Attacking Visible Watermarking Schemes

Chun-Hsiang Huang, Ja-Ling Wu, *Senior Member, IEEE*

ABSTRACT

Visible watermarking schemes are important IPR protection mechanisms for digital images and videos that have to be released for certain purposes but illegal reproductions of them are prohibited. Visible watermarking techniques protect digital contents in a more active manner, which is quite different from the invisible watermarking techniques. Digital data embedded with visible watermarks will contain recognizable but unobtrusive copyright patterns, and the details of the host data should still exist. The embedded pattern of a useful visible watermarking scheme should be difficult or even impossible to be removed unless intensive and expensive human labors are involved. In this paper, we propose an attacking scheme against current visible image watermarking techniques. After manually selecting the watermarked areas, only few human interventions are required. For watermarks purely composed of thin patterns, basic image recovery techniques can completely remove the embedded patterns. For more general watermarks consisting of thick patterns, not only information in surrounding unmarked areas but also information within watermarked areas will be utilized to correctly recover the host image. Although the proposed scheme does not guarantee that the recovered images will be exactly identical to the unmarked originals, the structure of the embedded pattern will be seriously destroyed and a perceptually satisfying recovered image can be obtained. In other words, a general attacking scheme based on the contradictive requirements of current visible watermarking techniques is worked out. Thus, the robustness of current visible watermarking schemes for digital images is doubtful and needs to be improved.

Keywords: Visible Watermarking, Attacks, Image Inpainting, Image Recovery

1. INTRODUCTION

In recent years, digital watermarking techniques have been extensively exploited and regarded as a potentially effective solution against illegal reproduction or theft of multimedia contents. Consequently, many watermarking schemes have been proposed, and several classifying methods have also been established according to the characteristics of watermarking schemes. One important classification is to divide watermarking techniques into visible and invisible ones according to the visibility of watermark data in embedded contents. Generally, invisible watermarking [1-3] is suitable to be treated as an IPR mechanism for most forms of digital contents. Users cannot perceptually recognize the difference between invisibly watermarked contents and original ones, unless explicit watermark extraction procedures are used. The copyrights of content providers or authors are protected in a passive way by checking if watermarks can be found in questionable contents. On the other hand, visible watermarking schemes [4-10] are used to protect digital images or videos that have to be released for certain purposes, such as contents used in distant learning web sites or digital library, while illegal copying or reproduction is prohibited. Visible watermarking schemes protect IPR in a more active way: visibly watermarked content often contains recognizable but unobtrusive copyright patterns indicating the identity of IPR owners. Unless watermark patterns can be completely removed without destroying visual quality of contents they protect, no one could use visibly watermarked data directly. For the time being, papers discussing the security of invisible watermarks are in abundance, but thorough discussions about the robustness of visible watermarking techniques are relatively rare. In this paper, we will focus on possible attacks against visible image watermarking techniques, and work out an effective attacking method based on the contradictive requirements that must be satisfied in current visible watermarking schemes. In section 2, we will introduce state-of-the-arts of visible watermarking techniques, list the basic requirements of visible watermarking schemes, and derive a general model for visible watermarking. In section 3, based on some important observations, an effective attacking scheme is derived, and the experimental results of attacks against current visible watermarking schemes are presented to justify our arguments. Section 4 gives some discussions and concludes this paper.

2. SOME BASICS ABOUT VISIBLE WATEDRMARKING

2.1 State-of-the-Arts

Visible watermarking is a common IPR protection mechanism for digital images and videos that have to be released for certain purposes while the IPR of content owners should be protected at the same time. A well-known visible watermarking algorithm is proposed by Braudaway et al. in [4]. In this algorithm, watermark images have the same dimensions as those of host images. That is, a one-to-one correspondence between each pixel in the watermark image and that of the host image has been established. Pixels in the watermark image can be divided into transparent and non-transparent ones. Pixel values in the host image

corresponding to the transparent regions of the watermark will remain the same after watermark embedding while pixels corresponding to the non-transparent regions of the watermark will be altered according to the corresponding watermark values and the adopted embedding models.

During watermark embedding procedures, amounts of brightness increasing and decreasing should be perceptually equal for a fixed change occurred everywhere in the color space. It implies that an approximately uniform color space should be used, such as the CIE 1976 (L*u*v*) space and the CIE 1976(L*a*b*)-space [11]. The pixel brightness alternation process can be described as:

$$\tilde{Y}_{n,m} = Y_{n,m} + \frac{(\mu_{n,m} - \mu_\tau)}{|\mu_A - \mu_\tau|} \frac{Y_w}{38.667} \left(\frac{Y_{n,m}}{Y_w}\right)^{2/3} \Delta L^* \qquad (1)$$

where $Y_{n,m}$ and $\tilde{Y}_{n,m}$ are the brightness values of each pixel in the unmarked original and the watermarked image, respectively. Y_w is the brightness of the "scene white", which approximates the viewing illuminant reflected from a perfect diffuse reflector. $\mu_{n,m}$ is the brightness of the corresponding pixels in the watermark image. μ_τ is the transparent brightness, while μ_A is the maximum amplification or maximum absorption according to the inequality between $\mu_{n,m}$ and μ_τ. ΔL^* is the degree of increment chosen manually to adjust the perceptibility of watermark patterns and image details.

Figure 1 shows the original and watermarked lena images, in which the watermark was embedded based on the aforecited visible watermarking algorithm.

Meng and Chang proposed a visible video watermarking scheme in [5]. The same model of pixel alternation used in [4] is adopted, but the model has now been extended to the DCT domain by simple statistic model approximation for the convenience of processing directly in the MPEG-compressed domain. In [6, 7], local features related to the degree of distortion tolerances, such as edge locations, texture distributions and luminance sensitivity, are taken into consideration so that more unobtrusive watermarked images can be generated. Simple statistics of block-DCT coefficients are calculated and analyzed to decide the watermark embedding energy of each block. Edge integrity will be preserved, in these approaches, since the edge information is essential to maintain the image quality. And the energy of the embedded watermark is larger in highly textured areas than in smooth ones due to different noise sensitivity. In additions, the watermark energy of mid-gray regions is also smaller than other areas since the noises are more visible against a mid-gray background. In [8], the model of watermark energy setting is the same as that of [6,7], but in addition to the visibly embedded watermark, a fragile invisible watermark is also adopted to check if the

visible watermark is altered or not. In [9], the perceptibility of the watermark patterns is adjusted according to the standard deviation of pixels in each equal-divided blocks of the original image. The larger the contrast of a block is, the larger the amount of alternation for the gray-level values of pixels in that block will be. In [10], a wavelet-domain visible watermarking method is proposed. The scaling factors of the alternated pixels are determined by the effect of luminance and spatial characteristics.

