Embedded Media Barcode Links: Optimally Blended Barcode Overlay on Paper for Linking to Associated Media

Qiong Liu, Chunyuan Liao, Lynn Wilcox, Anthony Dunnigan
FX Palo Alto Laboratory
3400 Hillview Avenue, Bldg.4
{liu, liao, wilcox, tonyd}@fxpal.com

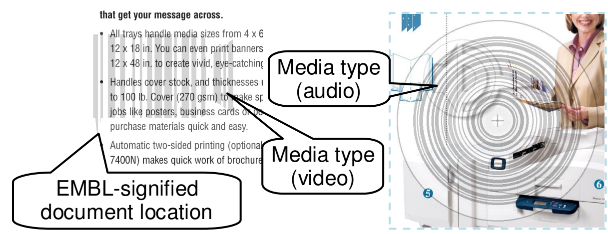
ABSTRACT
Embedded Media Barcode Links, or simply EMBLs, are optimally blended iconic barcode marks, printed on paper documents, that signify the existence of multimedia associated with that part of the document content (Figure 1). EMBLs are used for multimedia retrieval with a camera phone. Users take a picture of an EMBL-signified document patch using a cell phone, and the multimedia associated with the EMBL-signified document location is displayed on the phone. Unlike a traditional barcode which requires an exclusive space, the EMBL construction algorithm acts as an agent to negotiate with a barcode reader for maximum user and document benefits. Because of this negotiation, EMBLs are optimally blended with content and thus have less interference with the original document layout and can be moved closer to a media associated location. Retrieval of media associated with an EMBL is based on the barcode identification of a captured EMBL. Therefore, EMBL retains nearly all barcode identification advantages, such as accuracy, speed, and scalability. Moreover, EMBL takes advantage of users' knowledge of a traditional barcode. Unlike Embedded Media Maker (EMM) which requires underlying document features for marker identification, EMBL has no requirement for the underlying features. This paper will discuss the procedures for EMBL construction and optimization. It will also give experimental results that strongly support the EMBL construction and optimization ideas.

Categories and Subject Descriptors
H.5.1 Multimedia Information Systems: Artificial, augmented, and virtual realities. H.5.2 User Interfaces: Interaction styles.

General Terms
Algorithms, Design, Documentation, Human Factors.

Keywords
Augmented paper, barcode, camera phone, document recognition, marker on paper, vision-based paper interface.

1. INTRODUCTION
Although paper is one of the most widely used devices for viewing information, it cannot play dynamic media such as video and audio. On the other hand, cell phones are increasingly used to play audio and video but cannot match paper's high resolution, large display size, flexibility in spatial organization, outdoor-readability, and robustness for static content. It is now possible to combine the two, using image recognition technology to link paper documents to corresponding dynamic media. A cell phone camera is used to capture an image of a document patch. The document patch is identified using features in the image, and digital media linked to that location in the document is retrieved and then played on the cell phone.



(a) (b)

Figure 1. EMBL Examples.

A common method for creating this type of media link on a paper document is to print markers on the document. One obvious example is barcode [13]. However, existing barcode printing requires an exclusive space and thus may interfere with the document content layout. Microsoft Tag [22] alleviates this issue by merging data cells with the user-specified image background, but still requires an opaque black-white border for the reader to locate the data cells. DataGlyphs [16] overcome these problems by printing a nearly invisible machine-recognizable pattern on the paper. However, this type of marker requires high resolution printers for printing and high resolution cameras for query image capture. Electronic markers like RFID can also be used [18], but this approach increases the production costs.

Other systems compute features of the document content itself for identifying the document patch and thus creating a media link. HotPaper [2] and Mobile Retriever [9] use features based on document text such as the spatial layout of words. Other systems such Bookmarkr [6] and MapSnapper [4] use pixel level image features, such as the SIFT [10] algorithm, to recognize generic document content such as pictures and graphic elements. With these systems, exclusive spaces are not required for marker printing.

Both marker-based methods and document-appearance-based methods fall short in providing visual guidance for users. Although bar codes and Data Glyphs are visible, they do not directly indicate the existence or type of media associated with them. When appearance-based feature are used, there is no on-paper indication at all to the user that there is media linked to the document. As a result, a HotPaper [2] user has to pan a camera

ICMI-MLMI'10, November 8-12, 2010, Beijing, China.

phone over the paper document to look for hotspots until feedback such as a red dot or vibration is presented on the cell phone.

To solve this problem, researchers augment paper with a meaningful awareness-mark, called Embedded Media Marker (EMM) [5], which indicates the existence, type, and capture guidance of media links. On seeing an EMM, the user knows to capture an image of the EMM-signified document patch with a cell phone in order to view associated digital media. This is analogous to Web pages that use underlines, font differences, or image tags to indicate the existence of links that users then click for additional information. Unlike barcodes, EMMs are nearly transparent and thus do not interfere with the document appearance. Unlike Embedded Data Glyphs [16] or Anoto patterns [17], EMMs can be printed with a regular low-resolution printer and identified from an image captured by a normal cell phone camera. Unlike other appearance-based approaches, EMMs clearly indicate signified document patches and locations. The design of EMMs includes an icon that indicates what type of media (e.g. audio, video, or image) is associated with the EMM-signified document location. Furthermore, by requiring the captured image to cover the whole mark, the EMM system can improve feature construction accuracy, matching accuracy, and efficient resource usage. EMM solves many past problems. It also requires the underlying document to have sufficient features for patch identification. Additionally, the scalability and error correction capability for EMMs are still unknown.

