FRAME-PATCH MATCHING BASED ROBUST VIDEO WATERMARKING USING KAZE FEATURE

Ta Minh Thanh[1], *Pham Thanh Hiep*[2], *Ta Minh Tam*[3], *Kohno Ryuji*[2]

[1]Dept. of Network Security, Le Quy Don Technical University, 100 Hoang Quoc Viet,
Cau Giay, Ha Noi, Viet Nam. Email:taminhjp@gmail.com
[2]School of Engineering, Yokohama National University, Yokohama, Japan.
Email:hiep@kohnolab.dnj.ynu.ac.jp
[3]VIETNAM Satellite digital television Co. Ltd.. Email: tam.ta@vstv.vn

ABSTRACT

A frame-patch matching based robust video watermarking using KAZE feature is proposed in this paper. We employed the KAZE feature for matching the feature points of frame-patch with those of all frames in video. In our method, the watermark information is embedded in Discrete Cosine Transform (DCT) domain of randomly generated blocks in the matched region. In the extraction process, we synchronized the embedded region from the distorted video by using KAZE feature matching. RST (rotation, scaling, translation) parameters are estimated and the watermark information can be extracted. Experimental results showed that our proposed method is very robust against geometrical attacks, video processing attacks, temporal attacks, and so on.

***Index Terms*—** KAZE feature, Video watermarking, Geometrically invariant, Feature matching

1. INTRODUCTION

The rapid growth of network and PC technique have facilitated the wide use of digital multimedia content. It makes numerous advantages over the analog multimedia content, like easy copy, economic transmission via network or by physical media (CD, DVD, *etc.*). However, these advantages have outstretched the security concerns of digital multimedia content such as copyright protection, owner's right problem, legal user verification and so on.

In order to protect and preserve the copyright of digital content, many copyright protection techniques have been proposed. Digital watermarking is a promising technique used for copy control and media identification and tracing. In digital watermarking, a watermark information is embedded into an image or video without affecting the quality but that can be detected using dedicated extraction software.

However, various attacks are effective against watermarking methods and watermark information can be destroyed under those attacks [1]. Geometrical distortion is known as one of the most difficult attacks to resist because it resynchronizes the embedded location. Therefore, the watermark synchronization process is required to restore the embedded location.

A number of research related to watermark synchronization has been conducted. Bas et al. [2] proposed a content-based synchronization method, in which they use a set of disjoint triangles through Delaunay tessellation based on the salient feature points extracted from distorted image. In this method, a mis-detection of feature points may deteriorate the disjoint triangles, then the sets of triangles generated during watermark insertion and detection are different. With another view point, Nikolaidis and Pitas [3] proposed a watermark synchronization method based on image-segmentation using an adaptive k-mean clustering technique. The drawback of this method is that the image segmentation in k-mean clustering depends on the image-content, so that image distortions severely affect the segmentation results. Dajun et al. [4], Ho et al. [5], and Lee et al. [6] proposed the watermark synchronization methods using object-based watermarking, where the object location is assumed to be known to make the extraction easy. But such assumption may not hold in real situations. With the similar motivation, Viet and Aizawa [7] used SIFT (Scale-Invariant Feature Transform) features to detect and match the object region. However, in their method, the object region is needed to select manually in advance. These conventional methods can be used only in "proof of ownership" applications and are not suitable for video broadcasting.

For robust watermarking of watermark synchronization, the selection of features (global feature or local feature) is very important. It is considered that the local features are more useful than global ones [8]. The KAZE features[1] is considered as local image properties and it is invariant to rotation, scaling, translation, and partial illumination changes [9]. In this paper, we propose a robust video watermarking method based on frame-patch matching for synchronization of embedding and extracting watermark. In this paper, we extract the KAZE features from a video frame and match them

---

[1] http://www.robesafe.com/personal/pablo.alcantarilla/kaze.html

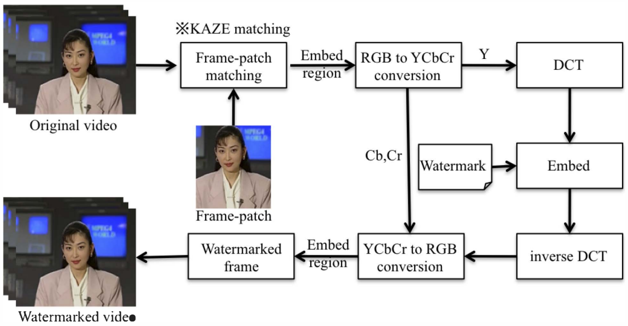


Fig. 1. Frame-patch matching based embedding system.

with those of frame-patch to detect the matched region in all frames. Using the matched region, we generate the embedding and extracting region for watermark information. In our method, the watermark is embedded into the matched region based on Discrete Cosine Transform (DCT) domain. To detect the embedded information in the video, we first detect the matched region by using KAZE feature matching. And by calculating the affine parameters, we can geometrically recover the video frame, and can easily extract the watermark information.

