows how well the rotation functions work on images that should auto-correlate perfectly. For example, in line 2 of the Summary Table, the effects of compression and quantization are removed, and without adjusting for rotation, the auto-correlation of a +4°

rotated image with it's own un-rotated feature set produces a Low Auto value of 0.8793. But in line 5, with adjustment for rotation, Low Auto is 0.9598, which is much better.

This improvement is also evident when the images are rotated -4°, and when compression and quantization are taken into account; as can be seen when comparing lines 3 vs. 6, 8 vs. 11, 9 vs. 12, 14 vs. 17 and 15 vs. 18. It is also noteworthy that adjusting for rotation, when there was no rotation, did not cause problems, which can be seen by comparing Low Auto from lines 1 vs. 4, 7 vs. 10 and 13 vs. 16. Furthermore, adjusting for rotation significantly improves almost all the other parameters too! Notice that the most important parameter, Headroom, always increases (compare lines 1 vs. 4, 2 vs. 5, etc.). The High Fail levels are always lowered after adjusting for rotation, and the Low Match levels are almost always higher; except in cases when there was no rotation to adjust for. Overall, adjusting for rotation was one of the most useful tools in the identification process.

While the expressed purpose of adjusting for rotation is to improve correlation, compression and quantization have unrelated benefits, and may actually degrade correlation performance. To determine the effects of compression alone, comparison is done on groups 1 vs. 3 and 2 vs. 4. The overall effect of compression can best be seen by a comparison of the Low Auto values of lines 1 and 7, where the auto-correlation should be a perfect "one", and compression is the only difference. In line 7, the auto-correlation suffers a minor loss of 0.0158 from compression. Interestingly, however, the cross-correlation values mostly increase, due to the fact that compression filters out the differences between images. The increase in Low Match values is desirable, but the increase in High Fail values is not. And overall, there is a reduction in the critical Headroom values. It was expected that compression would have some cost, but it has proven to be small enough that the cost does not outweigh the benefits.

Finally, the effects of quantization can be viewed by comparing groups 3 vs. 5 and 4 vs. 6. Again, the most direct comparison can be seen by comparing the Low Auto value of lines 7 and 13, where the only difference is quantization, and there is no rotation or adjustment for rotation to obscure the results. The drop in auto-correlation due to quantization is only 0.0001. The biggest drop in all 6 Low Match cross-correlations, due to quantization, is only 0.0007, the biggest increase in the High Fail cross-correlations is 0.0003, and the biggest effect on Headroom was to lower it just 0.0007. These effects are insignificant and help to illustrate the value of the equalization filters. Although there was no data taken without the equalization filters, it was visually obvious that without them there was a significant amount of random noise in the inverse-transformed images.

The Summary Table was also useful for setting the threshold used to decide whether a badge passes or fails. Since the final system does include compression, quantization and adjustment for rotation, only the data in group 6 was considered. The lowest of the three Low Matches is 0.9047, and the highest of the three High Fails is 0.7377, for a Headroom of 0.1670. If the acceptance level was set halfway between the two, at 0.8212, this would provide an equal margin of error (8.35%) in both directions. However, given a larger set

of data, it is very likely that there will be badge photographs which are different, but just happen to cross-correlate at a level higher than 0.7377. On the other hand, it is not expected that there would be a significantly lower Low Match value than 0.9047, no matter how much bigger the data set gets. This is because the Low Match correlation value is the result of changes in rotation and lighting, which has been pretty thoroughly tested in the worst cases. Therefore, without testing a larger sample of images, an acceptance level of 0.85 is recommended.

Data Authentication and Key Control

Cryptographic authentication of the data printed on the identification card is a key element in the counterfeit resistance of this system. If the validation process indicates valid data, the user can be assured that the document was issued by an authorized facility and that the data has not been altered after the issuing process. This prevents a potential counterfeiter from producing fake documents or altering data. The cryptographic algorithm would have to be broken or the keys discovered in order for a counterfeiter to be successful.

The data could be authenticated by encryption or authentication algorithms. Both were considered. The encryption process uses a key to make the data unreadable to anyone who does not possess the key required for decryption. Authentication algorithms leave the data intact, but add a much shorter message authentication code (MAC) or tag that is calculated from the data and a secret key. This MAC or tag can be used by anyone possessing the validation key to verify that the data is valid. In general, authentication is considerably faster than encryption for large blocks of data.

Encryption would serve as an effective authentication tool in this application, but the data would have to be decrypted before it could be used to verify the validity of the picture and other information on the document. In fact, every block of data must be decrypted, requiring a great deal more computational time than would be required for an authentication algorithm. This would be cumbersome in some applications because of the time required for the decryption process. For example, in point-of-sale applications, only the lower level process of reading the name and account number would be adequate for the majority of customers. The full verification process would probably only be used on a random basis.