To compensate for these issues with EMM, we designed a media awareness-mark, called Embedded Media Barcode Link (EMBL). Figure 1 shows EMBLs printed on paper documents, where the EMBL iconic marks indicate linked video and audio respectively. EMBL is a semi-transparent media-icon-modified barcode overlay on paper document content for linking to associated media. Unlike traditional barcodes, EMBLs are semi-transparent and thus have less interference with the original document appearance. Unlike Embedded Data Glyphs [16] or Anoto patterns [17], EMBLs can be printed with a regular low-resolution printer and identified from an image captured by a normal cell phone camera. It can also indicate the existence, type, and capture guidance of media links. On seeing an EMBL, it will be intuitive for a user to associate it with a traditional barcode and for them to transfer their barcode capture skill to EMBL. Since it uses the same decoding scheme as a barcode, its accuracy, speed, scalability and error correction ability are supported by previous barcode testing experiments. It can also work on documents that do not have many features, such as a blank page with one or two text lines (a difficult case for EMM and many other appearance based approaches). EMBL is still more intrusive than EMM, but it has other advantages and can compensate for documents that are difficult for EMM.

In the next section, we explain various EMBL components and give some EMBL construction examples. We then discuss the detailed requirements for EMBL construction. After the requirement discussion, we formulate the barcode-content-blending problem in EMBL construction as a communication problem. Based on these EMBL construction ideas and the formulation, the EMBL authoring tool and the optimization algorithm for semi-automatic arrangement of an EMBL on a document are described, followed by a series of experiments to verify our hidden hypothesis on the EMBL agent and its optimization process. We end the presentation with an EMBL summary and future work discussion.

2. EMBL CONSTRUCTION OVERVIEW

EMBL is a semi-transparent media-icon-modified barcode overlay on paper document content for linking to associated media. It uses an 'EMBL-signified document location' to define the precise location for media association. An EMBL uses semi-transparent form to reduce interference with original document content and to be closer to an EMBL signified location. It uses a semi-transparent barcode to identify signified document patches, and uses iconic information to reveal associated-media information to the user. The form of an EMBL can be horizontal, diagonal, vertical, or circular. Figure 1 shows two EMBL examples. In these examples, each EMBL uses an iconic mark to reveal information of the linked media type. It also uses an arrow to accurately signify the location of the media link.

2.1 EMBL Construction Ideas and Examples

The barcode outline modification may be achieved by masking an existing barcode with an iconic mask. Figure 2 illustrates the process for modifying the EMBL outline. In this process, we first construct a black and white media icon, a video icon in this example. Then, we mask a barcode with the icon so that the barcode extension pattern changes with the media icon envelop. Before we use this modified barcode in an EMBL application, we also need to verify the masked barcode with the barcode reader. By looking at this example, it is not difficult for readers to extend this idea to other shapes, such as company logos etc.

Figure 2. Left: Media icon mask. Middle: Use media icon mask to modify the barcode outline. Right: Verify barcode using reader. In this example the identification result is shown here for illustration. It will not be printed on a document page for real applications.

Moreover, we may also alpha-blend a media icon directly with a barcode to generate an EMBL. To make sure the EMBL can work in the document, we also need to verify the EMBL base form (without document content blending) with a barcode reader. EMBLs constructed with this approach can reveal more details of a media icon. However, the blending process for this type of EMBL is more complicated than the earlier type. EMBL constructions are not limited to these two types. Figure 1 (b) shows a circular type EMBL overlaid on a document.

The most significant difference between an EMBL and a traditional barcode is the marker's transparency. Traditional barcode standards are machine oriented standards. More specifically, they only consider the barcode itself and barcode readers. The EMBL design considers the user, document, and barcode all together, and every EMBL mark is a proper barcode-content blending that benefits both users and documents without losing barcode advantages.

To facilitate readers' understanding of document contents as well as barcode overlay identification, we need to minimize the cross-interference between the barcode signal and the content signal. Previous research used invisible toner for the barcode so that the content signal and barcode signal use different light spectrums. Even though this approach is good for human reading and barcode decoding, it totally eliminates the visual cue for a user to initiate user-paper interaction. Moreover, it requires printer modification and special invisible toner to support applications of that technology. EMBL addresses this problem with a proper blending of barcode signal and document contents so that the original document content is readable to humans and the barcodes are readable to a barcode reader.

2.2 Requirements of a Media Marker

To link multimedia data to a paper document location, we need a media maker that indicates the existence of the media. The media marker should have the following properties:

1. The marker should act as a prompt for a cell phone capture when a user is interested in the marker signified location.
2. By looking at the marker, it should be easy for a user to figure out the marker capture guideline.
3. The marker should provide information about the linked media so that it is not considered as a price tag or other unrelated items.
4. The marker should be close to the location on the paper document where the multimedia is linked.
5. The marker identification speed should be fast to ensure a short waiting time for the user.
6. The marker should be distinctive enough to ensure media retrieval accuracy.
7. The marker should be scalable to a large number of documents.
8. The marker should not depend on document features so that it can handle all document regions including nearly blank regions and regions that are the same on different documents.
9. The marker should not interference with the original document content and layout.
10. The marker identification process should not require too much disk, memory, and computational cost.