This paper is organized as follows. Section 2 introduces our proposed frame-patch matching based video watermarking. Section 3 presents the results of the experiments. Section 4 concludes the paper.

2. PROPOSED VIDEO WATERMARKING

In order to synchronize the embedded region and extracted region, we decide to choice the local image KAZE feature that is used to match the feature points of all frames and those of frame-patch. According to the matched KAZE feature points, the distorted video can be restored and the watermark information can be successfully extracted.

2.1. Frame-Patch Matching based Embedding

As shown in **Fig. 1**, we describe how to embed watermark information in our method. There are six steps in the scheme:

–Step 1: Extract one frame F from the original video V and extract the KAZE feature points of frame F and frame-patch F_p. After that, those feature points are matched each other to detect the region for embedding. The KAZE matching method is explained in Subsection 2.3.

–Step 2: Convert the RGB frame of the matched region to YCbCr color space.

–Step 3: Transform Y-component to a frequency domain using Discrete Cosine Transform (DCT).

–Step 4: Embed $W(i,j) \in \{0,1\}, 1 \leq i,j \leq L$ to Y-component in the frequency domain, where $L \times L$ is the size of watermark. $W(i,j)$ is converted into a linear array $W_l(k) = W(i,j), 1 \leq k \leq L^2$. One bit $W_l(k)$ is extracted to embed into DCT coefficient in frequency domain. The detailed explanation of this step is described in Subsection 2.4.

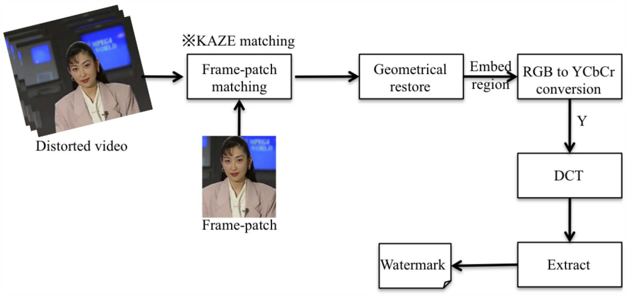


Fig. 2. Frame-patch matching based extraction system.

–Step 5: Compute the inverse DCT to obtain the modified Y and compose with the Cb and Cr components.

–Step 6: Convert the modified YCbCr frame to obtain the modified RGB frame.

Repeat Step 1 to Step 6 for all frames in video, we can obtain the watermarked video.

2.2. Frame-Patch Matching based Extraction

Fig. 2 describes how to extract the embedded information from the watermarked video by using KAZE feature points matching in frame-patch. This procedure consists of the following steps.

–Step 1: Extract one frame F' from the watermarked video V' and extract the KAZE feature points of it. Next, the feature points of frame-patch F_p is used to match with those feature points of F' and detect the embedded region.

–Step 2: Based on matched feature points, the rotation, scaling and translation parameters of distorted video is calculated (see Subsection 2.5). Then, the distorted video is restored.

–Step 3: Convert the RGB frame of the matched region to YCbCr color space.

–Step 4: Transform Y-component to a frequency domain using DCT.

–Step 5: Here, the embedded information $W_l(k)$ can be extracted from matched region. The detail explanation of this step will be described in Subsection 2.4.

Repeat Step 1 to Step 5 for all frames in video, we can get the watermark information from the watermarked video.

2.3. Frame-patch matching method

In order to synchronize the embedding region and extracting region, we have to find the common local features between the distorted video and the original video. By using the KAZE feature matching, we found that we can detect the matched feature for recovering the distorted video.

First, the KAZE features extracted from a target frame F are matched with those of a frame-patch F_p. Here, the Pablo's method [9] is used for matching.