The use of an authentication algorithm allows the data to be read directly, but adds a number of bytes to the data that must be stored. The number of bytes added depends on the algorithm used. RSA Data Security's public key authentication algorithm[6] adds 15 bytes, and the Digital Signature Standard (DSS)[7] adds 20 bytes.

Another selection to be made is whether to use a private key or to use public key algorithm. Private key algorithms execute much faster, but only users who have access to the secret key could use the authentication feature of the system. The same secret key is used to verify the validity of the data as is used to sign the data by calculating the

message tag. Therefore, anyone who can verify the validity of a document could also issue them. A public key approach uses different keys for the signing and verifying processes. This allows all users to have access to the verification key (the public key) while the signing key is kept secret. Calculating the secret key from the public key is termed a "difficult problem," requiring many years of time on the largest arrays of computers available to an adversary.

The secret key can be generated in the computer used for signing the documents when they are originated. No human being needs to know the secret key. This eliminates a major vulnerability present in all secret key cryptographic systems. It also greatly simplifies the key distribution problem, since disclosure of the keys to unauthorized individuals is not of concern. In fact, the only security concern in the key distribution is to ensure that no unauthorized keys are added to the key list. If an adversary can substitute his own keys into the system, he can then generate bogus documents that will be accepted.

Each authorized issuing station will have its own pair of secret and public keys. These keys would be used for a period of time, probably on the order of a few years, and then changed. The public keys of the authorized issuing stations would be distributed to the validation stations electronically, using either phone lines or the Internet. These key transmissions would be authenticated using a public key authentication algorithm. The keys for the transmission authentication will be transferred using a separate channel, such as mailing in tamper indicating packages or publication in a public journal. The process must protect against an adversary substituting his own keys for these transmissions. If he can substitute his own authentication key for validating key transmission, he can defeat the system by distributing his own key lists.

The Digital Signature Algorithm was selected for the authentication process for this application. This selection over other equally well suited candidates was mainly due to the ready availability of a royalty-free, extremely fast implementation for the Intel processors used in personal computers. This implementation was developed by a group of data authentication specialists in the Cooperative Monitoring Systems Department at Sandia National Laboratories.

A key length of 1024 bits was selected. This should provide adequate security to be secure from cryptographic attack for considerably greater than the expected useful life of any document produced by the system. If great strides are made in calculating the secret keys from the public keys, this key length can be easily increased in the future.

Data Storage

A considerable amount of data must be stored in order to verify that a picture has not been altered to look like a person other than the one for whom the identification document was issued. The badge application was chosen to demonstrate the feasibility of the system partly because it is the most limiting with respect to the amount of space available for data storage.

This required a method for storing data on the badge that has a much higher data density than is possible for either magnetic stripe or traditional bar code. The least expensive form of storage for the authentication data on the document is to print the information using a two-dimensional code similar to Code 1 or PDF-417[8]. The total data storage requirement, when coupled with the small area available and the available printing technology, does not allow these existing technologies to be used directly. We are therefore developing our own two-dimensional code for this application. Using this type of data storage technology allows us to produce badges that are not significantly more expensive than those in use today. Figure 17 shows the proposed data array.

