2017 International Conference on Computer Network, Electronic and Automation (ICCNEA 2017)

Multi - scale Target Tracking Algorithm with Kalman Filter in Compression Sensing

Yichen Duan
Department of Electronic and Information Engineering
Xi`an China
13488232750@163.com,wp_xatu@163.com

Xue Li
Department of Electronic and Information Engineering
Xi`an China
13488232750@163.com,wp_xatu@163.com

Peng Wang
Department of Electronic and Information Engineering
Xi`an China
13488232750@163.com,wp_xatu@163.com

Dan Xu
Department of Electronic and Information Engineering
Xi`an China
13488232750@163.com,wp_xatu@163.com

***Abstract*—Real-time Compressive Tracking (CT) uses the compression sensing theory to provide a new research direction for the target tracking field. The algorithm is simple, efficient and real-time. But there are still shortcomings: tracking results prone to drift phenomenon, cannot adapt to tracking the target scale changes. In order to solve these problems, this paper proposes to use the Kalman filter to generate the distance weights, and then use the weighted Bayesian classifier to correct the tracking position, and perform multi-scale template acquisition in the determined position to adapt to the changes of the target scale. Finally, introducing the adaptive learning rate while updating to improve the tracking effect.. Experiments show that the improved algorithm has better robustness than the original algorithm on the basis of maintaining the original algorithm real-time.**

Keywords-compression sensing; CT; multi-scale; Kalman filter

I. INTRODUCTION

Target tracking is the core research content in the field of machine vision. It has a wide range of applications in human-computer interaction, video surveillance, scene comprehension and behavior recognition. In recent years, domestic and foreign scholars have proposed a variety of tracking methods, such as target-based, regional matching tracking algorithm, but these algorithms in the practical application of the situation in the poor robust, easy to track failure. Most of the current tracking problems using background and target binary classification of ideas, that is tracking-by-detection.

Compressed sensing theory is a kind of signal expression based on sparse expression in recent years, which has great influence in mathematics and engineering application. Has been applied to wireless communications, image processing, pattern recognition in many areas. Kaihua Zhang et al[8]. Applied the compression perceptual theory to the target tracking problem. The algorithm is simple and efficient, and real-time，which provides a new research direction for the target tracking field, but the algorithm still has some shortcomings .

Compression sensing algorithm in the tracking process tracking frame scale unchanged, easy to introduce background error, resulting in tracking box drift, the final tracking failure. Aiming at this problem, a simple scale transformation method is proposed, which can adapt to the real-time performance at the same time. Compression tracking algorithm is a typical tracking-by-detection framework, once the target part of the block, must introduce the error, resulting in drift, the location is difficult to restore. In response to this problem, this paper proposes the use of Kalman filter tracking box position correction. Compression tracking classifier update using fixed update mode, that is, a fixed learning rate, this approach will inevitably lead to update rate cannot adapt to the target changes, at the same time easy to introduce errors, and then drift. This paper presents a relatively simple way to update, can be better adaptive target changes.

II. REAL-TIME COMPRESSIVE TRACKING

The compression perceptual tracking algorithm is a tracking algorithm based on the compression perceptual theory [1]. The compression perceptual theory[2] states that if a signal can be compressed and the random perceptual matrix satisfies the Johnson-Lindenstrauss inference[3], there is a higher probability that the Y is reconstructed by X .

$$V = \phi X \quad \phi \in R^{m \times n} (m << n) \tag{1}$$

The compression tracking uses the formula (1) to extract the target feature[4], X is the characteristic matrix of the target high dimension, ϕ is the random perception matrix, V is the low-dimensional characteristic matrix of the target, and its sparse Chengdu depends on the sparse degree of ϕ. The compression tracking uses a very sparse and satisfying Johnson-Lindenstrauss inference of a random projection matrix defined as follows:

$$v_{ij} = \sqrt{s} \times \begin{cases} 1 & with \quad prob \quad \frac{1}{2s} \\ 0 & with \quad prob \quad 1-\frac{1}{s} \\ -1 & with \quad prob \quad \frac{1}{2s} \end{cases} \tag{2}$$

In the formula, s randomly selected from 2 to 4. Compression feature extraction target tracking algorithm is used with similar Hear-Like relative difference feature. Each element of the low-dimensional feature is a linear combination of spatial distributions of different scales.

