Enhance Tracker's Classifier via View Morphing

Xiang Li
Institute of Image Processing and Pattern Recognition
Shanghai Jiao Tong University
Shanghai, China
lostxine@sjtu.edu.cn

Yue Zhou
Institute of Image Processing and Pattern Recognition
Shanghai Jiao Tong University
Shanghai, China
zhouyue@sjtu.edu.cn

***Abstract*—Under the situation of model-free tracking, classifier in tracking system often suffers from limited input samples and bounded observation points. System would perform better, provided as much information of the target as possible. However, it would be extremely hard to select a perfect feature extractor in all cases. In this paper, we tried a novel way to dig out more information from existent pictures, without changing the feature extractor. A brand new module, called View Morphing, was installed on tracking system. It can use several basic target images to generate lots of virtual samples, even the camera parameters are unavailable. These morphed pictures described the target at various observation points, contained naive 3D constrains, and would help the classifier understand the target better. Experiments showed that view morphing module enhanced both classical tracker and deep-learning one. We believe that our module provided a possible solution on lack of training images.**

Keywords-view morphing; discriminative tracking; epipolar geometry; deep learning

I. INTRODUCTION

There are various of applications depending on the visual tracking system, such as human-computer interaction and video surveillance. Most of trackers can be categorized into generative one, which maintains a model of target and searches for the best proposal, and discriminative type, that holds a classifier to distinguish target and background. Recent benchmarks[1][2] witness the advantages of later one, especially when background is complicated..

According to the brilliant work of Wang *et al.*[3], a typical discriminative tracking system can be divided into five parts: motion model, feature extractor, observation model, model updater and ensemble post-processor. Wang also concluded that feature extractor played the most important role. However, it would be hard to create a brand new feature carrying more valuable information or just keeping available in various cases, features from deep-learning suffered from lack of samples, especially. At the same time, the classifier was constrained frequently by limited input samples and bounded observation points.

Previous work tried to overcome this problem by sampling 2D patch[4][5], 2D point cloud[6][7] or constructing sparse 3D model containing point and line[8], eventually dense 3D model[9]. All of these methods tried to design a framework that could cover as much information as possible. Although there is a risk of convergence failure and the process of modeling is time-consuming, experiments above showed that much more improvement came up with 3D modeling, especially when out-of-plane rotation changed the target appearance[8]. However, at most situation of monocular tracking, neither camera parameter nor scene information is available. That means, it will be difficult to calculate the exact Euclidean reconstruction precisely in a short time. Fortunately, as for real-time tracking problem, there is no need to acquire accurate target model. What really interests us is actually the relevant position of target, background and occlusion. Considered these chances and challenges, in this paper, we managed to recover relevant depth, avoiding using prior knowledge or reconstruction method. A brand new module, called View Morphing, was applied in visual tracking system, to discover multi-view information of the object for a given feature. The core of module came from Image-based rendering(IBR), of which [10] gives an excellent introduction.

Here comes the outline of this paper. Background, motivation and main idea are proposed first in this section. Then the implementation of view morphing is briefly reviewed. Section III covers the details of two typical enhanced tracking system. Finally, experiments are presented in section IV, followed by analysis and conclusion.

II. VIEW MORPHING

View morphing, an image-based rendering method, is used to generate views between two or more given images, which is first proposed by Steve Seitz[11]. Following works [12][13] provided several different approaches of view morphing. Xiao and Shah[14] expanded the number of views and available areas and brought much more freedom to the image morphing system. All of above achieved acceptable results and proved the feasibility of view morphing algorithm.

In this paper, the naivest view morphing method called three step algorithm is applied, including prewarping, image morphing and post warping. Relevant images in view morphing are shown in Figure 1. Suppose two original frames are I and I', there is a 3D point called $X = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ in space. The projections of X are

$x \in I$ and $x' \in I'$. Optical centers are C and C'. To simplified the calculation, set the origin at C and consider $\overline{CC'}$ as X-axis. We define the orientation of $\hat{I}$, $\hat{I}'$ and $\hat{I}_s$ as $I_{3\times3}$. Detail implementation of these three steps are given bellow.

