Research on target tracking algorithm based on Kernel Correlation

Shengbo Liu[1,2], Yi Guo[1], Yandong Zhao[1,3]*
[1]School of Technology, Beijing Forestry University, Beijing 100083, China
[2]Beijing Laboratory of Urban and Rural Ecological Environment, Beijing 100083, China
[3]Center for Intelligent Forestry Research, Beijing Forestry University, Beijing 100083, China

Received: November 15, 2021. Revised: May 21, 2022. Accepted: June 22, 2022. Published: July 26, 2022.

Abstract- With the development of sensor and image processing technology, computer vision plays an increasingly significant role in the chemical engineering because of its characteristics such as low cost, high resolution, and non-contact measurement. In this paper, the motion probability map can be obtained by sparse optical flow based on Harris corner point. Then the coarse contour of silicon dioxide particles which is the input of kernelized correlation filtering (KCF) algorithm can be generalized. KCF algorithm can easily complete tracking task under the influence of disturbance including light change, video shaking and so forth. A contour refining and tracking method are proposed. The geometric active contour (GAC) algorithm can use function as implicit expression of contour and can design the different energy functional to control contour evolution. By minimizing of energy functional, the refining contour is evolved. Then the target tracking is realized according to the refined contour.

Keywords- moving object detection, target tracking, contour extraction, correlation filtering, geometric active contour.

I. Introduction

Visual information accounts for a major amount of the information collected by humans. It can also assist people in better understanding and changing the world. Computer vision technology aims to improve the intelligence that allows computers to analyze and understand visual information to replace certain human labor. Under normal circumstances, the computer must first add a sensor to obtain the image, and then process the image signal using its own hardware or software resources to extract the image's critical information, such as the type of target, target position, and other high-level data. The general diagram is shown in figure 1.

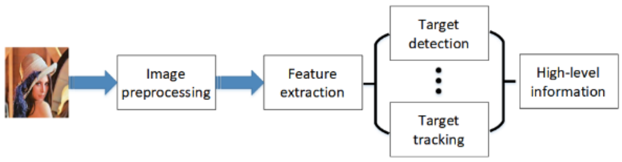

Figure 1. The general diagram of computer visual

The interesting part of an image or a set of video frames is the target. The contoured target region is always distinct from the background region. The split of the target region is identical to the extraction of the target contour [1,2].

The split of the target region is identical to the extraction of the target contour. The goal of target tracking is to find the target in each image of a video series. Some study calculates the similarity function between the candidate contour and the known contour using the specified features, and then determines the target contour for the next frame. Traditional approaches include the matching tracking algorithm [3], the tracking algorithm employing color and edge statistics [4], and the Hough transform method. Other research leverage existing filtering algorithm results [5,6], create a motion model for the contour, and apply the filtering algorithm to monitor the target contour. Other scholars also use denoising algorithm for filtering [7,8,9] and Biorthogonal wavelet trees [10,11] The contour tracking approach based on geometric active contour (GAC) can adapt well to the target contour's free deformation as well as the contour changes produced by target self-occlusion. The discriminant algorithm, in contrast to the Generative Algorithm [12], introduces background information, allowing the target tracking algorithm to focus on separating the target from the backdrop without being distracted by the part of the background that looks like the target. The classifier is at the heart of the discriminant algorithm. The varied training sets primarily reflect the different classifiers. After many affine transformations, the Mosse approach based on single feature [13] combines the known target with the single channel feature of the target picture. The kernel methodology and 31-dimensional Felzenszwalb Histogram of Oriented Gradient (FHOG) feature are used in the KCF method, which is based on the principle of cyclic sampling and frequency domain simplified calculation [14]. By combining the properties of multiple tiers of convolutional neural networks, C-cot [15] and later upgraded eco algorithm [16] improve tracking accuracy even more, although their frame rate is limited to a few frames per second. Dsst [17] and SAMF algorithm[18] with scale filter can not only obtain the target frame, but also the size of the target frame will change with the change of the actual target scale.

II. Object Detection Particles Detection and Kernelized Correlation Filter Tracking

For intuitiveness, the color image and grayscale image after preprocessing are shown in figure 2 and figure 3.

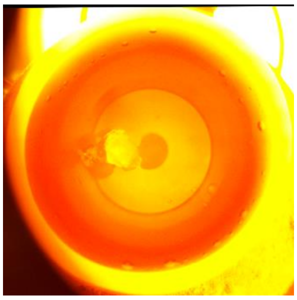

Figure 2. Color image

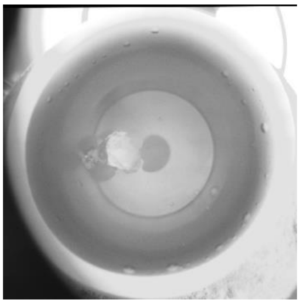

Figure 3. Gray-scale image

Kanade-Lucas-Tomasi (KLT) tracking uses Harris corner as a feature point [19]. There are a few causes for this. KLT is a registration algorithm for images. The bend has a large gradient change as well as a large gradient direction shift. Typically, a corner is detected when two edges connect, and the gray level of each direction in the corner neighborhood changes. The image from the preceding technique can easily get good image registration results. Figure 4 depicts the characteristic point results derived by Harris corner detection.

