An Iterative Template Matching Algorithm Using the Chirp-Z Transform for Digital Image Watermarking

Shelby Pereira and Thierry Pun

1. INTRODUCTION

The popularity of the World Wide Web has clearly demonstrated the commercial potential of the digital multimedia market. Unfortunately however, digital networks and multimedia also afford virtually unprecedented opportunities to pirate copyrighted material. As a result, digital image watermarking has become a an active area of research. Techniques for hiding watermarks have grown steadily more sophisticated and increasingly robust to standard image processing techniques. Current techniques embed watermarks by means of adding a spread spectrum (SS) signal in the spatial domain or a transform domain such as the DCT, DFT, or Wavelet transform. In order to render the algorithm resistant to cropping, scaling and rotation, the idea of using an invisible template embedded in the DFT domain has also been adopted.[1] The template contains no information but is merely a tool used to recover possible transformations in the image. Ultimately, the recovery of the watermark is a two stage process. First we attempt to determine the transformation (if any) undergone by the image, then we invert the transformation and decode the spread spectrum sequence. Consequently the DFT is appropriate for embedding a template since linear transformations in the spatial domain imply linear transformations in the DFT domain.[2]

Recent algorithms have used log-polar maps followed by cross-correlation between a known reference template and the recovered image to recover changes in scale and rotation.[1,3] Recently, however, Petitcolas[4] presented a benchmark against which watermarking algorithms can be objectively evaluated. Results indicate that current algorithms perform poorly against combinations of small rotations, scale changes and cropping. In order to improve the performance of the template matching, we propose a new algorithm which is more accurate and faster than current techniques.

2. TEMPLATE MATCHING

Once the watermark has been embedded into the image, a template consisting of approximately 25 points uniformly distributed in the DFT is added to the image.[1] The magnitude is chosen so that the point is locally a peak and the phase is set to 0. The problems during detection are to determine if the image has undergone a transformation and then to invert the transformation before decoding the watermark. Here we limit ourselves to rotations, scale changes and cropping. We first consider the properties of log-polar maps which will be used in the template matching algorithm. We then present a new algorithm for detecting the template in spite of the sampling problems associated with the logarithmic scale.

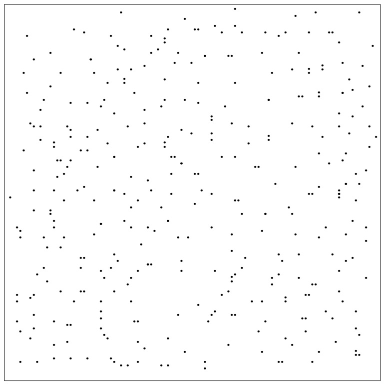
Consider a point $(x, y) \in \Re^2$ and define: $x = e^{\mu}\cos\theta; y = e^{\mu}\sin\theta$ where $\mu \in \Re$ and $0 \leq \theta < 2\pi$. One can readily see that for every point (x, y) there is a point (μ, θ) that uniquely corresponds to it. The new coordinate system has the properties that rotations and scales are converted to translations as follows: $(\rho x, \rho y) \leftrightarrow (\mu + \log\rho, \theta)$ and $(x\cos(\delta) - y\sin(\delta), x\sin(\delta) + y\cos(\delta)) \leftrightarrow (\mu, \theta + \delta)$.

The template matching process is divided into two parts. In the first part, we extract the rotation and scale changes and then invert them. In the second part we recover the translation offset required to synchronize the decoding of the watermark with the encoding. This second step is necessary in situations where the image has been cropped since we must account for the fact that points in the image are missing. Consequently, correctly locating the starting point is essential to the recovery of the watermark.