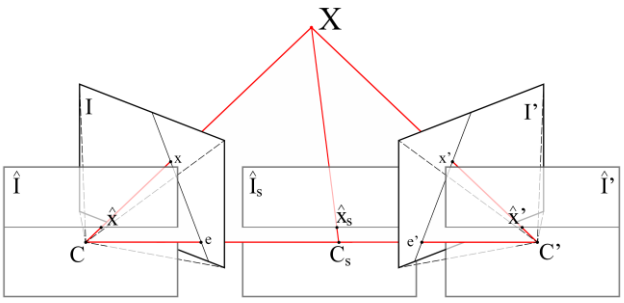


Figure 1. Relevant images in view morphing

A. *Prewarping*

To lower the time complexity of image matching and morphing, input images need to be transformed into same orientation. Prewarping is designed to produce aligned images by rotating but not shifting the cameras. Richard[15] proved that only a projective transformation performed during the warping.

Define $P = \left[R \middle| -RC\right]$ as the projection matrix of image I, similarly, $\hat{P} = \left[\hat{R} \middle| -\hat{R}C\right]$ is the projection matrix of image $\hat{I}$. Therefore, the projections of X can be linked by following transformation:

$$\hat{x} = PX = \hat{R}R^{-1} \cdot Px = R^{-1}x \tag{1}$$

If we use the 3×3 matrix $\hat{R}R^{-1}$ as a projective transformation, the orientation of camera can be updated without shifting optical center. After calculating the fundamental matrix between I and I', a common direction can be concluded. Two projective transformations above (for both cameras) can turn these images into parallel views $\hat{I}$ and $\hat{I}'$, whose common orientation is defined as $I_{3\times3}$.

B. *Image Morphing*

Aligned images provide great convenience for dense image matching, which is the key of image morphing. Suppose number $s \in [0,1]$ (default: $s = 0.5$) describe the position of virtual optical center C_s, through following equation:

$$C_s = (1-s)C + sC' \tag{2}$$

Combined Eq.2 with the definition $P = \left[R \middle| -RC\right]$, we could conclude the relationships between projection matrices of parallel views:

$$\hat{P}_s = (1-s)\hat{P} + s\hat{P}' = \left[I_{3\times3} \middle| -C_s\right] \tag{3}$$

Therefore, the projection $\hat{x}_s$ can be written into:

$$\hat{x}_s = \frac{1}{z}\hat{P}_s X = (1-s)\frac{1}{z}\hat{P}X + s\frac{1}{z}\hat{P}'X \tag{4}$$

Noticed that $\frac{1}{z}$ was introduced to make homogeneous coordinates of $\hat{x}_s$ looks like $\hat{x}_s = \left[u_s \; v_s \; 1\right]^T$. u_s and v_s are exactly the coordinates of $\hat{x}_s$ in plane $\hat{I}_s$.

Finally, the formulation can be simplified as:

$$\hat{x}_s = (1-s)\hat{x} + s\hat{x}' \tag{5}$$

Based on above equations, several feature points could be located at the morphed image. As for other plenty of 'normal' pixels, dense stereo matching algorithm is performed to find their correspondences and generate a disparity map. Changes of s make the virtual camera move along $\overline{CC'}$.

C. *Postwarping*

After morphing, the orientation of virtual camera is still same as any other prewarped images. While in realistic world, it should rotate from I to I' smoothly. Postwarping is applied to bring the camera back.

Expand the projection matrix $\hat{P}_s = \left[\hat{R}_s \middle| -\hat{R}_s C_s\right]$, with fixed C_s in Eq.2. As we supposed above, the common orientation of parallel views was $\hat{R}_s = \hat{R} = \hat{R}' = I_{3\times3}$. With the help of Eq.1, the projective transformation of postwarping is:

$$R_s I^{-1} = R_s = (1-s)R + (s)R' \tag{6}$$

Conclusively, view morphing avoids to reconstruct accurate 3D model of the scene and uses only correspondence to generate virtual images between views. Since these pictures were 'token' from different places, they could bring more information to the classifier without changing feature extractor.