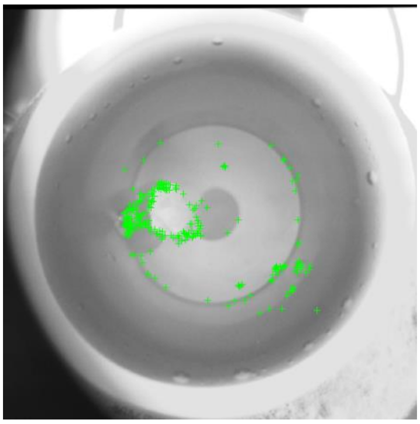

Figure 4. Harris corner detection results

To discriminate between target and background feature points, K-means clustering is performed. Figure 5 and 6 demonstrate the classification results of feature points. Figure 5 shows a schematic model of optical flow feature space classification, while figure 6 shows a schematic diagram of feature point classification in a real image.

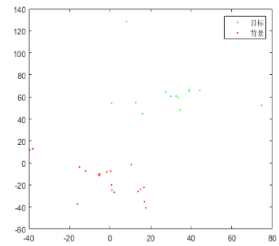

Figure 5. Feature space classification results

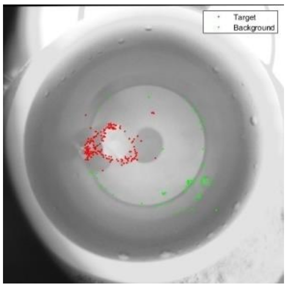

Figure 6. Classification results of real image feature points

To solve an accurate affine motion model, the Expectation Maximization (EM) clustering method was used. The posterior probabilities of all pixels pertaining to the target and background are calculated using the two motion models of the target and background established in the previous phase, and the motion probability map is created.

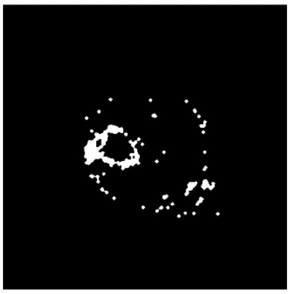

Figure 7. Binarization

Figure 8. Remove small objects

As shown in figures 7 and 8, the resultant motion probability map is binarized and small items are deleted, then the convex hull of the contour is calculated to get the rough contour, as shown in figure 9.

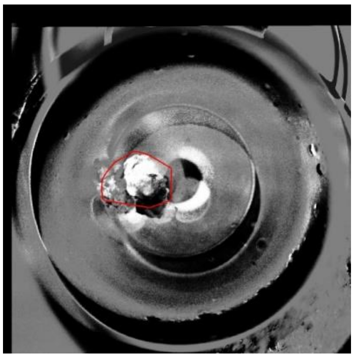

Figure 9. Rough outline

III. TARGET TRACKING ALGORITHM BASED ON KERNELIZED CORRELATION FILTERING

The goal of this section is to discuss the kernel correlation filters-based target tracking algorithm (KCF algorithm)[20]. Between a video series and the first frame, the position and size information of the external rectangle is known, i.e., the position of the target is retrieved in the first frame. A zone of interest with a certain background is created by extending the target's external rectangle, and the features of the region of interest are used to train a classifier.

The sample of this classifier is each row of the circulant matrix. The tag is the Gaussian similarity function between the sample and the real target. In the next frame, we usually use the classifier to detect the scores of each candidate image block. This algorithm has a fast detection method. The image block with the highest score is considered as the target of this frame. Update the classifier according to a certain strategy, and then continue tracking.

A. *Linear ridge regression classifier training*

In this section, ridge regression will be discussed, and the initial tracking time image is shown in figure 10. It has a simple closed-form solution, and can achieve some classifier, such as Support Vector Machines. The purpose of training is to find a function $f(z) = w^T z$ that minimizes the square error between sample x_i and ridge regression label y_i.

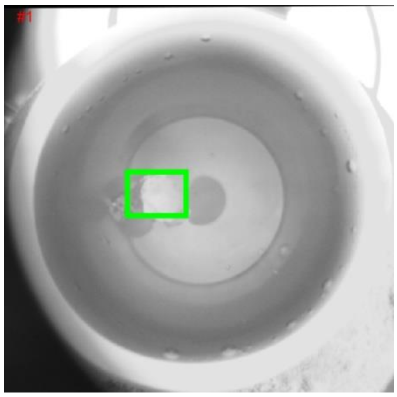

Figure 10. Initial tracking time image

$$\min_{w} \sum_i (f(x_i) - y_i)^2 + \lambda ||w||^2 \qquad (1)$$

λ is a regularization parameter that controls over fitting, and it is also used in the Support Vector Machine (SVM). The minimizer has a closed form, which is given by equation 1.

$$w = (X^T X + \lambda I)^{-1} X^T y \qquad (2)$$

Each line of data matrix X has a sample x_i, and each element of y is a recession regression target y_i, and I is an identity matrix.