2.2 A General Model for Visible Watermarking

Although different models and embedding domains have been chosen by aforecited watermarking schemes, their goals are the same. A useful visible watermarking technique should meet the following requirements:

- Perceptibility of host image details: All the details in the original images should remain visible after watermark embedding. In other words, contents should not be rendered useless after being visibly watermarked. For example, assume that some details in a image showing a recently discovered ancient vase in a historical research website is rendered unrecognizable by an embedded institute symbol, archaeologists visiting this web site may miss important information that these details should disclose, and the original intent to release information via Internet will now be in vain.
- Visibility of watermark patterns in embedded contents: Visible watermarks often consist of meaningful patterns, symbols, trademarks, or texts to identify content providers or owners. These copyright data should be able to be easily recognized from the original contents by naked eyes. No explicit watermark extraction techniques are required.
- Robustness: Embedded visible watermark patterns should be difficult or impossible to remove unless exhaustive and costly user interventions are adopted. Furthermore, it is even more desirable that professional skills in fine arts are required to transform a watermarked content back to its unmarked state.

Without loss of generality, a visible watermark embedding procedure can be represented by the following equations:

$$I' = K_1 * I + K_2 * W , \tag{2}$$

$$D(E_I(I'), E_I(I)) < Threshold_I , \tag{3}$$

$$D(E_W(I'), E_W(W)) < Threshold_W \tag{4}$$

Eqn. (2) models the process of visible watermarks embedding; Eqns. (3) and (4) represent the first two requirements mentioned above. Definitions of the consisting symbols of Eqn. (2) to (4) are given as follows: I' is the watermarked image, and I is the unmarked original image. W is the watermark image containing copyright patterns. K_1 and K_2 are the weighting factors reflecting the characteristics of nearby

areas within the original image, the watermark image, or both. These weighting factors are often decided block-by-block, and respectively multiplied to corresponding pixels of the original image and the watermark image to produce unobtrusive embedded results. For the scheme proposed in [4], $K_1 = 1$, and K_2 is decided by the last term on the right-hand side of eqn. (1). For other previously mentioned schemes, the values of K_i may be proportional or inversely proportional to the contrast value, amount of textures, or luminance sensitivity within nearby areas of each pixel. It is worthy to note, for the sake of security, the exact model of embedding-factor decision should not be available. More specifically, K_i and the exact intensity of W should be difficult to model or guess by collecting watermarked contents. Otherwise an inverse operation will be easily inferred from reverse-engineering processes, and close approximations of I will then be restored. D is a distance function measuring the perceptual difference of its two entries, and for most of the time, it is not strictly defined since whether a visible watermarking scheme works or not is often decided and judged by content users subjectively. E_I and E_W are image feature extraction operators for images being protected and watermark patterns, respectively. We will discuss their definitions in practical visible watermarking schemes later. $Threshold_I$ is the largest allowable distortion of image details that observers can tolerate and, at the same time, the signature of I' can be maintained. $Threshold_W$ is the largest allowable distortion of the embedded watermark pattern that the copyright information can be clearly recognized.

3. ATTACKS AGAINST VISIBLE WATERMARKING SCHEMES

3.1 Important Observations

Before designing an attacking mechanism for current visible watermarking techniques, we made some observations on the characteristics of visible watermarks. Here are several important conclusions we obtained:

- Since embedded visible watermark should be unobtrusive and original-detail-preserving, watermark patterns should not contain complex textures or shape structures. In fact, most copyright patterns adopted in the literature are simple logos or trademarks [4-10] composed of few colors and flat areas. So, it is reasonable to assume that feasible watermark patterns are of simple shapes and few constant colors, and the border of watermarked areas could be easily selected by common users.
- The perceptibility of image details in the embedded content relies on the preservation of edge information contained within the watermarked area. Thus, the aforecited image feature extraction operator E_I, given in eqn. (3), should involve edge operations so as to extract and preserve edge information for further usage.
- The reason why users can identify copyright patterns contained in watermarked contents mainly relies on the fact that the shapes (contours) of embedded patterns are preserved and could be differentiated

from host contents. If the contour of an embedded visible watermark is completely removed or greatly distorted without introducing serious visual quality degradation, the content owner will not be able to claim his copyrights against illegal users any more. Thus, the feature extractor E_W, defined in eqn. (4), should be capable of extracting self-identifying shape structures of watermark patterns in the input image.

- The robustness of visible watermarking schemes mainly lies on the inevitability of applying exhaustive and costly labors, if its removal is a must. Thus, an effective attacking scheme should involve as few user interventions as possible. But during attacking, a certain kind of user intervention – manually selecting the watermarked areas - is still required. It is because that no automatic recognition mechanisms can be applied to correctly find out the watermark embedded areas without having specific domain knowledge about the watermark patterns and the embedding parameters, nowadays.
- To devise a general attacking mechanism, only information in the watermarked contents can be of use. That is, in a marked image, only pixels in the unmarked area and the remaining information within the watermarked areas can be utilized during watermark removal. Other information, such as the embedding parameters or the actual intensity of the watermark image, is assumed to be unknown because only embedded contents are available to the attackers.

After grasping these observations, we can now proceed to design our attacking schemes.

3.2 Attacking Simple Watermarks with Image Recovery Techniques

Watermarks adopted in visible watermarking schemes are often composed of texts, trademarks, or symbols identifying the copyrights of the protected digital contents. Patterns of these watermarks are quite simple, and the widths (i.e. the degree of thickness) of these patterns are often limited.

After attackers selecting the areas supposed to be occupied by embedded watermark patterns, the watermark-attacking problem is quite similar to an image recovery problem [12,13]. The selected areas can be regarded as the areas to be recovered and their corresponding details are assumed to be unknown. Obviously there are high correlations between the selected areas and the surrounding unaltered areas; therefore, we can refill these watermarked areas according to the intensity information contained in the surrounding unaltered areas. The most straightforward attacking method is to reconstruct a watermarked pixel value with the average intensity values of all nearby unmarked pixels within a window of certain size. The watermarked areas can be shrunk one layer at a time, and the recovered pixel values can be used to recover the next inner pixels. Finally, an approximated version of the watermarked area will be obtained. This averaging technique can obtain good perceptual quality when the recovered areas are contained within flat areas of the original images, but obvious blurring artifacts will occur when watermarks are embedded in positions across object boundaries or containing complicated textures.

Figure 2 shows the pepper picture embedded with a simple watermark pattern by the prescribed visible watermarking algorithm and the corresponding recover-by-averaging result of it. A zoom-in version of one watermarked area refilled by averaging techniques is also depicted. Blurring artifacts across object boundaries can be easily observed. Thus, a better approximation model to recover watermarked areas should be employed to successfully attack a visible watermarking scheme.

Image inpainting [13] is an image recovery technique that modifies an image into an undetectable form so that damaged areas can be recovered or undesired objects can be removed. Information from the surrounding area is propagated into the selected area. The reason why inpainting-technique-based attack works better than the averaging attack lies on the fact that the inpainting technique prolongs the approaching edges arriving at the border of the areas to be inpainted. The whole inpainting process can naturally be implemented as an iterative process and described by the following equations:

$$I_{n+1} = I^n + \Delta t I_t^n \tag{5}$$

$$I_t^n(i, j) = \left(\delta L^n(i, j) \cdot \frac{N(i, j, n)}{\left| N(i, j, n) \right|} \right) \left| \nabla I^n(i, j) \right| \tag{6}$$

where δL stands for the measure of smoothness difference in the neighborhood of the pixel $I(i, j)$, N stands for the direction orthogonal to the gradient, i.e. it represents the isophote direction. Only known pixels are used during iterative processes. Details of the equations can be found in [13]. To employ the inpainting techniques in watermark attacking problems, the watermarked areas selected by attackers are now regarded as the areas to be inpainted.