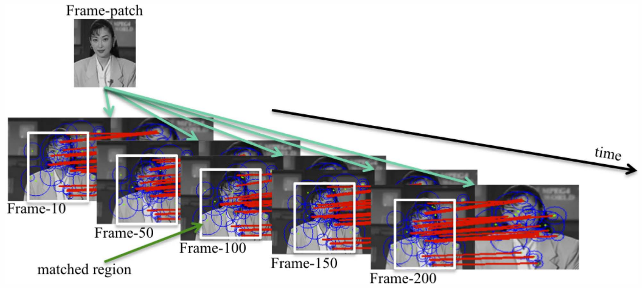


Fig. 3. Matched region using KAZE feature matching.

Suppose that the KAZE feature points p_l, q_k are extracted from frame F and frame-patch F_p, respectively:

$$\begin{aligned} p_l &= (x_l, y_l, \lambda_l, o_l, \boldsymbol{f}_l), \text{ for } l \in 1, \dots, L, && (1) \\ q_k &= (x'_k, y'_k, \lambda'_k, o'_k, \boldsymbol{f}'_k), \text{ for } k \in 1, \dots, K, && (2) \end{aligned}$$

where x_l, y_l, λ_l and o_l are, respectively, the x- and y-position, the scale and the orientation of the l-th detected feature point of frame F. The element $\boldsymbol{f}_l$ is a 64-dimensional or 128-dimensional local edge orientation histogram of the l-th point. The symbol " $'$ " denotes the parameters of KAZE feature points in F_p. In our matching experiment, we used 128-dimensional local edge orientation histogram.

To find the matched point, for each point p_l in F, we compute its distances d_{1l} and d_{2l} to its two nearest neighbors in F_p:

$$\begin{aligned} d_{1l} &= \underset{k}{\operatorname{argmin}} \|\boldsymbol{f}'_k - \boldsymbol{f}_l\|, && (3) \\ d_{2l} &= \underset{k \neq k_{\min}}{\operatorname{argmin}} \|\boldsymbol{f}'_k - \boldsymbol{f}_l\|, && (4) \end{aligned}$$

where $k_{\min}$ is the index of a feature point which had the minimum distance d_{1l}. Next, a ratio r_l is defined as $r_l = \frac{d_{1l}}{d_{2l}}$. Given a threshold τ, we can obtain a set of the matched points is $\mathbf{M} = \{(p_l, q_k) \mid r_l < \tau\}$. Using $\mathbf{M}$, we can detect the matched region from all frames for embedding and extraction performance as shown in **Fig. 3**. Note that, the matched region is defined that the rectangle which covered all matched points in F.

2.4. Embedding and Extraction algorithm

The embedding and the extraction method are performed on DCT frequency domain of the matched region. First, we segment the DCT coefficients into 8×8 blocks. In each block, two coefficients at (x_i, y_i) and at (y_i, x_i) are selected randomly from the 64 coefficients and their DCT coefficients $f(x_i, y_i)$ and $f(y_i, x_i)$ are modified with the watermarking strength a ($a > 0$):
When $W_l(k) = 0$,

$$\begin{cases} f'(x_i, y_i) = \dfrac{f(x_i, y_i) + f(y_i, x_i)}{2} - \dfrac{a}{2} \\ f'(y_i, x_i) = \dfrac{f(x_i, y_i) + f(y_i, x_i)}{2} + \dfrac{a}{2} \end{cases} \quad (5)$$

When $W_l(k) = 1$,

$$\begin{cases} f'(x_i, y_i) = \dfrac{f(x_i, y_i) + f(y_i, x_i)}{2} + \dfrac{a}{2} \\ f'(y_i, x_i) = \dfrac{f(x_i, y_i) + f(y_i, x_i)}{2} - \dfrac{a}{2} \end{cases} \quad (6)$$

In the information extraction, the embedded bit can be extracted by comparing $f'(x_i, y_i)$ and $f'(y_i, x_i)$, where f' indicates that corresponding distorted DCT coefficients: If $f(x_i, y_i) > f'(y_i, x_i)$ then $W'_l(k) = 1$, otherwise $W'_l(k) = 0$.

The watermark strength a effects to quality of watermarked video. Therefore, in our experiments, set the basis value of a to 0.15. This value will be increased or decreased based on the feature of the local region.