The plural form of equation 2 may be used later:

$$w = (X^H X + \lambda I)^{-1} X^H y \qquad (3)$$

Where X^H is the Hermitian transpose, i.e., $X^H = (X^*)^T$, and X^* is the complex-conjugate of X. For real numbers, equation (3) reduces to equation (2).

To compute the solution, a large system of linear equations must typically be solved, which might be prohibitive in a real-time situation. In the following paragraphs, we'll look at a specific instance of x_i that gets around this restriction.

If the classifier sample does not take several actual image blocks, it consists of the target image block and the target image block shift image. If the classifier sample does not take several actual image blocks, it consists of the target image block and the target image block shift image. This situation can also form a sample set to a certain extent and training the desired trainer. The shifted example is shown in figure 11.

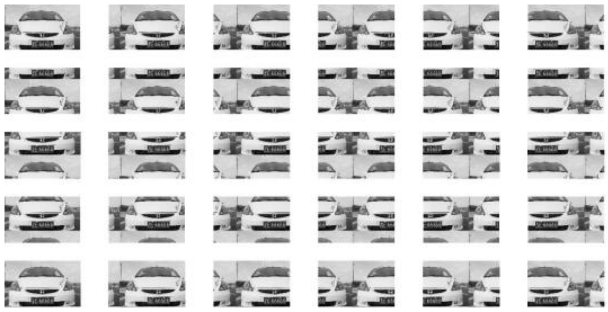

Figure 11. Example of 2D image

Figure 11 shows how the samples on both sides are the most similar to the target samples. After that, you can get the labels for each sample as follows: Set up a two-dimensional Gaussian function and shift it so that the function value with the biggest Gaussian function center corresponds to the upper-left corner sample.

When using the unique target X and the samples of other rows of the circulant matrix as the sample set of the classifier, the data matrix X is equal to the circulant matrix C, then the property of diagonalization used in equation 3 can be reduced to:

$$F^H \mathrm{w} = diag\left(\frac{\hat{\boldsymbol{x}}^*}{\hat{\boldsymbol{x}}^* \cdot \hat{\boldsymbol{x}} + \lambda}\right) F^H y \quad (4)$$

Namely,

$$\hat{\boldsymbol{w}}^* = diag\left(\frac{\hat{\boldsymbol{x}}^*}{\hat{\boldsymbol{x}}^* \cdot \hat{\boldsymbol{x}} + \lambda}\right) \hat{\boldsymbol{y}}^* \quad (5)$$

where sign with the up-script denotes the Fourier transform form of the corresponding vector, and sign with the up-script * denotes its conjugate, and point multiplication denotes the equivalent multiplication of components. The large-scale inverse operation is reduced into element point multiplication, resulting in a significant reduction in calculation time. Adding a cos window to the target sample with a high core area weight and a low surrounding area weight can improve findings, but it also weakens the influence of background information.

B. Rapid detection

When there is a subsequent frame of picture, the detection and tracking algorithm must first detect each possible image block around the region of interest, calculate its score, and then deliver the target position in this frame. To achieve similar results, the KCF algorithm does not require to detect the actual image block, but rather an actual image block and the fictional image block represented by each row of its circulant matrix. In this method, the scores of all real and artificial picture blocks can be determined using only one formula. Equation 6 is a good example.

$$\boldsymbol{y} = X\boldsymbol{w} \quad (6)$$

Simplify to

$$\hat{\boldsymbol{y}} = diag(\hat{\boldsymbol{x}})\hat{\boldsymbol{w}} \quad (7)$$

Equation 5 and equation 7 are the formulas for fast training and fast detection.

C. Kernel trick

Firstly, the training method is introduced. w^T is replaced by kernel operator$\sum \alpha_i \varphi(\boldsymbol{x_i})^T \varphi(i = 1,2,\cdots,n)$, and n is the number of samples. If the score of the sample $\boldsymbol{x}$ is

$$\sum_i \alpha_i \varphi(\boldsymbol{x_i})^T \varphi(\boldsymbol{x}) \quad (8)$$

Combining the conclusion of SVM algorithm and substituting the kernel operator into equation 1, *a* (composed vector *ai*) of equation 1 is calculated by the following formula

$$\alpha = (K + \lambda I)^{-1} y \quad (9)$$

Where, the element (I,J) in K is $\varphi(\boldsymbol{x_i})^T \varphi(\boldsymbol{x})$, and obviously K is also a circulant matrix. Equation 9 can be reduced to

$$\hat{\boldsymbol{\alpha}}^* = \frac{\hat{\boldsymbol{y}}^*}{\hat{\boldsymbol{K}}^{xx} + \lambda} \quad (10)$$