In this list, the 1-7th requirements are for improving the users' experience, the 8-9th requirements are for readability of original document, and the 10th requirement is for reducing hardware and energy cost for a large scale system. The EMBL design aims to satisfy these requirements. Since most users are familiar with barcode and barcode capture with a cell phone, the semi-transparent barcode look of EMBL is a good prompt for cell phone capture and a good icon for generating usage confidence in the general public. The EMBL shape provides media type information to users. Because of its semi-transparent form, it is easy to move an EMBL close to any media associated location without changing the original document layout. Because the EMBL decoding is similar to barcode decoding, the decoding accuracy, speed, and scalability are verified by many early barcode applications. Also because of the semi-transparent form, an EMBL mark has less interference with the original document content and layout than traditional barcode. Its hardware cost is also easy to estimate and affordable for most applications. When EMBL is permitted on document surface via printing or transparent-sticker, it can support nearly all multimedia-paper associations. Besides those applications, it can also work on applications that need to differentiate users, documents, or machines. For example, a salesperson can use EMBL to customize links to the same product brochure based on customers. Different magazines can link different media reports to the same photo. A company may also use EMBL with different barcode to route users of the same media to different regional servers.

2.3 Problem Formulation

The EMBL printing and identification process can be modeled as a communication process where each document patch is the communication channel, the embedded barcode is the transmitted signal, and the document content and iconic marks are the noise for signal transmission. With this model in mind, we can design algorithms to find a proper channel in a user selected neighborhood for barcode transmission with proper content-barcode blending. From the well-known Shannon-Hartley theorem (discussed in a later section), the barcode-content blending (i.e. signal-noise mixing) process enables us to adjust the channel capacity of a document patch. With a reasonably high channel capacity, we assume that a high barcode identification rate can be ensured. This model is used in our EMBL authoring tool to find the best document patch for barcode-printing and a proper barcode-content blending coefficient. It can also be used for EMBL optimization over different barcode types and other parameters.

3. EMBL AUTHORING TOOL

Similar to adding links to web page content, an authoring tool is also needed for manually adding EMBL links to contents that an author wants to signify. It is difficult for a user editing a document to optimally blend a barcode with document content. To facilitate EMBL creation, the EMBL authoring tool should arrange EMBL based on barcode blending coefficient optimization in a neighborhood. The EMBL-content blending result will then be provided to the authoring tool user for original document readability check before finalization. There are three criteria for the authoring tool:

1. The tool should minimize document editor's effort. More specifically, the authoring tool should only request the user to select an EMBL anchor point for each EMBL generation.
2. The tool should minimize the EMBL interference with the original document. More specifically, the tool should minimize the barcode blending coefficient of EMBL so that a document reader can easily 'see through' an EMBL.
3. The tool should minimize the noise of the transmission channel. More specifically, the tool should find the document patch with lowest 'noise' around an anchor point for reliable barcode communication with a low barcode blending coefficient.

3.1 EMBL Optimization

The EMBL optimization starts with an anchor point (m,n) specified by the user editing the document. The anchor point is the point that the editor wants to be associated with the linked media. This can be done with a mouse-click on an editing interface. After the authoring tool gets the anchor point input, it will find the best EMBL parameters for that anchor point.

As we explained in the previous section, the EMBL authoring should minimize the EMBL interference with the original document. Denoting the barcode image with I_B, the original document image with I_C, the barcode blending coefficient with α, and the blended image with I_{BC}, we can describe the barcode blending process with the following equation:

$$I_{BC} = \alpha \cdot I_B + (1-\alpha) \cdot I_C \quad (1)$$

To make the original document I_C more visible to users, the blending coefficient α should be as small as possible. On the hand, α should be large enough for reliable barcode recognition. Denoting the barcode decoding process with D, we can get the EMBL optimal blending coefficient α* with the following equation:

$$\alpha^* = \min_{D(I_{BC}(\alpha))=D(I_B)}(\alpha) \quad (2)$$

If we allow more choices for EMBL optimization, the tool may also change I_C (corresponding to barcode transmission channel selection or location in the document), and I_B (corresponding to barcode transmission signal or barcode type or data). The EMBL placement parameters depend on the type of EMBL we use. If we use the linear barcode shown in Figure 1 (a), the authoring tool only needs five parameters for EMBL placement. These five parameters include 2 parameters for the EMBL placement origin (x,y), one parameter for the EMBL scale λ, and two parameters for the EMBL placement orientation (u,v) on the paper surface. In addition to EMBL pose optimization, EMBL may also be optimized over barcode types (signal transmission bases) based on the original definition of channel capacity. The I_B change depends on the barcode-type set S_B if the system does not have the freedom to change the encoded data. With these EMBL adjusting parameters, equation 2 becomes:

$$\alpha^* = \min_{D(I_{BC}(\alpha, I_C(x,y,u,v,\lambda), I_B(S_B)))=D(I_B(S_B))}(\alpha) \quad (3)$$

To simplify the explanations in this paper, we fixed the barcode base set S_B to code-128, fixed the linear barcode orientation with bars perpendicular to the document regular line direction, and fixed the barcode size. With these parameters fixed, we only need to do the optimization over α, x and y. Through understanding the optimization over α, x and y, it is not difficult for readers to understand optimization over more parameters.