After extracting the watermark $W'_l(k)$, two dimensional watermark $W'(i, j)$ is formed from $W'_l(k)$ as,

$$W'(i, j) = W'_l(k), 1 \leq k \leq L^2, 1 \leq i, j \leq L \quad (7)$$

The watermark is extracted from all the frames. Suppose, there are F_N frames, the number of watermarks extracted at the receiver side will also be F_N . The final watermark W_v needs to be constructed from these F_N watermarks based on some decisions. In our method, voting method is made using maximum occurrence of bit value (either bit "0" or bit "1") corresponding to same pixel location in all extracted watermarks. This voting method can be represented by mathematical method using the following equation,

$$W_v(i, j) = voting(W'_f(i, j)), 1 \leq f \leq F_N, 1 \leq i, j \leq L \quad (8)$$

2.5. Estimation of RST parameters

According to the KAZE feature point matching, the rotation, scaling and translation parameter can be estimated based on some sets of matched points $\mathbf{M}$.

First, we can estimate the scale parameter Λ between the frame and frame-patch as follows:

$$\Lambda = \frac{\sum_{i=1}^{M} \lambda'_i}{\sum_{i=1}^{M} \lambda_i} \quad (9)$$

where λ_i and λ'_i are the scales of matched feature points of frame F and the frame-patch F_p, respectively.

Next, we estimate the angle α of rotation by using the feature points matched to each other. The rotation angle is estimated as follows:

$$\alpha = \frac{\sum_{i=1}^{M} (\alpha'_i - \alpha_i)}{M} \quad (10)$$

where α_i and α'_i denote the centre angle of the feature point i of frame-patch and that of the corresponding feature point i rotated distorted frame, respectively.

After adjusting the differences of scale and rotation, we calculate the translation parameters δx and δy, which corresponds to the differences in width and height, respectively. Let the coordinates of frame-patch feature points i is (x_i, y_i) and that of the corresponding distorted frame feature points (x'_i, y'_i). Then the translation parameters are estimated as follows:

$$\delta x = \frac{\sum_{i=1}^{M}(x'_i - x_i)}{M}, \delta y = \frac{\sum_{i=1}^{M}(y'_i - y_i)}{M} \tag{11}$$

3. EXPERIMENTAL RESULT

3.1. Experimental environment

3.1.1. System and datasets

To evaluate the proposed method fairly, we used different standard of three video sequences[2] (Akiyo: 300 frames, 25fps; Coastguard: 300 frames, 25fps; Carphone: 382 frames, 12fps) in QCIF format ($Width \times Height = 176 \times 144$). All experiments were performed in the Mac OSX 10.6.8 system. The watermark is denoted by a binary image with the size $L \times L = 64 \times 64$. GCC version 4.0.1[3] and the MPlayer version 1.1-4.2.1[4] were used to convert and view the experimental video data.

3.1.2. Evaluation measurement

In order to evaluate the performance of the proposed method, the transparency and the robustness for watermarking are used to measure the system performance.

The transparency is important measure to evaluate the watermarking. The peak signal-to-noise ratio (PSNR) and Structural Similarity Index Measure (SSIM) are used as a criterion to estimate the invisibility [10]. In these experiments, the PSNR and SSIM were calculated for every video sequences. Since PSNR and SSIM do not take the temporal activity into account, to compare the perceptual quality, the video quality metric (VQM)[11] is also employed. This metric is between zero and one; zero means not having any distortions while one shows maximum impairment. The original sequences and the watermarked sequences are used as the original and the processed clips.

For robustness, it is also an important factor in watermarking. A measure of the normalized correlation (NC) [12] calculated the difference between the extracted watermark $W_v(i, j)$ and the original watermark $W(i, j)$ is used to evaluate the performance system.

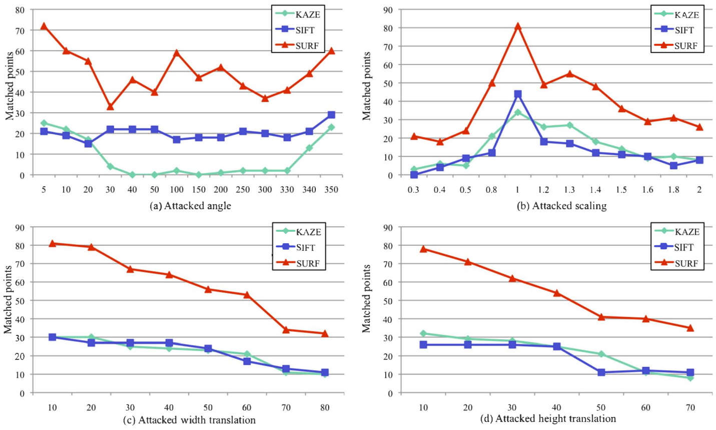


Fig. 4. Comparison of matched points of akiyo: (a) Clockwise rotation attacks with angle from 5^o to 350^o; (b) Scaling attacks with scale factor from 0.3 to 2.0; (c) Width translation attacks with δx from 0 to $Width/2$; (d) Height translation attacks with δy from 0 to $Height/2$. Best viewed in color.