III. Enhanced Tracking System

To verify the improvement of view morphing, we modified the general pipeline of visual tracking system (Figure 2.) and implemented two independent systems. One called VM-Diag is a classical tracker based on [3], the other named VM-MDNet comes from state-of-the-art deep learning tracker, MDNet [16]. Detail implementation of

original modules in both systems, except view morphing, are listed in TABLE I.

Our view morphing module in tracking system is based on min-eigenvalue feature and semi-global matching. Linear interpolation is included to fix holes in disparity map during dense matching. Considered the different motion between object and background, only the points around target were involved with fundamental matrix calculation.

For every new frame, motion model first gives several predictions based on target positions in last frame. Feature extractor tries to describe these predicted areas then. Observation model, actually a classifier, evaluates each predictions and give scores representing similarity to the target. Final prediction usually comes from the pooling result of several high-score predictions.

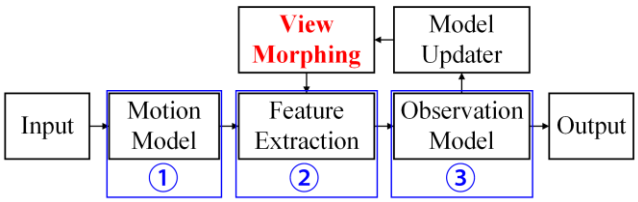


Figure 2. General pipeline of enhanced visual tracking system

TABLE I. MODULES IN TWO ENHANCED TRACKING SYSTEM

Module	VM-Diag.	VM-MDNet
①	particle filter	Gaussian distribution
②	HoG + raw	MDNet(CNN)
③	logistic regression	softmax regression

As the classifier gives scores, model updater judges whether there is a need retraining the classifier. To avoid failure of view morphing or model drift, view morphing is performed only when both last and current frame are 'good' enough, which means they both get high scores from observation model. New morphed images will be regarded as new training samples; they will pass through feature extractor to improve the classifier.

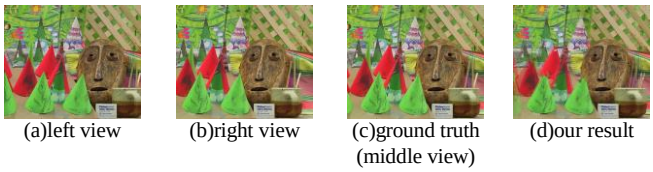
IV. EXPERIMENT

To evaluate our algorithm from theory to practice, we first tested the availability of view morphing, which will be used to enhance two trackers. Then both modified trackers were performed on OTB100[1] and compared with the their origins to exhibit improvement.

A. *Availability of View Morphing*

Stereo data from [18] were used to evaluate our view morphing algorithm. Given two related images shown in Figure 3. (a)(b). Images in (c)(d) shows the ground truth and view morphing output when $s = 0.5$

Another experiment in Figure 4. introduces the disparity map that is used to tell target from background. The difference in depth around the bounding box gives a clear boundary of target and background, which will be used to optimize feature extractor.



(a)left view (b)right view (c)ground truth (middle view) (d)our result

Figure 3. Availability of view morphing

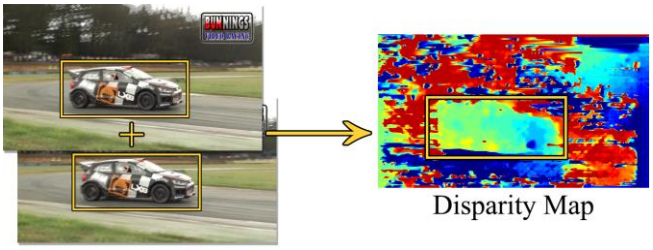


Figure 4. Disparity map from view morphing

Experiment results above proved the availability of view morphing in tracking situation, and the possibility of extracting more information from two related frames. That would provide convenience and improvement in tracking system.