Where $\hat{\boldsymbol{K}}^{xx}$ is the generation vector of K, that is, the Fourier transform result of the first line of K. Consider the element j of $\hat{\boldsymbol{K}}^{xx'}$

$$\begin{aligned} \hat{K}_j^{\boldsymbol{xx'}} &= \varphi^T(\boldsymbol{x})\varphi(\boldsymbol{x'}) \\ &= \varphi^T(\boldsymbol{x})\varphi(P^{j-1}\boldsymbol{x'}) \end{aligned} \quad (11)$$

Taking point multiplication kernel as an example, $\hat{K}_j^{\boldsymbol{xx'}} = x(P^{j-1}\boldsymbol{x'})^T$, so

$$\begin{aligned} \hat{K}^{\boldsymbol{xx'}} &= \boldsymbol{x}C(\boldsymbol{x'})^T \\ &= \boldsymbol{x}F^T diag(\boldsymbol{x'})F^H \\ &= \hat{\boldsymbol{x}} diag(\boldsymbol{x'})F^H \end{aligned} \quad (12)$$

Namely,

$$\hat{K}^{xx'} = \hat{\boldsymbol{x}} diag(\boldsymbol{x}') = \hat{\boldsymbol{x}} \cdot \hat{\boldsymbol{x}}' \quad (13)$$

Then, the detection method is introduced, and the scores of regions of interest and fictitious image blocks can be calculated from the following equation:

$$\begin{aligned} \boldsymbol{y} = f(z) &= \left[\boldsymbol{\alpha}^{\boldsymbol{T}}\left(C(k^{xz})\right)\right]^T \\ &= \left(C(k^{xz})\right)^T \boldsymbol{\alpha} \\ &= F^H diag\left(\hat{k}^{xz}\right) F \boldsymbol{\alpha} \end{aligned} \quad (14)$$

Namely,

$$\hat{\boldsymbol{y}} = \hat{k}^{xz} \cdot \hat{\boldsymbol{\alpha}} \quad (15)$$

Where, $\hat{k}^{xz}$ can be obtained by replacing $\boldsymbol{x}'$ in equation 13 with z.

Because of the nature of kernel, equation 13 can be easily extended to the case of multi-dimensional feature input. Therefore, KCF algorithm can use 31 dimensional fHOG features for training and detection to achieve better tracking effect.

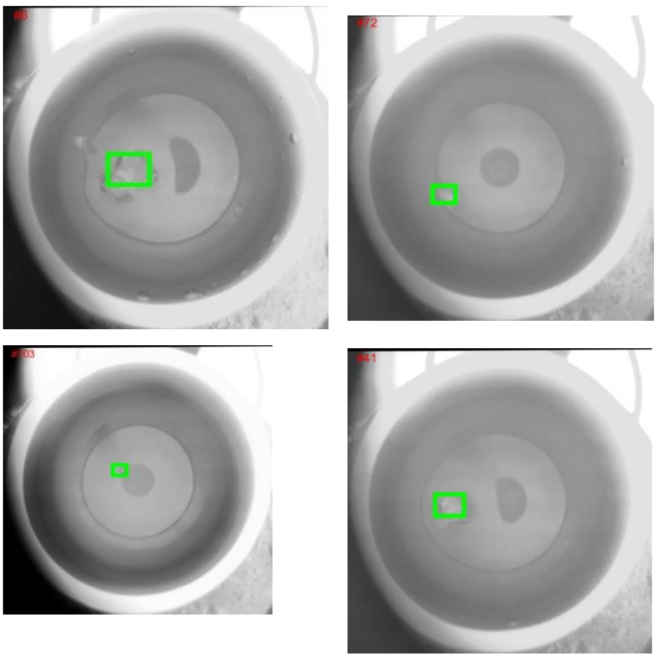

Figure 12. Tracking process image

The figure in the tracking process is as shown in figure 12.

IV. GAC CONTOUR EXTRACTION ALGROTHIM

The GAC algorithm is introduced based on color and move information novation. The GAC algorithm defines the energy function of the contour function, which describes the desired contour by reducing the energy function. The GAC expresses the contour using the Level Set function, defines the Level Set's energy function, which is paired with motion information, and then uses variational knowledge to generate the partial differential equation of the function. By lowering the energy function, the development of this function can acquire the Level Set function of the actual contour.

A. *Contour extraction using GAC algorithm based on motion information*

The posterior probability of all pixel points belonging to the target and background is computed using the two motion models of the target and background determined previously, and the posterior probability of all pixel points belonging to the target and background is calculated. The GAC algorithm is then used to extract the target contour from the material's contour.