Pictures in Figs. 3 and 4 show the procedures and results of the inpainting attacks. The pixel values within the watermarked areas are assumed to be unknown, and all the refilled information is derived from the surrounding unaltered neighboring pixels. According to the experimental results, the inpainting attack is effective to remove simple watermarking patterns composed of thin lines or symbols. In other words, recovering areas embedded with watermark patterns composed of thin lines only is in essence the same as solving an image recovery problem, and image inpainting is a good choice to deal with this problem.

3.3 Attacking General Watermarks with the Proposed Algorithms

Image recovery techniques can successfully remove visible watermarking patterns consisting of thin lines. However, a general watermarking scheme may adopt watermark patterns composed of thick lines. For

example, a company may use bold-faced or emphasized texts in its trademarks or logos to catch consumers' eyes. If that kind of patterns is adopted directly in visible watermarking schemes, the watermarked areas will occupy a significant portion of the host images. Therefore, simple image recovery techniques considering only unmarked information will not work well. Fig. 5 shows the embedded lena image (previously shown in Fig. 1(a)) with its watermarked areas recovered by the inpainting attack introduced in the previous section. Obviously, the refilled areas corresponding to the inner watermark texts are incorrect. The reason is straightforward: since the under-attacking lines are thick now, pixels far from the borders of the watermarked areas have lower correlations with known pixels, and cannot be correctly predicted by using unmarked information only. Information from surrounding regions of different intensities will interfere with each other when recovering pixels in the center of watermarked areas.

3.3.1 Refilling flat areas divided by the borders of watermark patterns

Since the surrounding information is not sufficient to recover watermark areas composed of thick patterns, based on the observations listed in section 3.1, we try to explore the remaining information within the watermarked area in order to devise a successful attacking scheme for watermarking patterns composed of thick lines or symbols. As mentioned earlier, the edge information of the original image should be preserved after embedding. Otherwise, the requirement of image-detail perceptibility will not be satisfied. In general, the watermarked areas as well as the unmarked areas can be easily classified into edge regions and flat regions. Furthermore, since the shapes of the watermark patterns can be easily obtained in the beginning of attacking by user intervention, flat regions composed of both marked and unmarked areas (i.e., flat regions divided by the watermark boundaries) could be readily identified and extracted. Since pixels within a flat region would possess similar intensity values, we can easily refill the marked area within a flat region based on unmarked information within the same flat region**.** Figure 6 illustrates the concept mentioned above.

To prove the effectiveness of our attacking method, the embedded lena image, as shown in Figure 1(a), will be recovered by using the proposed algorithm. The surrounding watermark patterns composed of thinner texts can be easily identified and separated from the bold "NTU" patterns in the center of the lena image using morphological operators [14] and independently recovered by applying the aforecited inpainting attacks. And then, after applying a Sobel edge operator and a thresholding operation, the watermarked areas can be classified into edge areas and flat areas. There is a trade-off in deciding the threshold values classifying edge areas and flat areas. If the threshold value is high, most flat areas will be correctly recognized, but areas contain gradient colors will be incorrectly regarded as flat areas and altered with colors of surrounding flat regions. On the other hand, if a low threshold is adopted, the gradient regions will be correctly preserved, but some flat areas may be neglected, and no enough information can be used for correction. Since least user intervention is preferred to visible watermark removal, predefined threshold values will be more desirable. In our experiments, after applying the Sobel edge operator, a default threshold

value of 20 is used. And if the result is not perceptually satisfying, another threshold may be adopted to attack again until satisfying results are obtained with this setting. Of course, an exploration of the characteristics of embedded images will be helpful in choosing more appropriate threshold values.

Figure 7 shows the embedded lena image in which all surrounding thin watermark patterns have been removed by inpainting techniques and all possible flat regions within areas corresponding to the bold “NTU” word have also been updated by the flat-region recovery scheme we just mentioned. The shapes and structures of the original watermark patterns have been seriously distorted, and most of the refilled regions show perceptually correct colors. Although some watermarked areas remain unchanged, the embedded watermark patterns are no longer recognizable. That is, the embedded copyright patterns are rendered useless now.

3.3.2 Recovering remaining areas

Although all flat regions can be identified and classified by exploring the remaining edge information, only regions containing unmarked pixels can be refilled correctly. Flat regions that are completely contained within watermarked areas cannot gather any useful information and therefore are unaltered. Remaining watermark areas are now composed of fully contained flat areas and the edge regions within the areas corresponding to thick watermark patterns.

Originally, we try to regard the remaining edge regions as thin watermark patterns mentioned in the previous section, and by using similar inpainting techniques, an approximation of the edge regions can be obtained – but this approximation is not as good as the one predicted in the previous section for two reasons: First, now the fully-contained flat regions also contribute to the prediction results. Second, the inpainting algorithm has not yet been specially designed as an edge recovery mechanism.

Since all available information contained in watermarked areas as well as in unmarked areas have been used, the remaining areas can only be recovered by approximated prediction. We try to recover the edge areas automatically by preserving the differences of intensity between edges and their surrounding flat areas, which may originally be unmarked or recovered in the previous step. The intensity of an edge is adjusted according to the average amount of intensity increases or decreases in surrounding recovered flat areas. Although the actual color alternation model used by the watermarking algorithm may be quite complex and is unavailable during attacking, the recovered edge areas are perceptually unobtrusive. The result is reasonable because the contrasts between edge areas and surrounding flat areas are still preserved. Although the actual intensity of edge areas cannot be exactly recovered, the perceptual semantics is well preserved. In fact, the semantics always exists even after visible watermarks are embedded due to the requirement that the details of original image should be preserved.

Finally, in order to recover the fully contained flat areas, the same algorithms applied to watermarked edge areas are adopted again. The only difference is that now we adjust the pixel intensity according to the amounts of pixel intensity changes occurring in the surrounding recovered edges. Since the difference between the watermarked edges and the fully contained flat areas within the watermark are preserved, the recovered images are perceptually unobtrusive as we expected.

Instead of using prediction, we can also adjust the intensity values of the fully contained flat areas uniformly and manually because the remaining areas are quite small and sparsely distributed. Although user interventions are inevitable to decide the appropriate amount of uniform intensity alternation, the attackers' job is simple since the mask of remaining areas has been automatically produced in previous stages of the proposed attack. The attacker can easily decide the appropriate amount of intensity alternation in an iterative manner.

Figure 8 shows the finally recovered lena image of the watermarked image depicted in Figure 1. It is obvious that not only the copyright patterns have been nearly removed, but the attacked image quality has also been greatly improved, both objectively and subjectively.

