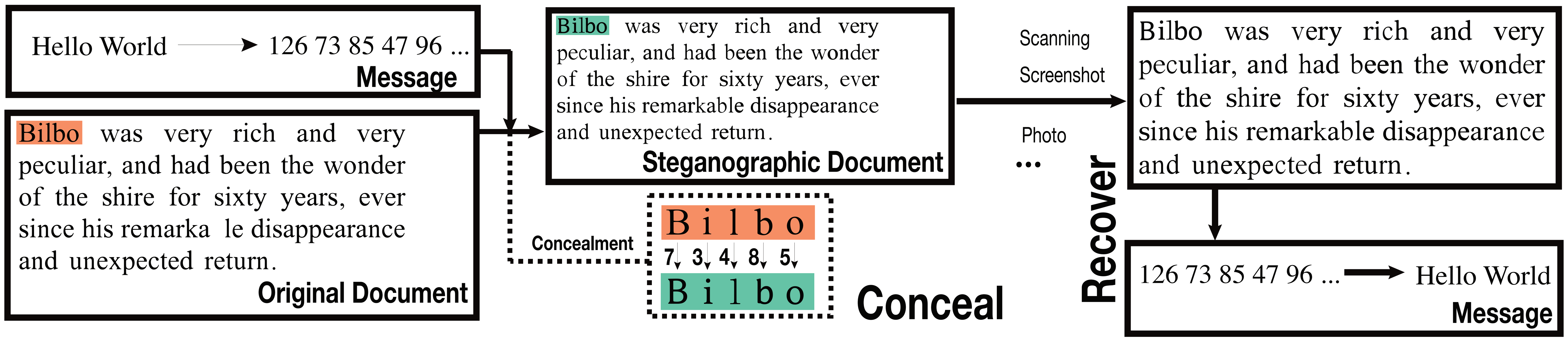
FontCode:
Embedding Information in Text Documents using Glyph Perturbation

ACM Transaction On Graphics

Chang Xiao, Cheng Zhang, Changxi Zheng

Columbia University

Method Overview

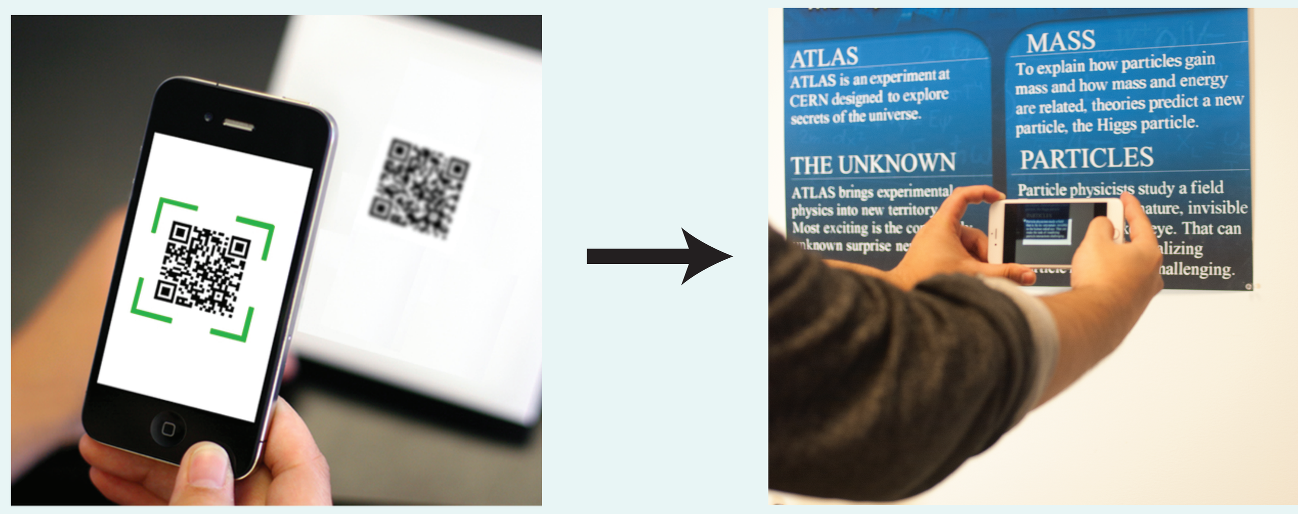


We propose FontCode, a new information embedding technique for text documents. Instead of changing text letters into different ones, we alter the glyphs (i.e., the particular shape designs) of their fonts to encode information, leveraging the recently developed concept of font manifold in computer graphics.