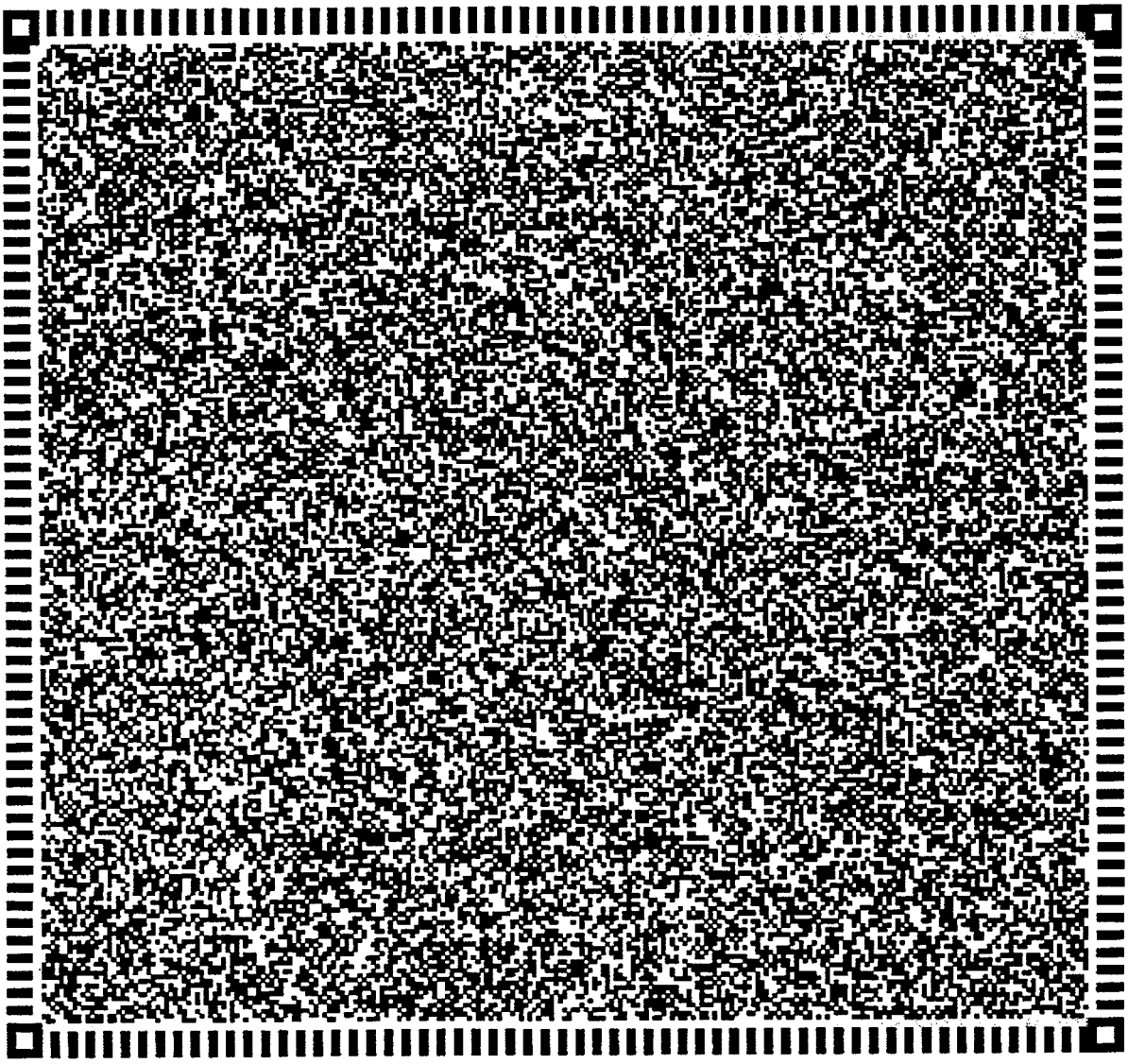


Figure 17
Two Dimensional Bar Code
(enlarged to show detail)

In the data array shown, each black square corresponds to a one in the data. Conversely, a white square denotes a zero. There are a total of 57,536 data storage locations arranged in a 240 by 224 position array. This allows a total data storage of 7192 bytes. The size of the actual printed data array is 1.71 inches by 1.60 inches.

The array is printed using a laser printer at a resolution of 300 dots per inch. Each data square is made up of four printer dots. This allows the laser printer to print each data square as a true square. Trying to print with less than four dots per square results in a loss of sharpness that could lead to reading errors. As laser printer technology continues to improve, more dense patterns may be possible. It may be possible to shrink the data array to 1.28 inches by 1.20 inches in the near future.

The data is read using a video camera and a frame grabber that captures a 512 pixel by 480 pixel image. Each data square is covered by four image pixels. This ensures that at least one pixel sees only the data square that corresponds to its location in the array. The theoretical limit for being able to resolve such a data array is 2.25 pixels per data square. This limit corresponds to an over sampling of 1.5 times in each direction. Our application uses an over sampling of two times in each direction.

This data array has many features that allow it to be read accurately. In each corner is a hollow square. These squares are used to define the limits of the image and to allow correction for magnification errors. Along each edge is a row of fiducial marks that allow further refinement in the corrections required to find an individual data square. Each of these fiducials is exactly two data locations wide and the spacing between fiducials is equal to their width. The interaction of these fiducials with the corner squares allow the orientation of the data array to be determined without ambiguity.

The following are the minimum data requirements determined for the badge application:

- Name - 20 bytes (thirty three characters stored in binary as base 27 numerals)
- Identification number - four bytes (nine decimal digits)
- Date and time of issue - four bytes (seconds past a reference date)
- Issuing station number - 2 bytes
- Special access identifiers - 2 bytes
- Image data - 2875 bytes

This gives a total data storage requirement of 2927 bytes. The remaining 4265 bytes are available for error detection and correction code. The error detection and correction codes were not developed for this application due to a lack of resources. This is another area requiring further development.

The original project plan called for design of the data correcting code system. Unfortunately, the level of effort required for image processing and data compression was

significantly above our original estimates. We therefore had to divert the resources that would have been used to design the data correcting code to the other, higher priority, task.

We had also planned to test the data reading system with newly made badges before and after laminating and also after they had been used for a period of time. The budget allocated to our project was adequate to prepare for this testing, although the actual test results would not have been available until shortly after the completion date of the project. The LDRD program office unilaterally reduced the funding for this project before the hardware required to do this testing was completed. The partially completed hardware was scrapped and the testing plan was abandoned.