Figures 9 to 12 show the results after applying our attacking procedures to several test images of different characteristics (Pictures showing our experimental results are also available at http://www.cmlab.csie.ntu.edu.tw/~bh/research.htm.) For each test image, the default edge/flat area classifying threshold value is set to 20, and each RGB color component is processed independently. For both Airplane and Lena images, perceptually satisfying results are obtained, as shown in figure 9 and figure 10, respectively. The "NTU" word as our watermark pattern has been successfully removed, and no visible artifacts remain in final results. While attacking the embedded Fruit image, we found that most of the watermarked patterns can be successfully removed, but the gradient surface of the pear is not correctly recovered. As explained above, the phenomenon means that the classifying threshold is too high for the embedded image. Thus, a lower threshold value of 10 is used to preserve the gradient characteristic of the fruit surface, and satisfying quality images are obtained (see figure 11). As for the Baboon image, the rough contours of the watermark patterns within the highly textured regions on the baboon face remain visible (as shown in figure 12) because no correct information in surrounding flat areas can be utilized. Adopting a higher threshold value may be of limited help, and even worse some originally correctly recovered areas may be incorrect now because more incorrect information is now utilized. No better results can be automatically produced. The only solution is to employ more user interventions on adjusting intensity values of watermarked areas in different highly textured parts. Although the proposed attacking schemes did not

remove the entire embedded patterns, but the contours of embedded patterns are destroyed seriously, and the claim of copyright assertion is much more weak now.

3.4 Attacking Other Enhanced Watermarking Schemes

We have successfully attacked the most basic watermarking scheme described in section 3. What will happen if we apply the attacking algorithm to against some more complex watermarking schemes? As we have introduced, in most recently published watermarking schemes, the degree of pixel alternation is decided according to the local characteristics of original image blocks. For example, highly textured areas can tolerate higher distortions than their flat counterparts, thus the degree of pixel alternation could be larger in highly texture areas. However, this kind of enhancement could not introduce serious artifacts or additional details within watermarked areas, otherwise the original image details would be interfered. Under these constraints, the classifying results of the watermarked areas into edges and flat areas shall be the same. Therefore, similar recovered images will be produced if the proposed attacking scheme is applied.

To prove our argument, we choose to implement the visible watermarking technique introduced in [6], and attack the embedded images so that the effectiveness of the proposed attacking algorithms can be evaluated. It is worthy to note that the implemented scheme is representative since visible watermarking techniques introduced in [6-8] adhere to the same watermark-energy adjusting principles. Only the statistics used in calculating the distortion tolerance and embedding parameters are slightly different. In [6], the original image is divided into blocks in the first step. Important local features such as edge distributions, amounts of textures, and the luminance sensitivity are taken into consideration while deciding the embedding weighting factors, i.e. K_1 and K_2 in eqn. (2), of each block. Areas with larger distortion tolerances will be altered more than those with smaller distortion tolerances. An obvious effect of this enhancement is that colors of areas that must be preserved (such as edges) within watermarked areas will be more similar to their original states.

Figs. 13 to 15 show the watermark images, the watermarked pictures, and the corresponding recovered ones. Similarly, visibly satisfactory results are obtained. There is one thing worthy of mentioning: according to the embedding mechanisms introduced in [6], all pixels in watermark images will contribute to the whole embedded images. In other words, even pixels in the original image that corresponds to backgrounds areas (areas other than watermark patterns in the watermark image) will now be altered during embedding. Thus, for truly reflecting the objectivity, while calculating the objective image quality index (such as PSNR), it is the image embedded with a “null watermark” (the watermark consisting of only background pixels) that shall be adopted as the comparing basis instead of the original one. Figure 16 shows the adopted watermark image and the null watermark image. According to the experimental results, not only the embedded watermarks are unobtrusively removed, the corresponding objective measures also show great

improvements in image quality.

3.5 The Proposed Visible Watermarking Algorithm

The proposed attacking algorithm against visible watermarking schemes can be represented by the following steps. Only the watermarked host image I' is assumed to be available in the beginning of attacking.

(i) Obtain the mask M representing the watermarked areas by user intervention.

(ii) Apply mathematical morphology operators to separate M into M_{thin} and M_{thick}, representing areas corresponding to thin patterns and thick patterns, respectively.

(iii) Recover areas belonging to M_{thin} by inpainting attacks.

(iv) Apply edge detectors to regions belonging to the whole image, and classify M_{thick} into edges (indicated by M^{E}_{thick}) and flat areas (indicated by M^{F}_{thick}) using threshold value T_{edge}.

(v) Based on M and M^{F}_{thick}, recover flat watermarked areas having unaltered flat neighbors (denoted by $M^{F\text{-}D}_{thick}$).

(vi) Attack remaining edged watermarked areas (M^{E}_{thick}) by approximated prediction basing on intensity adaptation information of nearby pixels in $M^{F\text{-}D}_{thick}$, obtained in step (v).

(vii) Attack fully contained flat watermarked areas (indicated by $M^{F\text{-}C}_{thick}$) by approximated prediction basing on intensity adjusting information of nearby pixels in M^{E}_{thick}, recovered in step (vi).

(viii) If the recovered image quality is not satisfying, set a new threshold value T_{edge}, return to step (iv), and iterate again.

Figure 17 shows the flow chart of the proposed general visible-watermark attacking scheme.

4. DISCUSSIONS AND CONCLUSIONS

4.1 The Effect of Watermarked Areas Selection

Users should be able to select the watermarked areas easily because an unrecognizable watermark cannot protect copyrights of content providers. And as mentioned in section 2, clearly identifying the shape of watermarked areas is one of the basic requirements for visible watermarks. However, the user selection process is not guaranteed to be perfectly accurate. Serious inaccuracy in selecting watermarked areas may result in poor recovery results.

4.2 Can Visible Watermarking Schemes Provide Secure Copyright Protection?

From the derivation of attacking schemes introduced in section 3, it follows that the perceptibility requirements of current visible watermarking schemes form security "holes" so that attackers can successfully remove the embedded copyright patterns without involving time-consuming computations or

expensive human labors. The requirement that the watermark patterns should be clearly recognizable ensures that attackers could select the contour of the watermark easily. Furthermore, the requirement that the image details must be recognizable provides the edge information needed to appropriately propagate surrounding unmarked information. In other words, attackers can easily select watermarked areas and recover the marked regions automatically.

To improve the robustness of visible watermarking schemes, here are our suggestions:

(i) During watermark embedding, edge information of the host images should be regarded as a parameter of watermark embedding because the shapes of the remaining watermark patterns after the flat-region recovery step are greatly affected by the edge structures of the host images. For example, as shown by our experiments, embedding visible watermarks into areas containing highly textured patterns or of gradient color distributions will undoubtedly increase the difficulty of watermark removal.

(ii) Adequately increasing the complications of shapes or textures of embedding patterns will increase the difficult to correctly select the watermarked area. For example, watermark patterns with anti-alias borders have blurred areas outside its exact contours. The intensity values of these blurred areas could be only slightly different from those of surrounding transparent pixels, and these areas may be easily neglected when proceeding watermarks selection. As a result, these areas may contain recognizable fragments of embedding patterns rendered in a color between the watermarked and unmarked intensity values after recovering by watermark attacking techniques mentioned above, and further user intervention will be inevitable.

(iii) As mentioned in [8], in order to detect if an image was once embedded with visible watermark patterns, invisible fragile watermarking techniques should be adopted. In other words, we can use fragile watermarks to guard the integrity of visibly embedded images. Thus, the security of visible watermarking techniques can be enhanced with the help of double insurance provided by the added invisible fragile watermarks. Of course, in this case, the problem of, “if a visible watermarking scheme is secure?” is now transferred to the security problem of fragile watermarks.