After the compression feature is extracted, the compressed features are entered into the naive Bayesian classifier to distinguish the background and the target. The target low dimension represents $V(z) = (v_1, ..., v_n)^T \in R^n$, assuming that each element is independent of each other, naive Bayesian model:

$$H(v) = \log\left[\frac{\prod_{i=1}^{n} p(v_i \mid y=1) p(y=1)}{\prod_{i=1}^{n} p(v_i \mid y=0) p(y=0)}\right] = \sum_{i}^{n} \log\left(\frac{p(v_i \mid y=1)}{p(v_i \mid y=0)}\right) \tag{3}$$

Provable high-dimensional random vector mathematically random projection is always in line with the Gaussian distribution [5]. Thus the conditional distributions $p(v_i \mid y=1)$ and $p(v_i \mid y=0)$ in the classifier conform to the Gaussian distribution.

$$p(v_i \mid y=1) \sim N(\mu_i^1, \sigma_i^1) \tag{4}$$

$$p(v_i \mid y=0) \sim N(\mu_i^0, \sigma_i^0) \tag{5}$$

In the formula, μ_i^1 and σ_i^1 represent the mean and standard deviation of the $i-th$ feature of the positive sample, respectively. μ_i^0 and σ_i^0 are the mean and standard deviation of the $i-th$ feature of negative samples, respectively. The maximum response value position is the most likely target location. After determining the maximum corresponding position, the relevant parameters can be updated from the adaptation background and target changes. Parameter updating formula is:

$$\mu_i^1 \longleftarrow \lambda\mu_i^1 + (1-\lambda)\mu^1 \tag{6}$$

$$\sigma_i^1 \longleftarrow \sqrt{\lambda(\sigma_i^1)^2 + (1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2} \tag{7}$$

In the formula, λ is the learning rate , $\lambda > 0$ and λ is constant, the value reflects the speed of the updating. μ^0 and σ^0 update the same as above.

III. IMPROVEMENT OF COMPRESSION TRACKING

A. Combined with Kalman filter correction

Compression tracking classifier formula (3) to determine the target area is the way to all the probability of a simple sum, the response to the maximum value of the region is the target location. But sometimes the characteristics of the target and background characteristics are similar, it is not conducive to the classification of the classifier. Assuming the true position of each frame, it is clear that the closer to the real position, the greater the probability. At this point can be generated distance weight, add it to the classifier, improve the classifier classification performance, to enhance the credibility of the classifier. Of course, the real location of the target cannot be known in advance, you can use the forecast position instead of the real location. Maintaining the Integrity of the Specifications.

Kalman filter is a linear system state equation, through to the system input and output data, the optimal estimation of the system state is presented. The Kalman filter can be used to estimate the target position of the next frame using the current motion target parameters [7].

$$X(k \mid k-1) = AX(k-1 \mid k-1) + BU(k) \tag{8}$$

In the formula, $X(k \mid k-1)$ is the upper frame target motion parameter. $X(k-1 \mid k-1)$ is the target tracking result for the last frame. $U(k)$ is the control amount, set to 0.

$$P(k \mid k-1) = AP(k-1 \mid k-1)A + Q \tag{9}$$

In the formula, $P(k \mid k-1)$ is the covariance corresponding to $X(k \mid k-1)$, and Q is the covariance of the target tracking system. The Kalman filter combines the predicted position of the current target with the current target. Calculate the optimal estimate of the position of the target $X(k \mid k)$. $K_g(k)$ is Kalman gain.

$$X(k \mid k) = X(k \mid k-1) + K_g(k)(Z(k) - HX(k \mid k-1) \tag{10}$$

After the predicted position obtained by the Kalman filter, the distance of each sample is measured to obtain the position weight, and the position weight is added to the classifier. Define the distance from the sample to the target location:

$$l_i = \sqrt{\left[o_{\det}^k(x) - o_{pre}(x)\right]^2 + \left[o_{\det}^k(y) - o_{pre}(y)\right]^2} \tag{11}$$