Even though equations 2 and 3 identify optimal parameters, they do not specify an efficient way to estimate these parameters. We designed efficient parameter estimation methods based on the guidance of the Shannon-Hartley theorem. Equation 4 is the mathematical formula for the Shannon-Hartley theorem:

$$C = B \log_2\left(1 + \frac{S}{N}\right), \quad (4)$$

where C is the channel capacity, B is the channel bandwidth, S is the signal magnitude, and N is the noise magnitude. Guided by this theorem, we assume that the system can get a reasonable barcode identification rate if the communication channel's signal-to-noise-ratio (SNR) is over a certain limit.

Blending coefficient optimization at a fixed location according to equation 2 can be performed by adjusting the barcode blending coefficient according to equation 1 and feeding the blended document patch to a barcode reader for verification. The optimization will be time consuming if we change the blending coefficient with small steps during the optimization. To reduce the computational cost, we assume that a larger blending coefficient α (darker barcode blending) will lead to better barcode identification. Since a larger blending coefficient α in equation 1 is equivalent to a higher SNR for barcode transmission, this assumption is consistent with the Shannon-Hartley theorem. It is also consistent with the experiment in [25]. With this assumption, we can perform a binary search of the optimal blending coefficient and stop the optimization when the search interval is smaller than a predefined threshold.

The binary search mechanism only saves computation for the blending coefficient optimization. Optimizing blending coefficient at every possible document patch with pose parameters V (including position and orientation parameters) is still time consuming. Assume the barcode detection time is 15 ms and we need to try 8 intervals during our binary search for good blending-coefficient accuracy. Then the blending coefficient search will cost 120 ms for each location. 120 ms is not a long time for the whole application. However, it is a long time if we want to try this optimization at every possible document patch in a neighborhood. Because we cannot assume 'barcode blending on the right is always better/worse than barcode blending on the left' or 'barcode blending on the top is always better/worse than barcode blending at the bottom', we cannot use binary search to speed up the optimization over EMBL poses. However, this optimization may be simplified with the guidance of Shannon-Hartley theorem and a proper noise model. Since we know the transmission signal S, finding the channel with the largest SNR is equivalent to finding the channel with the lowest noise. Therefore, we can search the document patch (channel) with lowest channel noise (less content interference) around an EMBL signified-location for barcode transmission with minimum magnitude (maximum transparency and least interference to document). Denote N as the noise level of a channel with channel pose parameters (x,y), and (x*,y*) as the optimal pose parameters of an EMBL, the location optimization process may be described with the following equation:

$$(x^*, y^*) = \underset{(x,y)}{\arg\min}\, N(x,y) \cdot \quad (5)$$

This equation converts a document patch optimization problem to a noise estimation problem. With the fast noise estimation approaches described in the following sections, we can use this equation to solve the document patch optimization problem in a fast way.

3.2 Noise Models

Because we know the transmission signal (barcode) in the optimization problem, the optimization over SNR becomes an optimization over noise. Since document content normally harms the barcode recognition after barcode-content blending, document content is considered as noise in our noise estimation process. To perform optimization over noise, we should have proper noise models for noise estimation. Since normal linear barcodes mainly use high frequency spectrum for data encoding, the DC and very low frequency signals have very little impact to barcode transmission. On the other hand, because some barcodes have directions on paper, the SNR for this type of barcodes changes over directions. Therefore, both omni-directional and directional noise models are needed for the optimization. Currently, we focus on three noise models:

1. The first model for noise estimation is the gradients of document content. This model suppresses low-frequency content components that do not have much impact on the barcode transmission. The gradient noise model can be

directional (gradients in a certain direction), or omni-directional (combined gradients in all directions).

2. The second model for noise estimation is based on feature points, such as SIFT feature points or SURF feature points. In theory, these feature points are distinguished gradient points at a certain scale. This model can also be directional or omni-directional depending on whether the system considers the direction of each feature point.
3. The third model for noise estimation is based on edges at a certain scale (i.e. band-pass filtering an image for a certain scale). The edge is the projection of gradients in a certain direction. This model is a directional model. The regular directions or this model are vertical, horizontal, or diagonal.

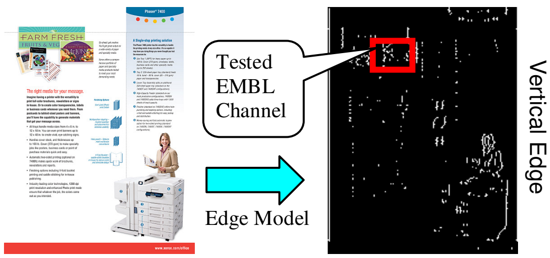


Figure 3. Left: Original image. Right: Vertical edge noise that affect the proper transmission of vertical bars.