4.3 Conclusions and Future Works

In this paper, we gave an introduction to some basic requirements of current visible watermarking schemes. Possible security problems and attacking schemes are also included. Visible watermarking schemes can be very useful in distributing images and videos with copyrights over Internet. A more robust visible watermarking scheme based on the conclusions of this paper will be the goal of our future work. For the time being, guarding visible watermarked images with fragile watermarking techniques may be a feasible solution.

Acknowledgement

The authors benefit greatly from the comments on earlier versions of this paper. They gratefully acknowledge the help of the Associate Editor, Dr. Mark Liao, and the anonymous reviewers.

Reference

[1] I. J. Cox, J, Kilian, F. T. Leighton, and T. Shamoon, “Secure spread spectrum watermarking for multimedia, ” IEEE Trans. on Image Processing, vol6. Dec. 1997

[2] C. Podilchuk and W. Zeng, “Image adaptive watermarking using visual models,” IEEE J. Select. Areas Commun., Special Issue on Copyright and Privacy Protection, vol. 16, pp. 525-539, May 1998

[3] C.T. Hsu and J. L. Wu, “Hidden digital watermarks in images,” IEEE Trans. On Image Processing, vol 8., No. 1, January 1999

[4] G. Braudaway, K.A. Magerlein, and F. Mintzer, "Protecting Publicly Available Images with a Visible Image Watermark," Proceedings of the SPIE International Conference on Electronic Imaging, San Jose, CA, February 1-2, 1996, Vol. 2659, pp. 126-133.

[5] J. Meng and S. F. Chang, “Embedding visible watermarks in the compressed domain,” Proc. of ICIP 98.

[6] M.S. Kankanhalli, Rajmohan and J. R. Ramakrishnan, “Adaptive Visible Watermarking of Images,” IEEE International Conference on Multimedia Computing and Systems, 1999.

[7] S. P. Mohanty, J. R. Ramakrishnan, and M. S. Kankanhalli, “A DCT domain visible watermarking technique for images,” Proc. of ICME 2000.

[8] S. P. Mohanty, J. R. Ramakrishnan, and M. S. Kankanhalli, “A Dual Watermarking Technique for Images, “ Proc. ACM, pp. 49-51, 1999

[9] P. M. Chen, “A visible watermarking mechanism using a statistic approach,” Proc. of WCCC-ICSP 2000

[10] Y. Hu and S. Kwong, “Wavelet domain adaptive visible watermarking,” Electronic Letters, vol. 37, Sep. 2001

[11] D. B. Judd and G. Wyszecki, Colors in Business, Science, and Industry, John Wiley & Sons, Inc., New York, 1975

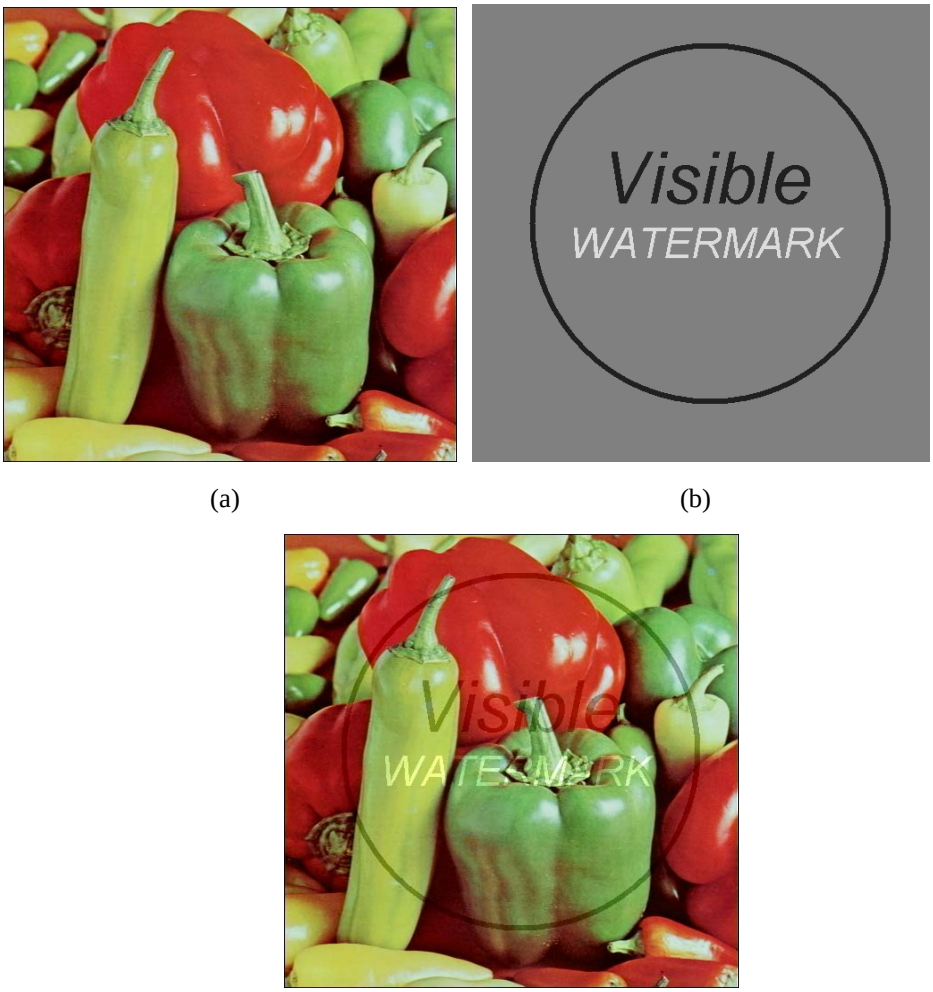
[12] A. C. Kokaram, R.D. Morris, W. J. Fitzgerald, P.J.W. Rayner, “Interpolation of missing data in image sequence,” IEEE Trans. on Image Processing, vol 11, 1995

[13] M. Bertalmio, V. Caselles, and C. Ballester, “Image inpainting,” SIGGRAPH 2000, Aug. 2000

[14] R. M. Haralick and L. G. Shapiro, Computer and Robot Vision, Addison-Wesley, 1992.

(a) (b)

Figure 1. (a) The visibly watermarked lena image produced using the algorithm proposed by Braudaway et al. (with a PSNR of 24.87dB), and (b) the corresponding watermark pattern. Alhough the embedded patterns are recognizable, important detail information in the lena image is preserved.