The threshold value of the binary picture is established based on the target motion probability graph generated prior to observation, and then the region with limited space is connected using the morpho-logical filtering expansion operation. Finally, the initial contour is the convex hull of the greatest linked region. Choose an appropriate-sized region of interest, calculate the Euclidean distance between each pixel in the region and the contour, and get the initial level set function 4. Construct the energy functional about motion probability $P_0(x_i)$:

$$\begin{aligned} &E_M(C) \\ &= \mu \text{Length}(C) + \vartheta \text{Area}(C) \\ &+ \lambda_1 \iint_\Omega \left(1 - 2P_0(x)\right) dx \\ &+ \lambda_2 \iint_\Omega (2P_0(x) - 1) dx \end{aligned} \quad (16)$$

When the other terms in equation 16 remain unchanged, the shorter the contour, the smaller the energy functional; the second term represents the area of the inner region of the contour, when the other terms remain unchanged, the smaller the area of the inner region of the contour, the smaller the energy functional. These two terms make the contour's stable solution as short as possible and the area as small as possible, resulting in a smoother and more accurate contour.

The inner region of the contour is the integral region of the third term in equation 16. The energy functional is smaller when the contour has more pixel points with a probability of target motion greater than 0.5. The region outside the contour is the fourth integral region. The energy functional is smaller when the contour has more pixel locations with a probability of target motion less than 0.5.

When the energy functional approaches zero, the contour seeks to include pixel locations with a probability of target motion more than 0.5 while rejecting pixel points with a chance of target motion less than 0.5. At the same time, the contour is smooth and, to some extent, similar to the actual contour. With reference to the study [21], the Heaviside function was introduced to make the Level Set function calculation easier.

$$H(z) = \begin{cases} 1, Z \geq 0 \\ 0, Z < 0 \end{cases} \quad (17)$$

Equation 16 can be rewritten as a functional of $\varphi(x)$), i.e.

$$\begin{aligned} E_M(\phi) = \mu \iint_\Omega |\nabla H(\phi)| dx \\ + v \iint_\Omega H(-\phi) dx \\ + \lambda_1 \iint_\Omega (1 - 2P_0(x)) dx \\ + \lambda_2 \iint_{\overline{\Omega}} (2P_0(x) - 1) dx \end{aligned} \quad (18)$$

$$\nabla H(\phi) = \delta(\phi)\nabla\phi, \delta(z) = \frac{dH(z)}{dz}$$

Using variational method, the gradient descending flow of the above equation is obtained:

$$\begin{aligned} \frac{\partial \phi}{\partial t} = -\frac{\partial E}{\partial \phi} = \delta(\phi)\left[\mu \mathrm{div}\left(\frac{\nabla}{|\nabla \phi|}\right)\right] \\ + v + (\lambda_1 + \lambda_2)(1 - 2P_0) \end{aligned} \quad (19)$$

The partial differential equation is formed with the initial conditions $\varphi(x, 0) = \varphi_0(x)$ and the stable solution is determined iteratively, which is the Level Set function matching to the real contour. Because Equation 19 is a partial differential equation that cannot be solved directly, the H and δ function must be approximated as:

$$\begin{aligned} H_\varepsilon(z) \\ = \frac{1}{2}\left(1 + \frac{2}{\pi} arctan \frac{z}{\varepsilon}\right) \\ \delta_\varepsilon = \frac{1}{\pi} \frac{\varepsilon}{\varepsilon^2 + z^2} \end{aligned} \quad (20)$$

ε is used to control the eff ective width around the contour can vary dramatically. Figure 13 is the graph of the two functions (when $\varepsilon = 5$).

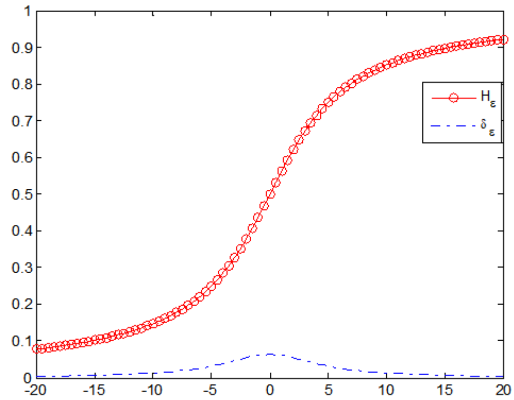

Figure 13. Regularized H and δ functions

The effect of GAC algorithm based on motion information has shown in figure 14. From the red line to the green area represents the whole process of contraction. Figure 15 shows the precise contour of silicon dioxide.

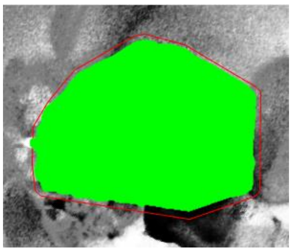
Figure 14. The effect of GAC based on motion information

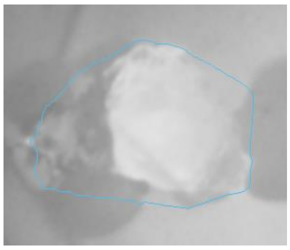
Figure. 15 Precise contour

B. *Contour extraction using GAC algorithm based on color information*

Using the contour and feature information of the initial frame, the target model is obtained, and GAC algorithm was used to evolve the contour to obtain the target contour of each frame. Since the domain of the Level Set function needs to contain the actual contour, the tracking framework is designed as follow.