In the formula, $o_{pre}(x)$ and $o_{pre}(y)$ predict the x coordinate and y coordinate of the target position by Kalman filter respectively. $o_{\det}^k(x)$ and $o_{\det}^k(y)$ are the x coordinates and y coordinates of the $i-th$ sample, respectively. In order to reduce the impact of noise on the weight of the reference[6], the normalization function is a hyperbolic tangent function. The normalized position weights are:

$$w_i = \frac{1}{2}\left\{\tanh\left[0.01\times\left(\frac{\frac{1}{l_i}-\mu}{\sigma}\right)+1\right]\right\} \tag{12}$$

In the formula, $1/l_i$ indicates that the distance from the predicted distance of the sample is, the smaller the probability that the sample is judged as the target, the smaller the position weight w_i is. When the sample distance is closer to the predicted position, the opposite is true. μ and σ are the mean and variance of $1/l_i$ respectively. The position weight is introduced into the formula (3):

$$H^{'}(v) = \log\left[\frac{\prod_{i=1}^{n} w_i p(v_i \mid y=1)p(y=1)}{\prod_{i=1}^{n}(1-w_i)p(v_i \mid y=0)p(y=0)}\right] = \sum_i^n \log\left(\frac{w_i\, p(v_i \mid y=1)}{(1-w_i)p(v_i \mid y=0)}\right) \tag{13}$$

By the normalization of $\tanh$, when w_i is 0.5, it does not affect the classifier. w_i greater than 0.5 when the $p(v_i \mid y=1)$ value becomes larger, less than 0.5 on the contrary. That is, when the predicted position is closer to the target, the W value is larger, and the closer the target response value is, the better the background and the target will be. This weighted Bayesian classifier improves the performance of the classifier and enhances the reliability of the classifier.

B. *Scale Processing*

The compression algorithm is fixed in scale, i.e., the size of the tracking window is constant. The theoretical analysis shows that when the target becomes larger, the negative sample acquisition area and the target area are too close, so that the generated classifier can reduce the performance of the target background. When the target becomes smaller, it is easy to introduce the background error, resulting in tracking box drift.

When the target position is obtained, it is assumed that the position is accurate and the scale is suitable and no error is introduced. The secondary acquisition can be carried out at this position. The acquisition is mainly carried out at this position for multiple scale templates, and the classifier is used again. Get the maximum response value, that is, the result of scale transformation.

This method assumes that the target does not undergo dramatic changes in the scale during the tracking process. So the collection of different scales of the size of the rectangular box can be in the up and down about four directions can be outward to the outside of the proportion of amplification and reduction. In this paper, the scale is chosen to be between 0.5 and 1.5, with an interval of 0.05. That is, you can collect a total of 20 different sizes of templates.

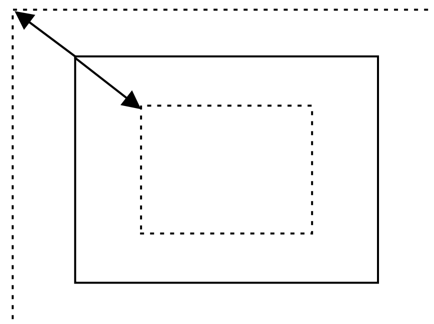

Figure 1. Scale Change

After the tracking frame scale changes, high-dimensional feature vectors cannot be mapped into low-dimensional space. In order to solve this problem, the essence of the nonzero term in the random sparse matrix is to sample the pixels in the tracking frame. After scale transformation, it is necessary to ensure that the position of the pixel sample and the relative position of the tracking frame remain unchanged. In practice, the random sparse matrix not only records the randomly generated weights, but also records the position of the feature expression. The operation of the feature is similar to that of the tracking frame, and the ratio of the scale and the scale of the tracking frame is the same in the four directions of the upper, lower, left and right directions, and the new random sparse matrix is obtained.

The above analysis shows that the new tracking box scale transformation method is simple. The new random sparse matrix is linearly transformed on the original random sparse matrix, and the overall computation is not large, and the real-time effect of the algorithm is limited. The scale transformation of this method can adapt to the change of target scale on the basis of keeping the algorithm real-time.

C. *Update Improvement*

From the perspective of algorithm design, the algorithm needs to be able to adapt to the appearance of the target to a certain extent. As can be seen from the above algorithm, the compression sensing algorithm will re-acquire the positive and negative samples and update the classifier after finding the "target position" of the frame. This approach is, to some extent, the appearance of the adaptive tracking target.