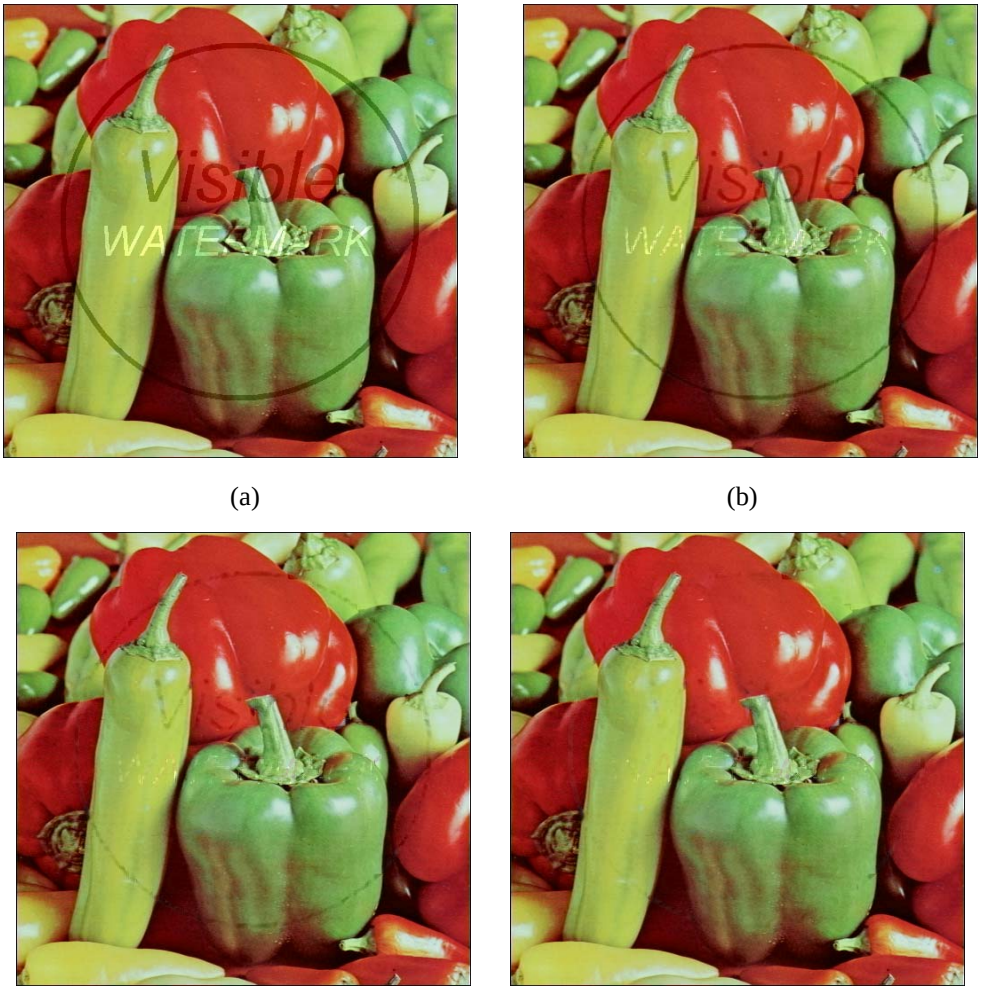
(a) (b)

(c)

(d) (e)

Figure 2. (a) The pepper host image, (b) the corresponding simple watermark pattern, (c) the visibly watermarked image, and (d) the recovered image obtained by using the simple averaging technique. Notice that watermarked areas

within flat regions of the original image are refilled correctly, but obvious artifacts can be found in the areas cross object boundaries as notified by the white grids. (e) A zoom-in version of one grid area in (d), in which obvious artifacts can be observed.



(a) (b)

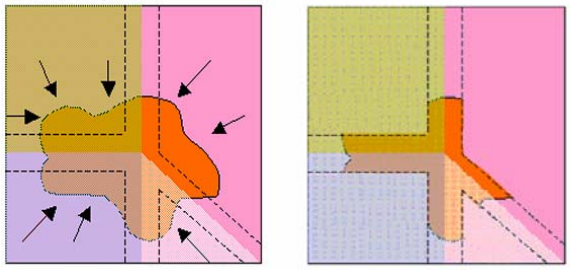
(c) (d)

Figure 3: The snapshots of recovered pepper images during inpainting iterations are listed. (a) The pepper image embedded with the watermark shown in Figure 2(b) before inpainting iterations. The pepper images after (b) 200, (c) 1000, and (d) 10000 inpainting iterations are shown, respectively. It is clear that the boundaries of the host images can be gradually recovered. The recovered image with nearly no visible artifacts can be obtained.

(a) (b)

Figure 4. (a) The embedded lena image and (b) the inpainted lena image.

Figure 5. The watermarked lena image, shown in Fig.1, is recovered by applying inpainting techniques. The watermarked areas occupied by the bold characters “NTU” cannot be successfully recovered because the widths of these areas are significantly large, and therefore, marked pixels far from borders of the watermark patterns cannot gather enough information from surrounding unaltered pixels.



(a) (b)

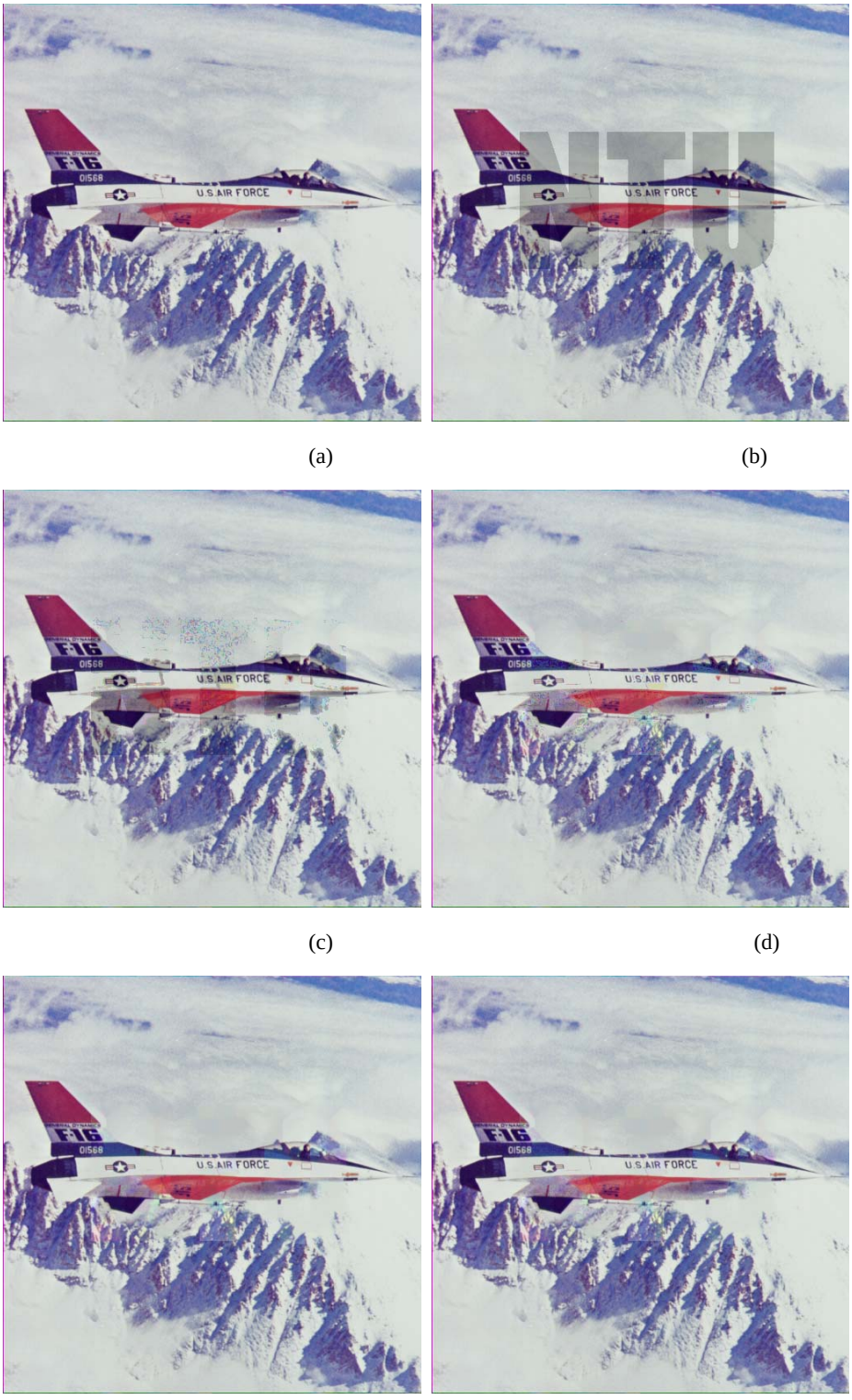
Figure 6. (a) The cloud area in the center represents the selected watermarked pattern. The image is composed of four