- **Initialization.** According to the initial frame and the known contour, Set the appropriate region of interest initialize the Level Set function, and get the initial target model.
- **Evolution.** When the following frames are obtained, the GAC contour evolves, the stable solution obtained from evolution is transferred to the initial contour, and the region of interest is updated with the contour centroid as the center.
- **Update the target model**. According to the change of contour features in the target contour, it is decided whether to update the target model.
- **Recycle.** Recycle steps 2 and 3 until no subsequent frame trace ends or the contour is too small and the

trace fails.

- **Initialization.** The initialization of the Level Set function, resulting in the initial Level Set function, i.e. $\varphi_0(\boldsymbol{x})$. The target model is the color histogram of the region within the target contour.

$$q(z)=\frac{\iint_{\Omega_0} w\big(z-I_0(x)\big)H(-\phi_0)dx}{\iint_{\Omega_0} H(-\phi_0)dx} \tag{21}$$

$$w\big(z-I_0(\mathrm{x})\big)=\begin{cases}1, z-I_0(x)\ larger\ than\ 0\ and\ less\ than\ 8\\0, The\ other\ situation.\end{cases}$$

To obtain the color histogram of RGB, The value of Z can be choose as:

$$Z=\{\mathrm{z}=[z_1,z_2,z_3]|z_i=\{8j|\ j=1,2,\dots,16\},i=1,2,3\} \tag{22}$$

where I_0 is the initial image of RGB channel.

- **Evolution.** The energy functional for the color histogram must be defined in this phase. On the similarity measurement of the two histograms, there is a special Bhattacharyya coefficient [22] that is proportional to the size of the similarity. To begin, determine the color histogram within the current contour:

$$q_\Omega(z;\phi)=\frac{\iint_\Omega w\big(z-I(x)\big)H(-\phi)dx}{\iint_\Omega H(-\phi)dx} \tag{23}$$

Color histogram outside the current contour:

$$q_{\overline{\Omega}}(z;\phi)=\frac{\iint_{\overline{\Omega}} w\big(z-I(x)\big)H(-\phi)dx}{\iint_{\overline{\Omega}} H(-\phi)dx} \tag{24}$$

Using the Bhattacharyya coefficient metric, the similarity between the region in the contour and the target model can be defined as:

$$B_\Omega(\phi)=\Sigma_{z\in Z}\sqrt{q_\Omega(z;\phi\phi)q(z)} \tag{25}$$

The similarity between the region outside the contour and the target model is defined as:

$$B_{\overline{\Omega}}(\phi)=\Sigma_{z\in Z}\sqrt{q_{\overline{\Omega}}(z;\phi\phi)q(z)} \tag{26}$$

The actual contour should make the inner region of the contour most like the target model, while the outer region of the contour most different from the target model, then the energy functional of GAC is defend as:

$$E_H(\phi)=\varsigma B_{\overline{\Omega}}(\phi)-(1-\varsigma)B_\Omega(\phi) \tag{27}$$

$$-\frac{\partial B_{\overline{\Omega}}(\phi)}{\partial\phi}=\frac{\delta(\phi)}{2A_{\overline{\Omega}}(\phi)}\left[\frac{q^{\frac{1}{2}}(Z^*)}{p_{\overline{\Omega}}^{\frac{1}{2}}(z^*;\phi)}-B_{\overline{\Omega}}(\phi)\right] \tag{28}$$

In summary, the gradient descent flow of E_H can be obtained as follows:

$$-\frac{\partial E_H}{\partial\phi}=$$

$$-\frac{1-\varsigma}{2A_\Omega(\phi)}\left[\frac{q^{\frac{1}{2}}(Z^*)}{p_{\overline{\Omega}}^{\frac{1}{2}}(z^*;\phi)}-B_{\overline{\Omega}}(\phi)\right]\delta_\varepsilon(\phi) \tag{29}$$

$$-\frac{1-\varsigma}{2A_\Omega(\phi)}\left[\frac{q^{\frac{1}{2}}(Z^*)}{p_{\overline{\Omega}}^{\frac{1}{2}}(z^*;\phi)}-B_\Omega(\phi)\right]\delta_\varepsilon(\phi)$$

Set up diff erential equations and solve them iteratively:

$$\begin{cases}\frac{\partial\phi}{\partial t}=\frac{\partial E_H}{\partial\phi}\\\phi(0)=\phi_0(x)\end{cases} \tag{30}$$

- **Update model.** When the similarity between the region in the contour and the target model is lower than the set threshold, the target model is updated.