However, once the target is partially blocked, the background is complex, it will inevitably introduce noise and background error. In the follow-up of the process of tracking the phenomenon of drift, with the drift of the phenomenon of accumulation, and ultimately tracking failure.

Equation (6) (7) analysis shows that the new parameter model has two parts. The first part is the model of the previous frame, which represents the stability of the target. The second part is based on the current frame of the target for the collection of positive and negative samples to learn the new model, represents the goal of change. The new and old models are linearly combined to form an updated classifier parameter model. The learning rate of the compression perceptual tracking algorithm is a fixed value. When the new template is suspicious, it cannot effectively suppress the update.

This paper proposes a learning rate that can be adapted to a target change. The histogram is calculated for the maximum response position of the classifier, and the distance between the previous frame and the current frame processing result is calculated using the Bhattacharyya distance.

$$L = \sum_{i=1}^{n} \sqrt{p_i q_i} \quad L \in (0,1] \tag{14}$$

The closer the distance, the higher the matching degree of the two images, on the contrary, the lower. Set a threshold at that time, you can update, on the contrary, it means that the current frame and the previous frame of the image difference is large, then refused to update to avoid the introduction of error. The new learning rate is:

$$\lambda' = \lambda / L \tag{15}$$

In the formula, A is the learning rate, and B is the Bhattacharyya distance of the previous frame and the current frame processing result. The larger the results are similar, the smaller the learning rate is needed, the smaller the difference is, the larger the learning rate is. In this way, adaptive control classifier learning rate. Because only need to calculate the previous frame and the next frame of the results of the histogram, and then Bhattacharyya distance operation, the overall calculation of the amount did not significantly improve, in keeping the algorithm on the basis of real-time better adapt to the appearance of the object changes.

D. Algorithm

Multi - scale Target Tracking Algorithm with Kalman Filter in Compression Sensing

Input: the $t-th$ image frame
1: Collect the image set for the first time and use the classifier to determine the target position.
2: The Kalman filter corrects the target position.
3: Scale transformation, and change the number of random sparse matrix columns, to ensure that the high-dimensional feature vector mapping to low-dimensional space.
4: The second collection of image sets, and extract low-dimensional features, update the classifier parameters.
Output: Tracking target location

IV. Experiment and Conclusion Analysis

A. Tracking Effect Analysis

This algorithm simulation platform for Visual Studio 2013, release mode, call opencv visual library programming, version number: 2.4.13. Tests use the same computer, and the test sequence comes from the common test set.

In this paper, the first group of test sequence name box, the background is complex, the target part of the occlusion and appearance changes. At the time of sequence 321, when the tracking target is partially blocked, the compression tracking algorithm begins to drift, and the 339th frame compression tracking algorithm fails to track completely. And the use of this algorithm, 324 objects occlusion can also be more accurate tracking to the target, in the first 339 series can continue to track the target. When the algorithm is improved, the tracking failure caused by the occlusion of the object is greatly reduced.