different flat regions. The crossroad regions formed by dotted lines are edge areas detected by exploring edge information in both watermarked and unmarked areas. (b) After identifying edge areas, flat areas occupied by watermarks in each of the four regions can be correctly recovered based on surrounding unmarked information within that region.

(a) (b)

Figure 7. (a) This mask image shows the remaining watermarked areas after recovering watermarked flat areas divided by watermark borders. Note that the correlation between the outlines of remaining areas and the edges of the embedded lena image can be clearly identified. (b) The lena image after refilling watermarked flat areas divided by watermark borders (PSNR=28.67). Although there are still some marked areas unaltered, the embedded pattern has been largely destroyed, and the resultant image is perceptually more similar to the unmarked original image.

Figure 8. The lena image after automatic guessing the edge information and manually adjusting the intensity of remaining flat areas (PSNR=30.95).



(a) (b)

(c) (d)

(e) (f)

Figure 9. (a)The original airplane image, (b) the embedded airplane image (PSNR=25.03), (c) the embedded airplane image in which flat watermarked areas divided by watermark borders have been recovered (PSNR=28.24), (d) the embedded airplane image after recovering watermarked edge areas (PSNR=33.43), (e) the embedded airplane image after recovering fully watermarked flat areas (PSNR=34.16), and (f) the airplane image in which fully contained

watermarked flat areas are recovered by manual intensity adjusting (PSNR=34.50).

(a) (b)

(c) (d)

(e) (f)

Figure 10. (a)The original lena image, (b) the embedded lena image (PSNR=27.32), (c) the embedded lena image in which flat watermarked areas divided by watermark borders have been recovered (PSNR=28.91), (d) the embedded lena image after recovering

watermarked edge areas (PSNR=32.43), (e) the embedded lena image after recovering fully watermarked flat areas (PSNR=32.87), and (f) the lena image in which fully contained watermarked flat areas are recovered by manual intensity adjusting (PSNR=33.33).

(a)

(b)

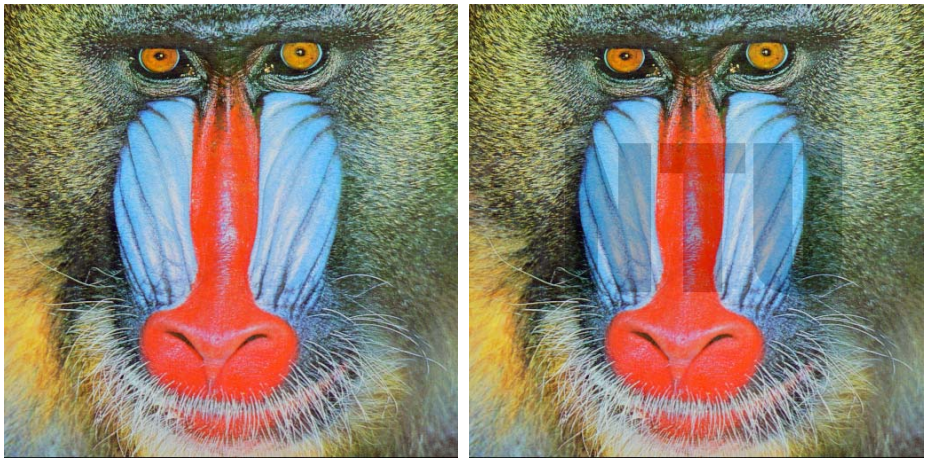
(c)

(d)

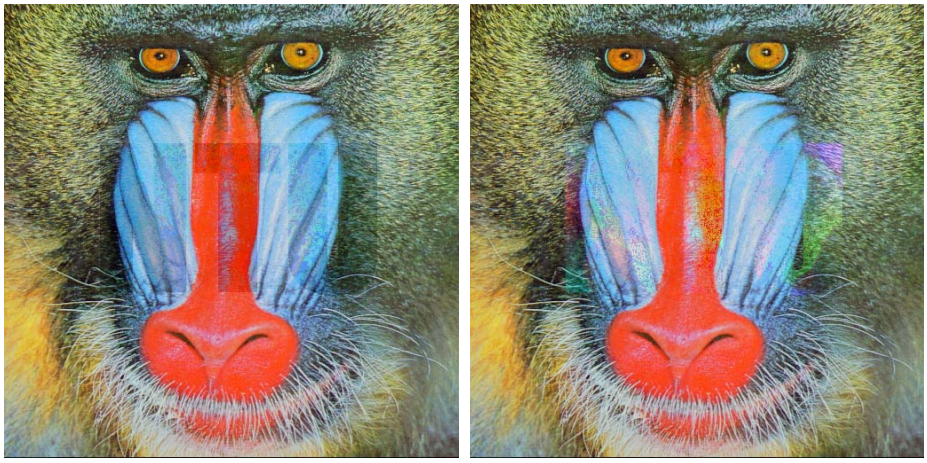
(e)

(f)

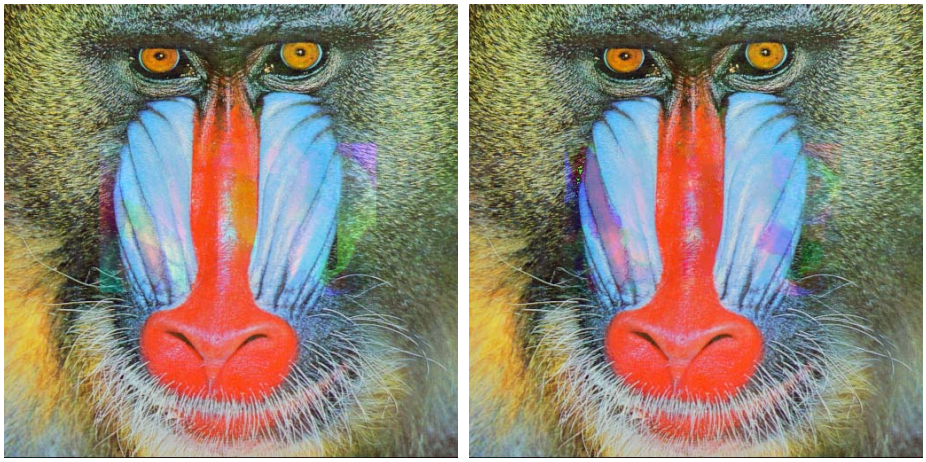
Figure 11. (a)The original fruits image, (b) the embedded fruits image (PSNR=24.53), (c) the embedded fruits image in which flat watermarked areas divided by watermark borders have been recovered (PSNR=26.71), (d) the embedded fruits image after recovering watermarked edge areas (PSNR=32.45), (e) the embedded fruits image after recovering fully watermarked flat areas (PSNR=35.80), and (f) the fruits image in which fully contained watermarked flat areas are recovered by manual intensity adjusting (PSNR=36.28).