V. Conclusion

This paper's data originates from the Ninth Asian Pacific College Students' mathematical modeling competition.

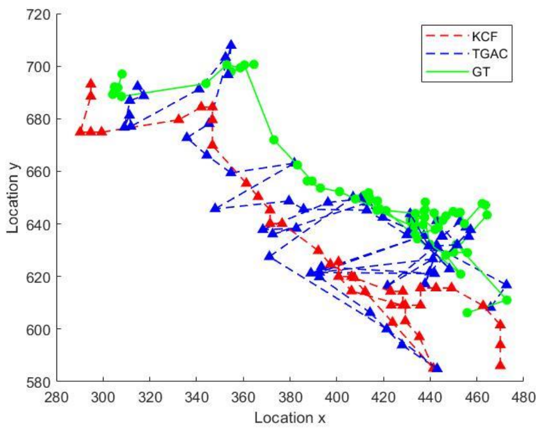

Figure 16. Motion trace image

The tracking track is obtained using the KCF method, and contour tracking is performed using the GAC algorithm based on color information. The outcomes are depicted in figure 16. (TGAC is a geometric active contour tracking system. GT stands for ground truth.)

The algorithm is better than using KCF alone after adding contour thinning; the total error is displayed in figure 17, and the errors in the X and Y directions are shown in figure 18. The error is minimal overall, but there is a considerable error between the 15th and 35th photos due to the fuzziness of some data sets. The KCF algorithm is used in this paper. KCF is a sophisticated target tracking algorithm that can track a target even when the light changes, the video shakes, and so on. The combination of KCF and GAC improves the precision and speed of contour refining solutions.

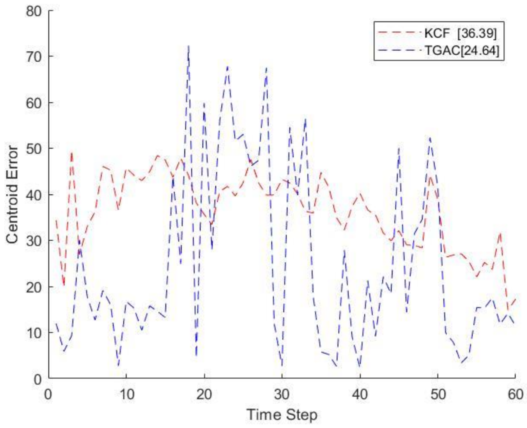

Figure 17. Error diagram

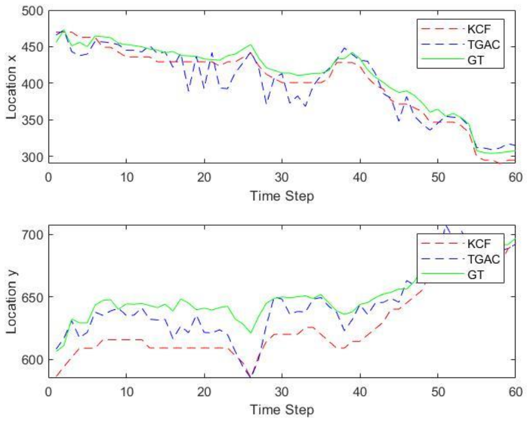

Figure 18. Error diagram in x and y directions

The paper analysis is reasonable. And the correction factors is convenient for engineering application. However, there are still shortcomings. For tracking very small targets, KCF algorithm has inherent disadvantages. For the part of color feature information in the GAC algorithm, the reference original algorithm is based on color image. And the complexity of gray distribution is lower than RGB distribution. As a result, the representability of target is insufficient.

REFERENCES

[1] Comaniciu, D.; Ramesh, V.; Meer, P. Real-Time Tracking of Non-Rigid Objects Using Mean Shift. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*; IEEE Comput. Soc: Hilton Head Island, SC, USA, 2000; Vol. 2, pp 142–149. https://doi.org/10.1109/CVPR.2000.854761.

[2] Comaniciu, D.; Meer, P. Mean Shift: A Robust Approach toward Feature Space Analysis. *IEEE Trans. Pattern Anal. Machine Intell.* **2002**, *24* (5), 603–619. https://doi.org/10.1109/34.1000236.

[3] Huttenlocher, D. P.; Klanderman, G. A.; Rucklidge, W. J. Comparing Images Using the Hausdorff Distance. *IEEE Trans. Pattern Anal. Machine Intell.* **1993**, *15* (9), 850–863. https://doi.org/10.1109/34.232073.

[4] Sato, K.; Aggarwal, J. K. Temporal Spatio-Velocity Transform and Its Application to Tracking and Interaction. *Computer Vision and Image Understanding* **2004**, *96* (2), 100–128. https://doi.org/10.1016/j.cviu.2004.02.003.

[5] Terzopoulos, D.; Szeliski, R. Tracking with Kalman Snakes. *undefined* **1993**.