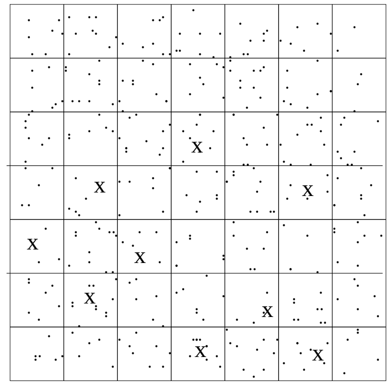
In the first part of the template matching process, the rotation and scale compensation is performed as follows: (1) if the image is rectangular, extract the largest available square from the image; (2) compute the magnitude of the DFT of the windowed image; (3) calculate the positions of the local peaks in the filtered DFT using a small window (10 to 14 works well) and store them in a sparse matrix; (4) compute the corresponding points in log polar space; (5) compute the positions of the points in log polar space of the known template; (6) exhaustively search the space of possible matches between the known template and the extracted peaks and thereby deduce the optimal offset; (7) use the offset to compute the scale and rotation change (8) invert the rotation and scale; (9) perform the Chirp-Z transform[5] of size 50×50 on the image with the frequencies centered at a known template point; (10) detect the location of the peak ; (11) if the peak is not in the center of the template, we must further compensate the x and y scales of the image. If necessary iterate steps 9-11 at other peak locations and average the result. In practice averaging over 5 peaks yields sufficient precision.

The main problem with steps 1-8 lies in the fact that the resolution of the DFT is limited by the number of samples in the image. Steps 9-11 therefore represent the key point of the algorithm. This iteration increases the accuracy by an important factor. The Chirp-Z transform allows us to tightly sample the frequency domain centered at a point of interest with a resolution independent of the number of samples in the image. Simply replacing the DFT in step 2 with a Chirp-Z transform is not feasible since we must then tightly sample the whole frequency domain which leads to a problem which is intractable. By performing the iteration in steps (9)-(11), we can limit ourselves to points of interest which are the known template points. Another key point is that this iteration allows us to recover small changes in aspect ratio (and not solely in uniform scaling), since when we perform the Chirp-Z transform, the x and y offsets provide the scale changes along each axis. Such small changes in aspect ratio are often incurred as a result of rounding errors, so these changes must be detected and compensated for.

We also note that step 6 can be rendered more efficient by sorting the points in the search space and dividing this space into a grid. Then, only points lying in the square of the template points need to be considered during the matching process. This is illustrated in figures (a) and (b) which contain the search space of recovered peaks which is divided into a grid. The template points are represented by "X" and only the peaks in the associated square need to be considered during the point matching. In the second part of the template matching process, we recover the translation offset required to synchronize the decoding of the watermark with the encoding. This is done by performing a fast correlation between the template and the image which has already been compensated for rotation and scale changes. This second step is necessary in situations where the image has also been cropped since we must account for the fact that points in the image are missing, and consequently determine the starting point of the encoding process in order to correctly synchronize the decoding of the watermark. For example, if the watermark



(a) Peaks



(b) Grid

was embedding by using blocks (LOT for example[2]) then it is essential to determine an appropriate point to start transforming the image. In particular, we require that exactly the same block locations be used during encoding and decoding.

3. RESULTS

Our results indicate that the proposed approach accurately recovers rotations to within 0.2 degrees and the scale of an image with sub-pixel accuracy. In particular, when we invert the scale and rotations incurred by the image, we recover the original image size. Furthermore the proposed approach performs well in conjunction with the LOT watermarking scheme[2] as evaluated by the watermarking benchmark tests.

REFERENCES

1. Alexander Herrigel, Joe Oruanaidh, H. Peterson, Pereira S., and Pun T., "Secure copyright protection techniques for digital images," in *2nd International Information Hiding Workshop*, Portland, Oregon, April 1998.
2. Shelby Pereira, Joe Oruanaidh, and Thierry Pun, "Secure robust digital watermarking using the lapped orthogonal transform," in *SPIE 11th Annual Symposium on Electronic Imaging: Science and Technology EI'99*, San Jose, California, January 1999.
3. Geoffrey B. Rhoads, "Steganography systems," in *International Patent WO 96/36163 PCT/US96/06618*, November 1996.
4. Fabien A.P. Petitcolas, Ross Anderson, and Markus G. Kuhn, "Attacks on copyright marking systems," in *2nd International Information Hiding Workshop*, Portland, Oregon, April 1998.
5. Giorgio Bonmassar and Eric Schwartz, "Space-variant fourier analysis: The exponential chirp transform," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, October 1997.