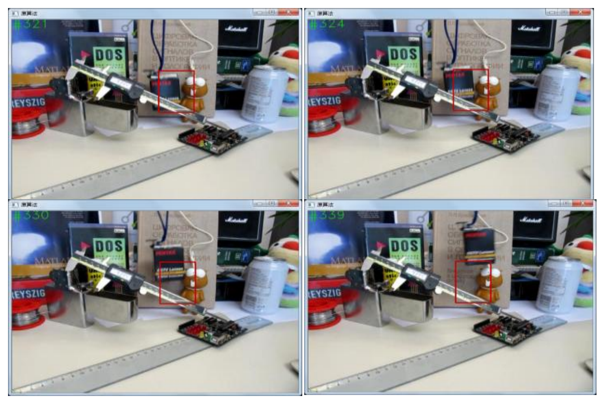

Figure 2. Box sequence of the original algorithm 321,324,330,339 frames

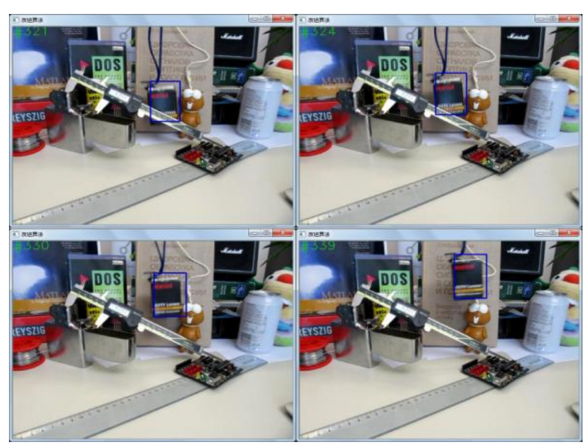

Figure 3. Box sequence of the improve algorithm 321,324,330,339 frames

In this paper, the second group of test sequence name Walking2, the background is more single, the target part of the block, in the tracking process has a similar target interference. In the 29th frame of the sequence, similar objects appear at 210 frames, and the similar target is partially blocked when the target is about 210 frames. At this time, the original compression tracking algorithm starts to track and drift, 220 frames are completely disturbed by similar target, 230 frame tracking algorithm is tracked failure. The use of this algorithm, in the 210 frame when the target is partially blocked in the case, to continue to track, 230 can also continue to track. After the algorithm is improved, the robustness of similar interference is improved.

Figure 4. Walking2 sequence of the original algorithm 195,210,220,230 frames

Figure 5. Walking2 sequence of the improve algorithm 195,210,220,230 frames

B. Date Analysis

After the above comparison using the sequence image, the following two kinds of algorithms for quantitative analysis, using two analysis methods. The first is the algorithm running speed comparison table 1, the unit is the number of transmission frames per second fps. Similar to the theoretical analysis, the speed of this algorithm is slightly lower than the original algorithm. The improvement scheme proposed in this paper basically maintains the real-time performance of the original algorithm. As can be seen from the data, for different image sequences are reduced, but the reduction is not significant.

The second is to track the success rate, the public test set to provide the tracking target real location and algorithm tracking position to compare. In the same sequence of test environment, tracking success rate has greatly improved, and enhanced the robustness of the algorithm.

TABLE I. ALGORITHM RUNNING SPEED COMPARISON

Sequence	CT	Improved
Box	69fps	64fps
Walking2	70fps	66fps

TABLE II. TRACKING SUCCESS RATE COMPARISON

Sequence	CT	Improved
Box	21.53%	73.55%
Walking2	20.15%	32.31%

V. CONCLUSION

The algorithm is mainly based on the compression-aware tracking algorithm, which is based on the improvement of the algorithm. In order to maintain the excellent real - time performance, the Kalman filter is introduced to improve the tracking effect of the tracking target scale transformation and partial occlusion by simple scale transformation. At the same time, it improves the parameter updating mechanism of the algorithm classifier, effectively suppresses the iterative accumulation of errors and improves the robustness of the algorithm. Experiments show that this algorithm is effective.

REFERENCES

[1] Donoho D. Compressed sensing [J]. IEEE Transactions on Infor-mation Theory, 2006, 52(4): 1289-1306.

[2] Candes E, Tao T. Near optimal signal recovery from random projections and universal encoding strategies [J]. IEEE Transactions on Information Theory(S0018-9448), 2006, 52(4) : 5406-5425.

[3] Achlioptas D. Database-friendly random projections : Johnson-Lindenstrauss with binary coins [J]. Journal of Computer and System Sciences(S0022-0000), 2003, 66 : 671-687.

[4] Baraniuk R, Davenport M, De Vore R, et al. Wakin M. A simple proof of the restricted isometry property for random matrices [J]. Constructive Approximation(S0176-4276), 2008, 28 : 253-263.

[5] Diaconis P, Freedman D. Asymptotics of graphical projection pursuit [J]. The Annals of Statistics(S0090-5364), 1984, 12(3) : 228-235.

[6] Jain A, Nandakumar K, ROSS A. Score normalization in multimo-dal biometric systems[J]. Pattern Recognition, 2005, 38 (12):2270-2285.

[7] MAZINAN A H, AMIR-LATIFI A. Applying mean shift, mo-tion information and Kalman filtering approaches to object tracking [J]. ISA transactions, 2012, 51(3) : 485-497.

[8] ZHANG Kaihua, ZHANG Lei, YANG Ming-Hsuan. Real-Time Compressive Tracking [C]. Proceedings of the 12th European conference on Computer Vision, Florence, Italy, Oct 8-11, 2012, 3 : 866-879.