The noise estimation in an EMBL channel can be computed by a summation of noise at all pixels inside an EMBL channel. Figure 3 gives an example for estimating noise in a potential EMBL channel. In this Figure, the left image shows the original document, and the right image shows vertical-edge-noise level of every pixel computed from the original image. Assume we want to put an EMBL in a document patch corresponding to the rectangular box in the right image; we can get the noise estimation by adding noise from every pixel in the rectangular box. With this noise estimation, we can compare the noise in one channel with noise in a different channel (moving the rectangular box to a different position.), and choose a channel with lowest noise for the barcode transmission. A low noise channel normally needs lower barcode signal magnitude than a high noise channel to achieve the same barcode recognition rate, and lower barcode signal magnitude means less interference to the original document. From the edge maps, we can clearly figure out that the algorithm will favor blank regions for the barcode overlay. This algorithm preference aligns well with our intuition.

3.3 A Noise Estimation Example and Algorithm Speedup

In this section, we will use an Edge-Detector based noise model as an example to explain our approach for speeding up the noise estimation algorithm. This example assumes that the EMBL uses a barcode with vertical bars. By understanding this approach, it will be straight-forward for readers to modify this approach for other models. The Edge-Detector based noise estimation algorithm can be implemented in the following way:

1. Use ‘Edge Detector’ to construct a vertical edge map at a proper scale depending on the barcode bar-width-parameter.
2. After the anchor point is given, try every bounding box in the search range to find a barcode bounding box around the anchor point that includes minimum edge points.

In this noise estimation algorithm, the edge point summation needs to go through all pixels. Since adding the value of pixels in a small window is much faster than having blending-coefficient optimization at the same pixel location, this algorithm can save us a large amount of time for EMBL location optimization. On the other hand, this bounding box search process will still use a large amount of time for a regular search range (e.g. 200 pixel vertical by 200 pixel horizontal). This problem can be solved by using an integral image map. Figure 4 illustrates the computation of an integral image map. In this Figure, the left image is an edge-point distribution map and the middle image is an edge-point integral map. In the edge-point integral map, there is a randomly selected point P with a value N_P. Corresponding to the point P in the edge-point integral map, there is a point P^* in the edge-point distribution map. Even though P and P^* are in different images, they have the same coordinates in these two images (have the same distances to the image-top-edge and image-left-edge). In the edge-point distribution map, there is also a rectangular box between the top-left corner and P^*. The value N_P of the pixel P in the integral image map is computed by adding all pixel values in the rectangular box of the edge-point distribution map. Besides using edge-point counts to measure noise, the edge-point distribution map may be a gradient-value map or a keypoint distribution map (1 if a pixel is a keypoint and 0 if not.) for other noise models.

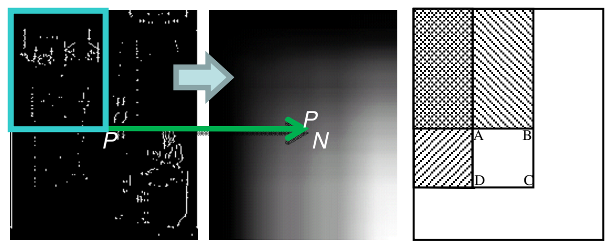


Figure 4. Compute an integral image map. Left: edge point distribution in an image. Middle: N_P equal to the number of edge points in the rectangular box in the left image. The edge point distribution image can also be a keypoint-distribution map, or a gradient-value map. Right: Estimate noise in a rectangular box ABCD.

The right image of the Figure 4 illustrates the noise estimation speedup with an integral image map. In this image, the estimated noise N_{ABCD} in the rectangular box ABCD can be achieved with the following equation:

$$N_{ABCD} = N_C - N_B - N_D + N_A \tag{6}$$

This equation allows us to estimate the noise level in a bounding box (a potential EMBL channel) with three operations. It is much cheaper in computation than going through all pixels in the bounding box. It is also much cheaper than optimizing blending coefficient at a certain location. Moreover, if we know the EMBL orientation and scale (box size), we can pre-compute the noise estimation in each bounding box and save it in an array. By using this array, we can save more real-time computation in the authoring tool.

3.4 EMBL Overlay Algorithm Design

By using the integral image map in the example, we can have a speedup algorithm for the EMBL overlay:

1. Use an ‘Edge Detector’ to construct a vertical edge map at a proper scale depending on the barcode line width parameter.
2. Construct the vertical edge integral map.

3. After the anchor point is given, use the integral map to find a barcode bounding box around the anchor point that has minimum edge points.

4. Adjust barcode blending coefficient and feed the blended document patch to a barcode reader for verification. The optimal blending coefficient can be found with a binary search of the blending coefficient range.

4. EXPERIMENTS AND EVALUATIONS

Because communication via a regular document patch expects lower channel capacity than a dedicated barcode patch, we did not start our testing with an extremely large data encoding set. In our experiment, we use 4 code-128 characters for each system-generated barcode. According to the Code-128 definition, the total number of codes in all these experiments is around 100x100x100x100 = 100 million without considering the start and stop code. We think this is a reasonable scale for initial EMBL applications. In studies conducted by the University of Ohio [24], the real life accuracy of code 128 is 1 error in 2.8 million tests. The accuracy and robustness of the barcode are enabled by the error correction code embedded in the barcode. They are more than enough for our EMBL applications.