We demonstrate that our technique enables a wide array of applications, using it as a text document metadata holder, an unobtrusive optical barcode, a cryptographic message embedding scheme, and a text document signature.

Motivations

Traditional QR code FontCode



Barcodes have numerous applications in advertising, sales, inventory tracking, robotics, augmented reality, and so forth. However, traditional barcodes such as QR code contains no information in its original appearance.

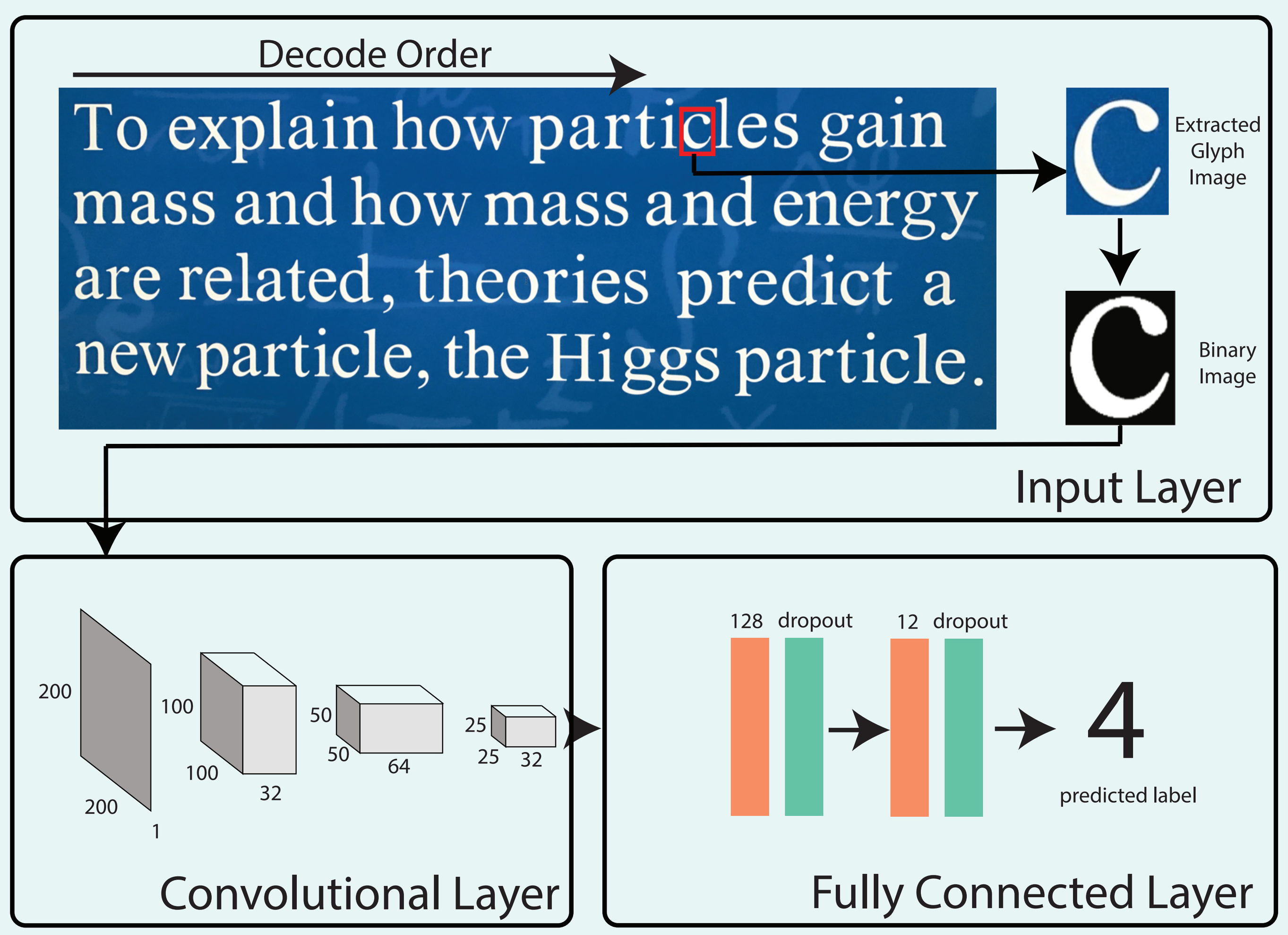
We introduce FontCode, an information embedding technique for text documents. Provided a text document with specific fonts, our method embeds user-specified information in the text by perturbing the glyphs of text characters while preserving the text content. It can be particularly suitable for use as a replacement of QR codes in an artistic work such as a poster or flyer design, where visual distraction needs to be minimized.