Conclusions and Recommendations

Biometrics and other random patterns combined with public key data authentication can be used to increase the security of identification cards, passports, and other similar documents when the cost of this increased security is justified by the potential cost of counterfeits. As the cost of the authentication equipment continues to decrease, more applications will become cost effective.

The badge application that was demonstrated in this study shows that the technology exists for making secure identification documents using this approach. More forgiving applications, such as passports, will be much easier to accomplish. If other biometrics, such as fingerprints or hand geometry, are used the amount of data that must be processed and stored is reduced significantly. The process of matching the person to the document would also be much less subjective, although additional equipment would be required.

The methods for compressing and comparing image data appear to work quite well. The algorithms are robust and provide a significant separation between the images that should match, and those that should not. At this point, only the statistical significance of the data is in question. It seems obvious that only 10 photographs, and a total of 28 variations of those photographs, is not enough data on which to declare the success of the project. A statistical analysis of the data should be performed to provide us with a level of confidence in the data we have and/or provide us with the sample size needed to achieve the level of confidence that we would like to have.

It should be reiterated that the Summary Table includes only "worst-case" data. As good as the results appear, they are in fact even better. Of the 10 original photographs used, two of them were so similar that they skewed the results and almost solely contributed to the worst-case values in the Summary Table. In spite of the small sample size, this was indeed a stringent test of the process, and the results should not be criticized too harshly.

System performance could be improved, however, in two ways. First, as mentioned previously, the fiducial marks are not optimal, and should be replaced with crosses or Xs or some other marks that could be easily, accurately and consistently found. And second,

the sub-image capture during the badge making process needs to be automated. Although the eyes, nose and mouth are probably the most identifiable areas in a facial photograph, it is a very compute-intensive process to automatically locate those features. Instead, it might be best for the computer to simply look for areas of the image that have the highest variance, or standard deviation, as this is what typically produces high auto-correlations and low cross-correlations. Whatever the method, it should be automated to standardize the process, and reduce the time required to make a badge.

Public key data authentication can certainly be used for the purpose proposed here. This is a mature technology that is well suited for this type of application.

It is indeed unfortunate that we were unable to test the data storage system. The two dimensional code proposed seems to be adequate, but tests with actual readers are required before making a final decision. Further work will be required to make the two dimensional code described here practical. However, as stated above, this may not be necessary if smart cards with large data storage capacities become more affordable.

Systems based on this concept could be fielded in the relatively near future. Additional work would be required to assess system vulnerabilities specific to the application. Further development might also be required to adapt the concept to the application. Reader/authenticator costs would be on the order of $1600 if a relatively small number of units are built. If a few hundred are built, these cost could decrease by approximately fifty percent. If several thousand are built, the cost per unit might be in the two to three hundred dollar range. The cost of the identification documents would remain essentially unchanged. These cost estimates do not include development costs, which could be considerable.

References

[1]*Counterfeit Deterrent Features for the Next-Generation Currency Design*, National Research Council, National Academy Press, Publication NMAB-472, 1993, pp. 74-75.
[2]ibid., p. 14.
[3]K. M. Tolk, "Reflective Particle Technology for Identification of Critical Items," *Proceedings of the 33rd INMM Annual Meeting,* Orlando, Florida, 1992, pp. 648-652.
[4]ibid.
[5]D. Jeffreys, XTEC Incorporated, personal communication, October, 1992.
[6] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography,* 433-438, CRC Press, 1996.
[7] FIPS 180-1 reference "Specifications for a Digital Signature Standard," *Federal Information Processing Standards Publication XX,* National Institute of Standards and Technology; August 19, 1991.
[8]"Two-Dimensional Bar Code Symbologies," *Scan-Tech 91 Proceedings,* Dallas, Texas, 1991, Session 3D.

Distribution

1	MS0449	T. J. Draelos, 6513
1	MS0449	V. A. Hamilton, 6513
1	MS0449	M. M. Jimenez, 6513
2	MS0619	Review & Approval Desk, 12698 For DOE/OSTI
1	MS0655	J. Bartberger, 5736
1	MS0656	J. C. Matter, 5749
10	MS0656	K. M. Tolk, 5749
1	MS0769	D. S. Miyoshi, 5800
5	MS0899	Technical Library, 4916
1	MS0970	J. R. Kelsey, 5700
1	MS1436	LDRD Office, 4523
1	MS9007	R. C. Wayne, 8400
1	MS9018	Central Technical Files, 8940-2
1	MS9101	W. C. Peila, 841
5	MS9101	R. M. Bell, 8411
1	MS9101	P. T. Larson, 8413
1	MS9106	G. L. Simpson, 8417