(a) (b)

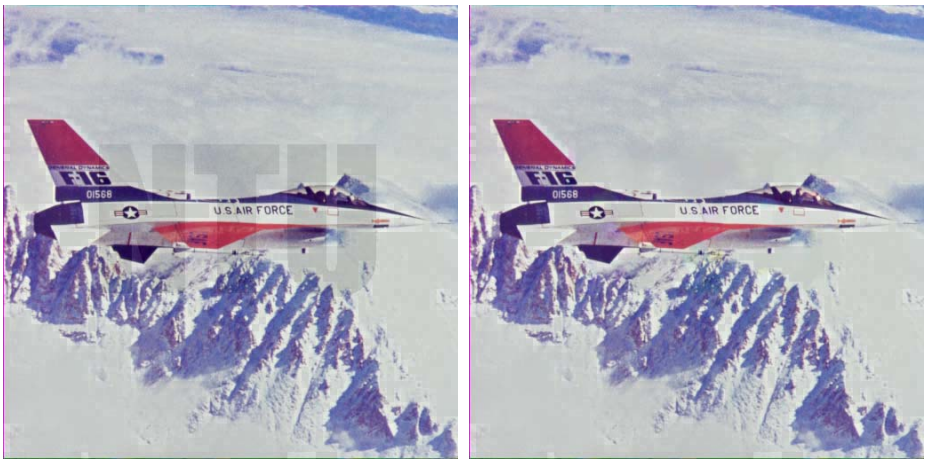


(c) (d)



(e)

Figure 12. (a)The original baboon image, (b) the embedded baboon image (PSNR=26.17), (c) the embedded baboon image in which flat watermarked areas divided by watermark borders have been recovered (PSNR=26.55), (d) the embedded baboon image after recovering watermaked edge areas (PSNR=29.20), (e) the embedded baboon image after recovering fully watermarked flat areas (PSNR=29.64), and (f) the baboon image recovering with threshold value 40 (PSNR=30.28).



(a) (b)

Figure 13. (a) The airplane image (PSNR=30.86) embedded with the enhanced visible watermarking technique and (b) the recovered image (PSNR=39.05).

Figure 14. (a) The lena image (PSNR=31.37) embedded with the enhanced visible watermarking technique and (b) the recovered image (PSNR=35.80).

Figure 15. (a) The fruit image (PSNR=30.96) embedded with the enhanced visible watermarking technique and (b) the recovered image (PSNR=40.89).

Figure 16. (a) The watermark image adopted in experiments showing the effectiveness of the proposed attacking schemes against enhanced visible watermarking schemes and (b) the null watermark used to create the image adopted as a new comparison basis in calculating objective image quality index values.

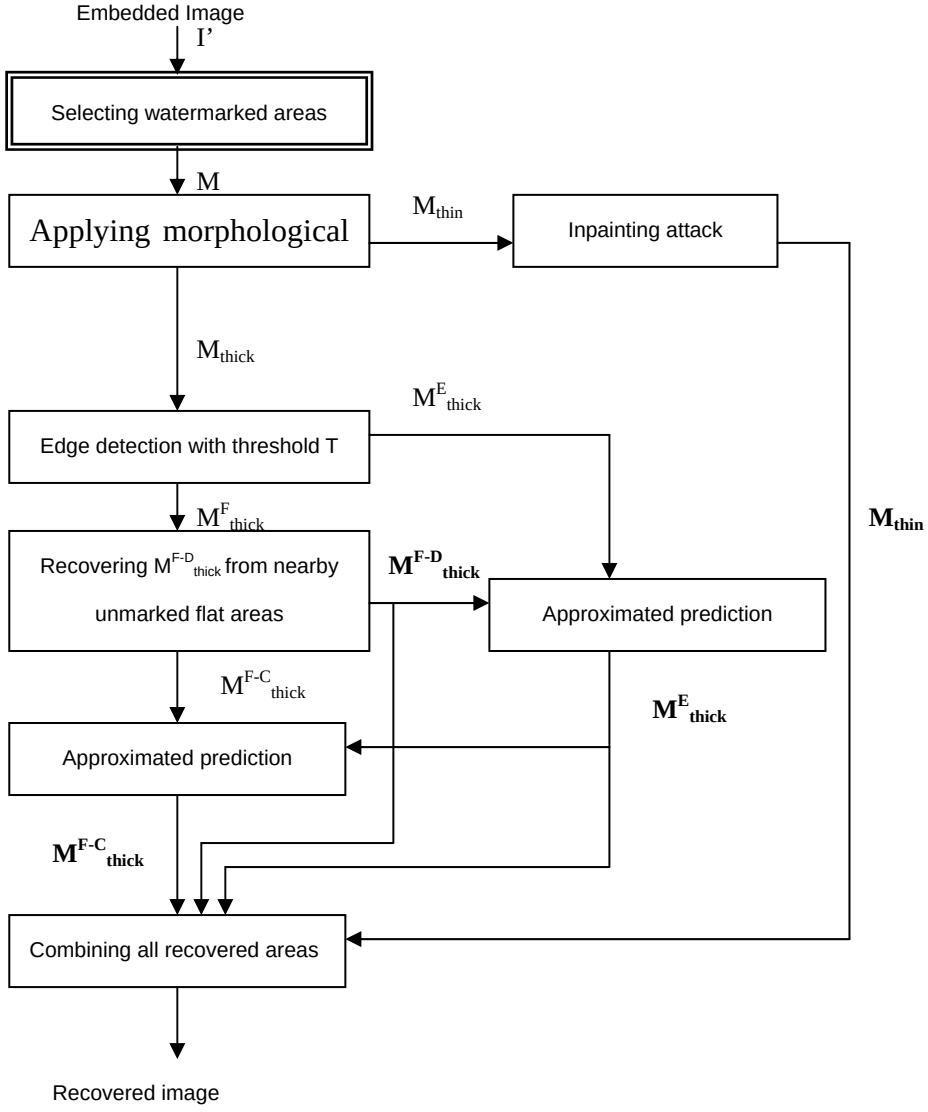


Figure 17. The flowchart of the proposed visible watermark attacking scheme. The function block with double outlines means user intervention is required. The notations represent different parts of the embedded image. Notations using bold fonts means being recovered.