One major difference between a traditional barcode and the EMBL is the blending coefficient. This blending coefficient exposes the chance for an intelligent agent (our optimization algorithm) to 'negotiate' with a barcode reader for maximum user benefit and document benefit. Because of this negotiation, we can move an EMBL close to an EMBL signified location and still get all the barcode advantages. Also, we do not need to change the original document layout and allocate extra space for barcode printing. However, all these EMBL benefits come from a hidden assumption that we can gain enough benefits from adjusting the barcode blending coefficient. More specifically, we assume that we can get a small barcode blending coefficient for most EMBL so that the EMBL is nearly transparent and does not have much interference with the original document. If the blending coefficient is close to 1.0 for many EMBLs, the original document patch at the EMBL location is nearly replaced by a barcode and that could affect users' understanding to the original document. Even though we developed an optimization algorithm for EMBL generation, we have not yet proved the correctness of our hidden assumption Therefore, the first experiment we designed for the EMBL is to test the EMBL blending coefficient distribution over all possible communication channels (document patch) in regular documents.

4.1 Barcode Identification Rate Change with Blending Coefficient

For blending coefficient distribution testing, we choose the ICME06 proceeding as our target document set for performance evaluation of the EMBL design and optimization algorithm. The proceeding has 2188 letter-size (8.5"x11") document pages with text, images, and figures. With the large page collection and content variation in the experiment, we believe that our test results will be reliable EMBL-performance evaluations for regular documents.

Because EMBL only uses a small patch in every document page, there are a large number of different EMBL communication channels in each page. Even though it is nearly impossible for us to test all these channels, we can get good estimation of barcode identification percentage change if we randomly sample a large number of channels. Based on this consideration, we randomly generate some patch locations (x,y) in these 2188 pages and get barcode identification percentage for different blending coefficients. To get a comparison baseline, we didn't do EMBL location optimization in the neighborhood of these randomly selected locations. Figure 5 shows the barcode identification percentage change for 1000, 10000, and 20000 channels. From this figure, we can see that the barcode identification percentage trend does not change much for 1000, 10000, or 20000 channels. Therefore, we believe that data from 10000 or more channels is already stable enough to capture the general trend of the identification percentage.

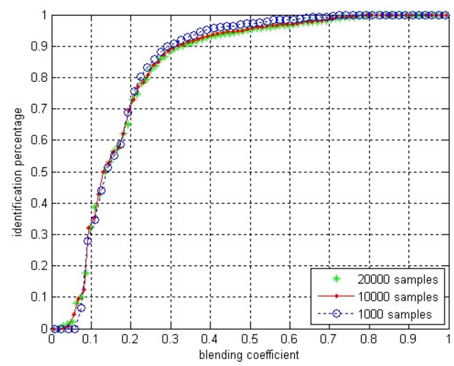


Figure 5. Barcode identification percentage without location optimization for 1000, 10000, and 20000 random patches.

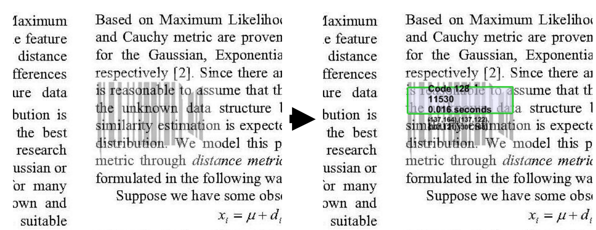


Figure 6. Left: alpha-blend a barcode with document contents using a 0.3008 blending coefficient. Right: verify the blended barcode with a barcode decoder. The decoded information is overlaid on-top of the EMBL base form.

From Figure 5, we can see that more than 85% barcode can be identified when α=0.3. Because we don't have bin-center exactly equal 0.3, Figure 6 shows an example of blending a barcode and contents with a 0.3008 blending coefficient. We randomly picked some text paragraphs and blended barcode with text using 0.3 and 0.5 blending coefficients. Then we asked 5 users to read text contents in the blended documents. We found that no user had difficulties to read the text contents. From this example, we find that most blending coefficients do not interfere much with users' understanding of regular documents. These experimental results strongly support the correctness of our hidden assumption of the barcode blending coefficients. More specifically, these experiments tell us that we can get small blending coefficients for most EMBLs. They also support our idea of optimizing blending coefficients.

By further exploring the identified EMBL patches, we find that the largest blending coefficient in this experiment is 0.9727. Figure 7 shows the blending result with this largest blending coefficient. From this Figure, we can see that the barcode overlay nearly occludes all the content in the randomly selected patch. This is not acceptable for users. To solve this problem, we tried this case with our noise-estimation-based EMBL patch optimization. With our noise-estimation-based EMBL location optimization, the algorithm finds an EMBL mixing location that only requires a 0.1523 barcode blending coefficient. The result of this optimization is shown in Figure 8. From this Figure, we can see that the EMBL is moved to a location with less noise (vertical edges of contents) for a lighter blending.

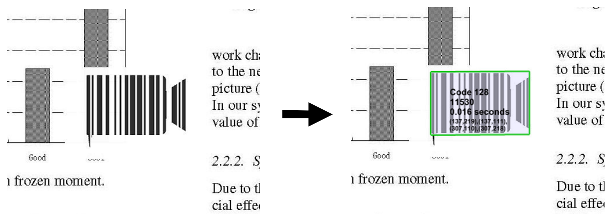


Figure 7. Left: Alpha-blend a barcode with document contents using a 0.9727 blending coefficient. Right: Verify the blended barcode with a barcode decoder. The identification results is shown on top of the EMBL base form.