Font Manifold

We alter the glyphs (i.e., the particular shape designs) of their fonts to encode information, leveraging the recently developed concept of font manifold.

A 2-dimensional font manifold was created for every character, such that every location on this manifold generates a particular glyph of that character. Then, it allows us to alter the glyph of each text letter in a subtle yet systematic way, and thereby embed messages.

Glyph Recognition



To retrieve information from a coded text document, the first step is to recover an integer from each letter.

We treat glyph recognition as an image classification problem: provided an image region of a letter which has a list of perturbed glyphs {u0,u1,...} in the codebook, our goal is to classify the input glyph of that letter as one from the list. Therefore, we train a Convolutional Neural Network (CNN) for each letter in a particular font.

To recognize the glyph perturbation of each letter using CNNs, we first preprocess the image, in order to overcome the gap between real image and training data, also decrease dimension of training data. We crop the region of each letter using its bounding box detected by the OCR. We then binarize the image region using the classic algorithm by Otsu. This step helps to eliminate the influence caused by the variations of lighting conditions and background colors. Lastly, we resize the image region to have 200x200 pixels. This 200x200, black-and-white image for each letter is the input to our CNN.

Our CNNs will be used for recognizing text document images that are either directly synthesized or captured by digital cameras. Correspondingly, the training data of the CNNs consist of synthetic images and real photos. Those synthetic data was created by a photo realistic renderer with different exposure setting.

Coding Scheme

Bit stream	010111	10111010	1000	1001001
Integer sequence	23	186	8	73
Letter seq.	Bilbo	was ve	ry ric	h and very...

Our embedding method takes an input message and a text document. It encodes the message into a series of integers and divides the letters into blocks. The integers are assigned to each block and embedded in individual letters. To recover the message, we extract integers by recognizing the glyphs of individual letters. Then, the integers are decoded into the original plain message.

We use an error-correction maximum likelihood coding scheme based on Chinese Remainder Theorem. For the example above, each block is consist of 5 letters, which are used to encode an ascii number from 0 to 255. Since different character has different number of usable glyphs, we use Chinese Remainder Theorem to choose which glyph to use, which guarantee to recover the original message under a certain amout of error rate.

Chinese Remainder Theorem

Let $p_1, p_2, \dots p_k$ denote positive integers which are mutually prime and $M = \Pi_{i=1}^{k} p_i$. Then, there exists a injective function:

$$\phi : [0, M) \rightarrow [0, p_1) \times [0, p_2) \dots \times [0, p_k)$$

defined as $\phi(m) = (r_1, r_2, \dots, r_n), \forall m \in [0, M), r_i = m \mod p_i$.

This theorem indicates that given k pairs of integers (r_i, p_i) with all pi being mutually prime, there exists a unique non-negative integer m < M satisfying ri = m mod pi for all i = 1...k.
Indeed, m can be computed using the formula:

$$m = CRT(\mathbf{r}, \mathbf{p}) = r_1 b_1 \frac{P}{p_1} + r_2 b_2 \frac{P}{p_2} + \dots + r_n b_n \frac{P}{p_n}$$

where $P = \Pi_{i=1}^{n} p_i$ and b is computed by solving a system of modular equation using the classic Euclidean algorith that is:

$$b_i \frac{P}{p_i} \equiv 1 \pmod{p_i}, i = 1 \dots k$$

Examples

Here we demonstrate how fontcode works. In this paragraph, we encoded two different hyperlinks. Our iphone app can decode those two messages and redirect to two different web pages.

Here we demonstrate how fontcode works. In this paragraph, we encoded two different hyperlinks. Our iphone app can decode those two messages and redirect to two different web pages.

Acknowledgements

We are grateful to Julie Dorsey for her suggestions of paper revision, Klint Qinami and Michael Falkenstein for video narration. This work was supported in part by the National Science Foundation (CAREER-1453101 and 1717178) and generous donations from SoftBank and Adobe. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authorsand do not necessarily reflect the views of the funding agencies or others.

Contact

email:
chang@cs.columbia.edu
web:
http://www.cs.columbia.edu/cg/fontcode/