3.2. Simulation results

3.2.1. Feature points matching

In order to evaluate the efficient of employing the KAZE feature, the rotation, scaling and translation attacks to video are taken into account and its feature points are matched with those of the frame-patch. We compared the number of matched points of KAZE with SIFT [14] and SURF [15], respectively.

As shown in **Fig. 4**, we observed that the number of the matched points based SURF feature (red) is larger than KAZE feature (green) and SIFT feature (blue). The reason for this is SURF feature vector has 64-dimension for matching, whereas SIFT feature vector and KAZE feature vector have 128-dimension. Therefore it is considered that the mismatching points of SURF feature are higher than others [9]. Then, the estimation of parameters from such mismatched points can be incorrect.

Since at least two matched points are required for geometrical recovery, we found that if we use KAZE feature, we can estimate the rotation parameters from -30^0 to $+30^0$ (Fig. 4(a)), the scaling parameters from 0.3 to 2.0 (Fig. 4(b)) and translation parameters from 0 to $Width/2$ for width translation and from 0 to $Height/2$ for height translation (Fig. 4(c) and (d)). In general, the cropping rotations are slight for video and the rotation angles are no more than 5 degrees. Therefore, the estimation of rotation angles from -30^0 to $+30^0$ is considered that is robust enough for rotation attacks. "$-$" and "$+$" denote the clockwise and the anti-clockwise of the rotation attacks.

---

[2] http://media.xiph.org/video/derf/ (Accessed on 12 Dec 2012)

[3] http://gcc.gnu.org/

[4] http://www.mplayerhq.hu/design7/dload.html

Table 1. Average PSNR[dB], VQM and SSIM value.

	[7]			Our method		
Video	PSNR	SSIM	VQM	PSNR	SSIM	VQM
Akiyo	35.42	0.93	0.22	**37.14**	**0.95**	**0.21**
Coas.	34.95	0.96	0.05	**36.46**	**0.98**	**0.04**
Carp.	35.31	0.94	0.06	**36.93**	**0.96**	**0.06**

3.2.2. Estimation attacked parameters

In order to geometrically recovery, we estimated the geometrical parameters (RST) from matched points. First, we implemented rotation, scaling and translation attack, respectively. **Fig. 5** shows the actual parameters "Truth" and the estimated parameters of KAZE, SIFT and SURF, respectively.

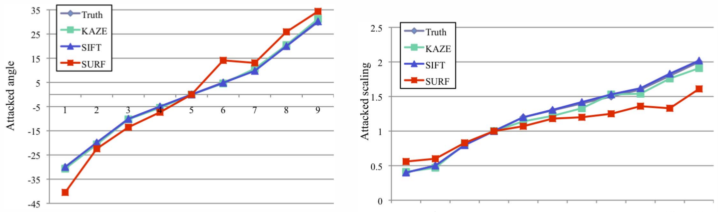
In the estimated results of rotation attack (Fig. 5(a)), we saw that we can estimate the rotation angle from -30^o to 30^o as SIFT feature. There is large error estimation if we use SURF feature in the rotation attack. The average error rotation estimations of KAZE, SIFT and SURF are 0.54^o, 0.11^o and 4.57^o, respectively. In the Fig. 5(b), we recognized that the estimated scaling factors are estimated close to "Truth" factor by using KAZE and SIFT. However, **if scaling factor equals 0.3, the scale parameter can not be estimated by SIFT and SURF, whereas it can be estimated successfully by using KAZE**. The average error scaling estimations of KAZE, SIFT and SURF are 0.05, 0.01 and 0.23. In the translation estimated results (Fig. 5(c) and (d)), we found that we can estimate the translation factor exactly from 0 to $Width/3$ and from 0 to $Height/3$ by using KAZE; from 0 to $Width/2$ and from 0 to $Height/2$ by using SIFT.