B. *Performance of Enhanced Tracker*

Both modified trackers with their origins, were performed on OTB100, which contains 100 interesting sequences. Eleven challenges in tracking such as deformation, occlusion and illumination variation are associated with relevant sequences.

The benchmark offers two evaluation metrics:

(a) Precision Plot. It is a classic metric reflecting the average center location error. As the location error threshold increases, the plot shows the growing percentage of frames whose estimated location is within the current threshold. Representative threshold = 20.

(b) Success Plot. This metric evaluates the bounding box overlap. Giving the ground truth bounding box b_g and tracker output box b_t, the overlap is defined as $S = \frac{\text{pixel number}(b_g \cap b_t)}{\text{pixel number}(b_g \cup b_t)}$

The plot shows the number of successful frames whose overlap S is larger than the given threshold. Representative threshold = 0.5.

For both metrics above, higher value means better performance. Each sequence can perform 3 types of test:

(a) One-Pass Evaluation. Run through the sequence with initialization from the ground truth position in the first frame.

(b) Temporal Robustness Evaluation. Start the sequence with initialization at several different start frames, covering OPE.

(c) Spatial Robustness Evaluation. The ground truth position for initialization is combined with disturbance.

All the types of test above can output two metrics.

For classical tracker VM-Diag compared with Diag, TABLE II. shows the overall comparison.

TABLE II. COMPARISON OF CLASSICAL TRACKERS ON OTB100

Metric	VM-Diag.	Diag.[3]	Struck[17]
Secc, SRE	**0.526**	0.520	0.499
Prec, SRE	**0.633**	0.624	0.617
Secc, TRE	0.623	**0.624**	0.584
Prec, TRE	0.709	**0.710**	0.680

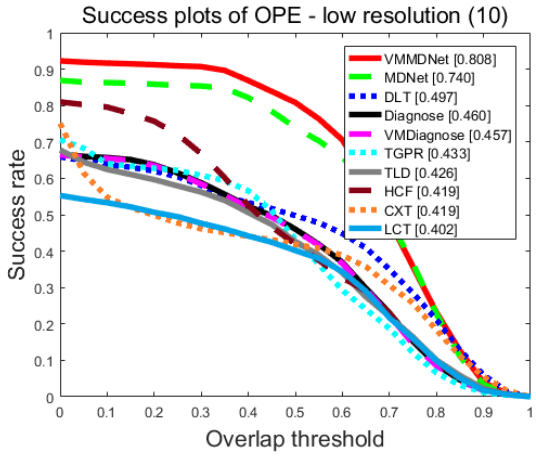
Under this situation, view morphing module in classical tracker achieved improvement on SRE and tiny loss on TRE. For further details, the SRE precision differences on each attributes were calculated and ordered in TABLE III.

TABLE III. SRE PRECISION COMPARISON OF CLASSICAL TRACKERS IN ATTRIBUTES

Attribute	VM-Diag.	Diag.	Difference
Motion blur	0.558	0.535	0.023
Out of view	0.573	0.556	0.017
Fast motion	0.561	0.545	0.016
Low-resolution	0.616	0.601	0.015
Illumination variation	0.646	0.632	0.014
Scale variation	0.621	0.609	0.012
Background clutter	0.645	0.634	0.011
Occlusion	0.611	0.603	0.008
Deformation	0.559	0.552	0.007

Obviously, for the classical tracker, view morphing module improved tracking performance significantly under motion blur, out of view, fast motion, and low-resolution situations.

For deep-learning tracker VM-MDNet compared with MDNet, the view morphing module was only involved into on-line main loop so that they shared pre-trained model. Significant improvement showed up in low-resolution condition (Figure 5.) and VM-MDNet was slightly better under other situations.