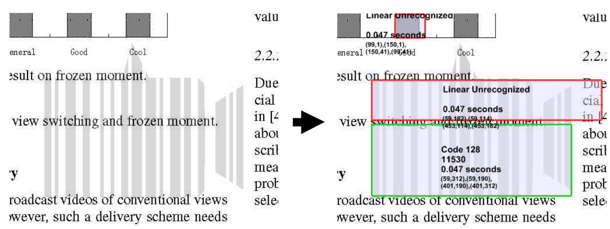


Figure 8. Left: alpha-blend a barcode with document contents using a 0.1523 blending coefficient. Right: Verify the blended barcode with a barcode decoder. The identification result is shown on top of the EMBL base form.

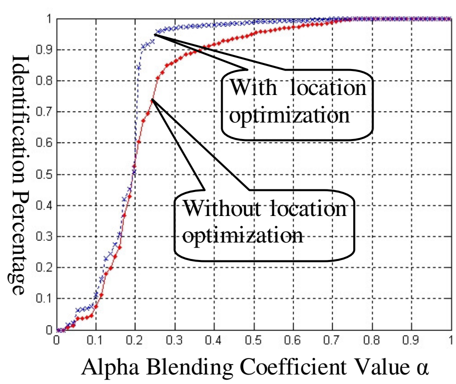
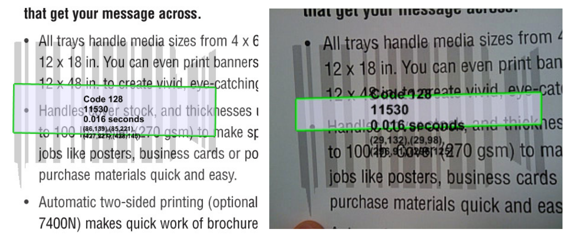


Figure 9. Identification percentage for 16000 randomly selected patch locations. The dashed line shows the identification percentage with location optimization. The solid line shows identification percentage without location optimization.

Beyond this single case, we also tested the statistical improvement generated by our location optimization procedure. Figure 9 shows the identification percentage change with and without the location optimization. Because these curves are measured without imaging noise from different applications, they are very close to blending coefficient lower bounds of different approaches (the location fixed blending approach or the location optimized blending approach). In other words, the curves on the right of these lower bound curves are curves corresponding to different application scenarios. The blending coefficient upper bound corresponds to a scenario with infinite high noise in a transmission channel. Since the upper bound is known by everyone and the lower bound of an approach is unique for that approach, we show the lower bound of each approach in the paper.

From Figure 9, we can see that the identification percentage of the location-optimized-EMBL approach 100% at a much faster speed than the identification percentage of the fixed-location-EMBL. Without location optimization, the barcode identification percentage gets to 85.5% at α=0.3 and 95.83% at α=0.5. With location optimization, the barcode identification percentage gets to 96.7% at α=0.3 and 98.73% at α=0.5. These facts indicate that our location optimization approach can lead to lighter EMBL blending with less interference to human and document. They strongly support our EMBL location optimization idea.

4.2 EMBL Experiment with Real Captures



(a) **(b)**

Figure 10. (a) EMBL identification after the barcode-content mixture. The original file of a mixed patch is sent to a barcode decoder. (b) EMBL identification in a real capture. The identification can be done when the capture is not perfect.

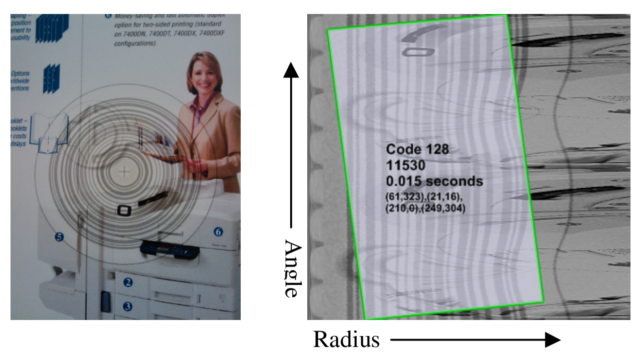


Figure 11. Left: A circular EMBL overlay. Right: Roughly estimate circle center and transform the circular EMBL overlay to the radius-angle domain, then send the radius-angle domain image to a decoder for data decoding. Decoded data is overlaid on-top of the image.

We also tested various EMBLs in different virtual and real environments. The real EMBL printing usually uses a little larger blending coefficient to make an EMBL more robust to capture noise. Currently, we use 1.1α* in real barcode blending for EMBL capture with regular office lighting or daylight. Figure 10 shows a real identification result. Figure 11 shows an experiment with a circular shape EMBL.

5. SUMMARY AND FUTURE WORK

In summary, EMBL is a semi-transparent media-icon-modified barcode overlay on paper document content for linking to associated media. One major difference between the traditional barcode and the EMBL is the optimized blending coefficient. The proposed EMBL construction algorithm can act as an agent to negotiate with a barcode reader on the blending coefficient for maximum user and document benefit. Because of this optimization based negotiation, EMBL minimizes interference with the original document contents and layout. It can also be moved closer to a signified document location so that it is easy for a user to figure out where the media is linked to.