According to our estimated result, SIFT feature, which is applied in [7], is better than another ones. However, the KAZE feature is considered to be enough for robust video watermarking and it can ensure the critical robustness of watermarking technique.

3.2.3. Quality evaluation

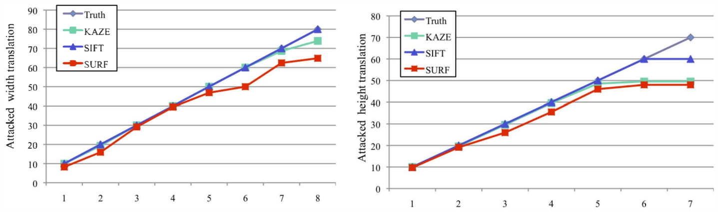
As shown in **Table 1**, the average PSNR values of three watermarked video sequences are 37.14, 36.46, and 36.93 decibels (dB), separately. All the values are higher than 36 dB. In addition, the average SSIM values of those are 0.95, 0.98, and 0.96, respectively. All the values are also very close to 1.0. Perceptually, the original video and the watermarked video are visually indistinguishable. This implies that the watermarking scheme can achieve visual transparence. By observing the VQM values of three watermarked videos, we confirmed that the videos, which have more motion (coastguard and carphone), are degraded litter than ones which have less motion (akiyo).

The performance of our method is compared with [7]. We can see that our approach yields much better results than [7].



(a) Estimation of attacked angle

(b) Estimation of attacked scaling



(c) Estimation of attacked width translation

(d) Estimation of attacked height translation

Fig. 5. Comparison of estimated parameters of akiyo: (a) Estimated rotation attacks with angle from -30^o to 30^o; (b) Estimated scaling attacks with scale factor from 0.4 to 2.0; (c) Estimated width translation attacks with δx from 0 to $Width/2$; (d) Estimated height translation attacks with δy from 0 to $Height/2$. Best viewed in color.

3.2.4. Evaluation of robustness

We implemented a video watermarking attack tool named VirtualDub[5] program to modify the watermarked videos with various types of attacks. We also applied the Vidmark Benchmark [13] to simulate the temporal desynchronization attacks. We considered the watermarking scheme to be robust if the computed NC is over than 0.9.

In general, the cropping rotations are slight for video signals and the rotation angles are no more than 5 degrees. We applied the rotation with cropping attack angles increase from -20 to $+20$ degrees, and confirmed that the NC values remain over than 0.9. In our experiments, we also applied several scale factors (from 0.3 to 1.2) to the watermarked video sequences.

Table 2 shows the experimental results of each robustness measure: cropping rotations, scaling, frame dropping, frame insertion, frame transposition, frame averaging, blur and gaussian. We also compared our proposed method with [7]. As it can be observed, when scale factor equals 0.3, method of [7] can not detect the matched region, whereas our method can detect the matched region. According to these results, the NC values are over than 0.9, indicating that our algorithm is robust to these attacks.

4. CONCLUSION

In this paper, we applied the KAZE feature matching to develop a robust geometrically invariant video watermarking. By doing so, we can estimate the geometrical parameters for

---

[5] http://www.virtualdub.org/

recovering the distorted video and we can synchronize the embedded region for watermarking extraction. The experimental results show that our proposed method can resist to very strong attacks such as 0.3x scaling, from -20^0 to $+20^0$ angle rotation, video processing attacks and some temporal attacks. In addition, we also applied the redundant embedding method to all frames, then we can get much better results for the video watermarking. It is robust against to a wide variety of attacks and it is suitable for many applications requiring high watermarking reliability and capacity.

In the future work, we want to improve the computation cost to apply our method for realtime video watermarking.

Table 2. Experimental results of robustness.