[6] Isard, M.; Blake, A. CONDENSATION—Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision* **1998**, *29* (1), 5–28. https://doi.org/10.1023/A:1008078328650.

[7] Schimmack M. et al., “A Structural Property of the Wavelet Packet Transform Method to Localise Incoherency of a Signal”, Journal of the Franklin Institute, vol. 356, no. 16, pp. 10123-10137, 2019.

[8] Schimmack M. et al. “An on-line orthogonal wavelet denoising algorithm for high-resolution surface scans“, Journal of Franklin Institute (Elsevier Publishing), vol. 355, no. 18, pp. 9245-9270, 2018.

[9] Schimmack M. et al. “A Wavelet Packet Tree Denoising Algorithm for Images of Atomic-Force Microscopy“, Asian J. Control (Wiley and Sons publishing), vol. 20, no. 4, pp. 1367-1378, 2018.

[10] Mercorelli, P., “Biorthogonal wavelet trees in the classification of embedded signal classes for intelligent sensors using machine learning applications”, Journal of Franklin Institute (Elsevier Publishing), vol. 344, no. 6, pp. 813-829, 2007.

[11] Mercorelli, P., “Denoising and Harmonic Detection Using Nonorthogonal Wavelet Packets in Industrial Applications”, Journal of Systems Science and Complexity (Springer publishing), vol. 20, no 3, pp. 325-343, 2007.

[12] Vojir, T.; Noskova, J.; Matas, J. Robust Scale-Adaptive Mean-Shift for Tracking. *Pattern Recognition Letters* **2014**, *49*, 250–258. https://doi.org/10.1016/j.patrec.2014.03.025.

[13] Bolme, D.; Beveridge, J. R.; Draper, B. A.; Lui, Y. M. Visual Object Tracking Using Adaptive Correlation Filters. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*; IEEE: San

Francisco, CA, USA, 2010; pp 2544–2550. https://doi.org/10.1109/CVPR.2010.5539960.

[14] Henriques, J. F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37* (3), 583–596. https://doi.org/10.1109/TPAMI.2014.2345390.

[15] Danelljan, M.; Robinson, A.; Shahbaz Khan, F.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In *Computer Vision – ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2016; Vol. 9909, pp 472–488. https://doi.org/10.1007/978-3-319-46454-1_29.

[16] Danelljan, M.; Bhat, G.; Khan, F. S.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Honolulu, HI, 2017; pp 6931–6939. https://doi.org/10.1109/CVPR.2017.733.

[17] Danelljan, M.; Häger, G.; Shahbaz Khan, F.; Felsberg, M. Accurate Scale Estimation for Robust Visual Tracking. In *Proceedings of the British Machine Vision Conference 2014*; British Machine Vision Association: Nottingham, 2014; p 65.1-65.11. https://doi.org/10.5244/C.28.65.

[18] Li, Y.; Zhu, J. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. In *Computer Vision - ECCV 2014 Workshops*; Agapito, L., Bronstein, M. M., Rother, C., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2015; Vol. 8926, pp 254–265. https://doi.org/10.1007/978-3-319-16181-5_18.

[19] Jianbo Shi; Tomasi. Good Features to Track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*; IEEE Comput. Soc. Press: Seattle, WA, USA, 1994; pp 593–600. https://doi.org/10.1109/CVPR.1994.323794.

[20] Henriques, J. F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In *Computer Vision – ECCV 2012*; Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C., Eds.; Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Series Eds.; Lecture Notes in Computer Science; Springer Berlin Heidelberg: Berlin, Heidelberg, 2012; Vol. 7575, pp 702–715. https://doi.org/10.1007/978-3-642-33765-9_50.

[21] Chan, T. F.; Vese, L. A. Active Contours without Edges. *IEEE Trans. on Image Process.* **2001**, *10* (2), 266–277. https://doi.org/10.1109/83.902291.

[22] Tao Zhang; Freedman, D. Improving Performance of Distribution Tracking through Background Mismatch. *IEEE Trans. Pattern Anal. Machine Intell.* **2005**, *27* (2), 282–287. https://doi.org/10.1109/TPAMI.2005.31.

Shengbo Liu, born in 1989, received his Master degree in control theory and control engineering from Beijing Forestry University in 2014. Now he works in the internship and experiment center of the school of technology in Beijing Forestry University. He is interested in researching of computer aided automation.

Yi Guo, born in 1996, received his bachelor's degree in engineering from Shandong University of Science and Technology in 2018. Since then, She has been studying for a master's degree in Beijing Forestry University, majoring in control engineering.

Yandong Zhao, born in 1965, is a professor at Beijing Forestry University. She received her Doctor's degree in China Agricultural University. She is interested in intelligent detection and information processing.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Shengbo Liu carried out the idea of the tracking procedure and built the infrastructure.

Yi Guo made the experiments and finished the analysis.

Yandong Zhao was responsible for the article.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

Special Fund for Beijing Common Construction Project