Besides these advantages, EMBL can use the semi-transparent barcode cue to act as a prompt for cell phone capture when a user is interested in the marker signified location. It can leverage the user's barcode capture skill and knowledge of its applications. It exposes the type of linked media data to attract users' interest. It also has the retrieval accuracy, speed, and scalability of the traditional barcode. Additionally, EMBL does not demand too much disk, memory, and computational cost. Because the EMBL has its own encoding pattern, it can also be used to handle most document regions including nearly blank regions. Moreover, this property may also help a machine to distinguish identical regions in different documents.

We are currently working on more thorough tests of different EMBL optimization parameters, such as barcode type (signal base for data transmission) and barcode size. We are also interested in investigating more accurate noise models for the blending coefficient optimization. With a better noise model for location optimization, we will have a better chance to find a smaller blending coefficient for each anchor point. Designing more media type icons that can benefit human, document, and machine at the same time is another important topic of our research.

6. REFERENCES

[1] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L. "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346-359, 2008.

[2] Erol, B., Emilio Antunez, and J.J. Hull. HOTPAPER: multimedia interaction with paper using mobile phones. In Proceedings of ACM Multimedia'08, pp. 399-408.

[3] Pixazza, Now a picture is worth more than a thousand words. http://pixazza.com/.

[4] Hare, J., P. Lewis, L. Gordon, and G. Hart. MapSnapper: Engineering an Efficient Algorithm for Matching Images of Maps from Mobile Phones. Proceedings of Multimedia Content Access: Algorithms and Systems II, 2008.

[5] Liu, Q., Liao, C., Wilcox, L., Dunnigan, A., and Liew, B. 2010. Embedded media markers: marks on paper that signify associated media. In Proceeding of the 14th international Conference on intelligent User interfaces (Hong Kong, China, February 07 - 10, 2010). IUI '10. ACM, New York, NY, 149-158.

[6] Henze, N. and S. Boll. Snap and share your photobooks. In Proceedings of ACM Multimedia'08, pp. 409-418.

[7] Hull, J.J., B. Erol, J. Graham, Q. Ke, H. Kishi, J. Moraleda, and D.G.V. Olst. Paper-based Augmented Reality. Proceedings of IEEE ICAT 2007, pp. 205-209.

[8] Liu, Q., H. Yano, D. Kimber, C. Liao, and L. Wilcox. High Accuracy and Language Independent Document Retrieval with a Fast Invariant Transform. In Proceedings of IEEE ICME'09, pp. 386-389.

[9] Liu, X. and D. Doermann, Mobile Retriever: access to digital documents from their physical source. Int. J. Doc. Anal. Recognit., 2008. 11(1): pp. 19-27.

[10] Lowe, D.G., Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision, 2004. 60(2): pp. 91-110.

[11] Nakia, T., K. Kise, and M. Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. LNCS, Vol. 3872, pp. 541-552.

[12] Parikh, T.S., P. Javid, S. K., K. Ghosh, and K. Toyama. Mobile phones and paper documents: evaluating a new approach for capturing microfinance data in rural India. Proceedings of ACM CHI'06, pp. 551-560.

[13] Rekimoto, J. and Ayatsuka, Y. 2000. CyberCode: designing augmented reality environments with visual tags. In Proceedings of ACM DARE 2000, pp. 1-10.

[14] Rohs, M. Real-world interaction with camera-phones. LNCS, Vol. 3598, pp. 74-89.

[15] Ke, Y. and Sukthankar, R., PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. Proceedings of IEEE CVPR 2004.

[16] Hecht D. L., Embedded Data Glyph Technology for Hardcopy Digital Documents. SPIE -Color Hard Copy and Graphics Arts III, Vol. 2171, pp. 341-352.

[17] Wikipedia, Digital paper. http://en.wikipedia.org/wiki/Digital_paper

[18] Reilly, D., M. Rodgers, R. Argue, et al., Marked-up maps: combining paper maps and electronic information resources. Personal and Ubiquitous Computing, 2006. 10(4): pp. 215-226.

[19] Costanza, E., J. Huang., Designable Visual Markers, Proceedings of ACM CHI'09, pp. 1879-1888.

[20] NODA TSUGIO, MOROO JUN, CHIBA HIROTAKA, Print-type Steganography Technology, Fujitsu 2006, Vol. 57. No. 3, pp. 320-324.

[21] Morrison, A., Oulasvirta, A., Peltonen, P., Lemmela, S., Jacucci, G., Reitmayr, G., Näsänen, J., and Juustila, A., Like bees around the hive: a comparative study of a mobile augmented reality map. In Proceedings of ACM CHI '09, pp. 1889-1898.

[22] Microsoft Tag. http://www.microsoft.com/tag/

[23] William, B., A. Judith D. Bar code scanner for reading a lower layer luminescent invisible ink that is printed below a upper layer luminescent invisible ink. United States Patent 5525798.

[24] Jorge Salcedo, DESIGN AND PILOT TEST OF A BAR CODE SYSTEM FOR INVENTORY CONTROL, Ohio University MS Thesis, 1990.

[25] Joseph, E. and Pavlidis, T. 1994. Bar Code Waveform Recognition Using Peak Locations. IEEE Trans. Pattern Anal. Mach. Intell. 16, 6 (Jun. 1994), 630-640.