	NC (Our method / [7])		
Attack	Akiyo	Coas.	Carp.
No attack	1.0/1.0	1.0/1.0	1.0/1.0
Rotation w. cropping (-20^0)	0.98/1.0	1.0/1.0	1.0/1.0
Rotation w. cropping (-10^0)	1.0/1.0	1.0/1.0	1.0/1.0
Rotation w. cropping (-5^0)	0.99/1.0	1.0/1.0	1.0/1.0
Rotation w. cropping ($+5^0$)	1.0/1.0	1.0/1.0	1.0/1.0
Rotation w. cropping ($+10^0$)	1.0/1.0	1.0/1.0	1.0/1.0
Rotation w. cropping ($+20^0$)	1.0/1.0	1.0/1.0	1.0/1.0
Scaling 0.3	**0.69/–**	**0.70/–**	**0.69/–**
Scaling 0.5	0.91/0.92	0.90/0.90	**0.91**/0.90
Scaling 0.8	1.0/1.0	1.0/1.0	1.0/1.0
Scaling 1.2	1.0/1.0	1.0/1.0	1.0/1.0
Frame dropping 10%	0.70/0.70	0.68/0.68	0.66/0.69
Frame dropping 20%	0.69/0.69	**0.70**/0.69	**0.69**/0.66
Frame insertion 10%	**0.70**/0.69	**0.71**/0.68	**0.68**/0.63
Frame insertion 20%	**0.70**/0.69	**0.70**/0.69	**0.70**/0.69
Frame transposition 10%	1.0/1.0	1.0/1.0	1.0/1.0
Frame transposition 20%	1.0/1.0	1.0/1.0	1.0/1.0
Frame averaging 10%	1.0/1.0	1.0/1.0	1.0/1.0
Frame averaging 20%	1.0/1.0	1.0/1.0	1.0/1.0
Blur	0.99/1.0	0.98/1.0	0.99/1.0
Gaussian 3x3	0.93/1.0	0.94/1.0	0.93/1.0

5. REFERENCES

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, “Attacks on copyright marking systems,h in Proc. Int. Workshop on Information Hiding, pp. 218–238, Springer-Verlag, 1998.

[2] P. Bas, J.-M. Chassery, and B. Macq, “Geometrically invariant watermarking using feature points,h IEEE Trans. Image Process. Vol. 11, No. 9, pp.1014–1028, 2002.

[3] A. Nikolaidis and I. Pitas, “Region-based image watermarking,h IEEE Trans. Image Process. Vol. 10, No. 11, pp.1726–1740, 2001.

[4] H. Dajun, Q. Sun, Q. Tian, “A secure and robust object-based authentication system,h EURASIP J. on Applied Signal Processing, vol.14, pp.1–14, 2004.

[5] Y. K. Ho and M. Y. Wu, “Robust object-based watermarking scheme via shape self-similarity segmentation,h Pattern Recognition Letters, vol.25 no.15, pp.1673–1680, 2004.

[6] J. S. Lee and W. Y. Kim, “A New Object-Based Image Watermarking Robust to Geometrical Attacks,h Proc. the 5th Pacific Rim Conf. on Advances in Multimedia Information Processing, pp.58–64, 2004.

[7] P. Q. Viet, T. Miyaki, T. Yamasaki, K. Aizawa, “Robust Object-based Watermarking Using Feature Matching,h IEICE Trans. Information and Systems, vol. E91-D, no. 7, pp. 2027–2034, 2008.

[8] Lee, H.Y., Kim, H. and Lee, H.K., “Robust image watermarking using local invariant features,” Journal of SPIE Optical Engineering. v.45, i3. 037002-1-037002-11.

[9] Pablo F. Alcantarilla, Adrien Bartoli and Andrew J. Davison, “KAZE Features,” In European Conf. on Computer Vision (ECCV), Fiorenze, Italy, 2012.

[10] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Processing, vol. 13, no. 4, pp. 600–612, 2004.

[11] M. Pinson and S. Wolf, “A new standardized method for objectively measuring video quality,h IEEE Tran. Broadcasting, 2004.

[12] H.Y. Huang, C.H. Yang and W.H. Hsu, “A video watermarking technique based on pseudo-3-D DCT and quantization index modulation,h IEEE Trans. Information Forensics and Security, vol.5, No.4, pp.625–637, 2010.

[13] A. H. Pedro, F. Claudia, R. Cumplido and J. J. Garcia-Hernandez, “Towards the Construction of a Benchmark for Video Watermarking Systems: Temporal Desynchronization Attacks,” Proc. of the 53nd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 628–631, 2010.

[14] D. Lowe, “Distinctive image features from scale-invariant keypoints,” Intl. J. of Computer Vision Vol. 60, pp.91–110, 2004.

[15] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, “SURF: Speeded up robust features,” Computer Vision and Image Understanding, Vol. 110, pp. 346–359, 2008.