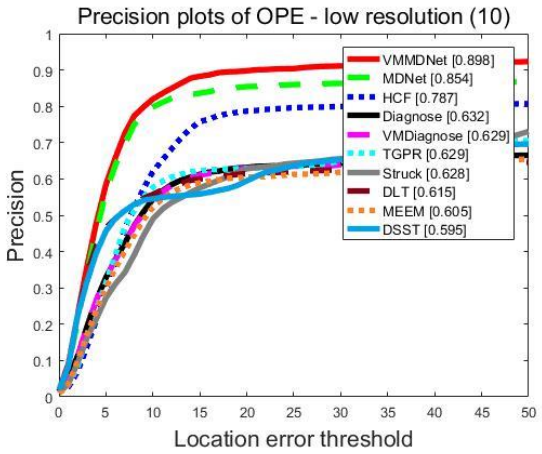


Figure 5. Low resolution performance of VM tracker on OTB100

After series of experiments on both classical one and deep-learning one, the characteristics of view morphing has been understood better. Two common main reason causing improvements above were concluded:

(a) View morphing selected proper sample and time to update model. When the module is activated, both input images has got high score from the classifier. Learning from these samples boosted the updating rate of model. Therefore, the classifier had more chance to learn high score samples of target and performed more precisely.

(b) View morphing provided plenty of samples with various quality. Due to learning multi-view images generated by morphing, classifier would perform better on perspective motion. At the same time, images blurred during warping and morphing. The improvement on blur and low resolution situation exactly proved that little blurred morphed image enhanced the robustness of classifier.

V. CONCLUSION

In this paper, we involved a new module called view morphing into two typical visual tracking systems. Experiments figured out how additional input samples improve the system accuracy and robustness especially under low resolution, motion blur and other situations. The new module can be easily implemented into any discriminative trackers.

Considered features above, we believe that view morphing will help boost performance in much more situations with limited input samples.

ACKNOWLEDGMENT

This work is supported by the National High Technology Research and Development Program of China (863 Program) under Grant 2015AA016402 and Shanghai Natural Science Foundation under Grant 14Z111050022.

REFERENCES

[1] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, “Object tracking benchmark,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 37, no. 9, pp. 1834–1848, 2015.

[2] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, Gustav Hager, Georg Nebehay, and Roman Pflugfelder, "The visual object tracking vot2015 challenge results," pp. 1–23, 2015.

[3] Naiyan Wang, Jianping Shi, Dit-Yan Yeung, and Jiaya Jia, "Understanding and diagnosing visual tracking systems," pp. 3101–3109, 2015.

[4] David A Ross, Jongwoo Lim,Ruei-Sung Lin,and MingHsuan Yang, "Incremental learning for robust visual tracking," International Journal of Computer Vision, vol. 77, no. 1-3, pp. 125–141, 2008.

[5] Sam Hare, Amir Saffari, and Philip HS Torr, "Struck: Structured output tracking with kernels," pp. 263–270, 2011.

[6] Mathias Klsch and Matthew Turk, "Fast 2d hand tracking with flocks of features and multi-cue integration," pp. 158–158, 2004.

[7] Luka Cehovin, Matej Kristan, and Ale Leonardis, "Robust visual tracking using an adaptive coupled-layer visualmodel," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 35, no. 4, pp. 941–953, 2013.

[8] Karel Lebeda, Simon Hadfield, and Richard Bowden, "2d or not 2d: Bridging the gap between tracking and structure from motion," in Computer Vision–ACCV 2014, pp. 642–658. Springer, 2014.

[9] Karel Lebeda, Simon Hadfield, and Richard Bowden, "Dense rigid reconstruction from unstructured discontinuous video," pp. 19–27, 2015.

[10] SC Chan, Heung-Yeung Shum, and King-To Ng, "Image-based rendering and synthesis," Signal Processing Magazine, IEEE, vol. 24, no. 6, pp. 22–33, 2007.

[11] Steven M Seitz and Charles R Dyer, "Toward imagebased scene representation using view morphing," vol. 1, pp. 84–89, 1996.

[12] Jin-Young Song, Yong-Ho Hwang, and Hyun-Ki Hong, "View morphing based on auto-calibration for generation of in-between views," in Computational Science and Its Applications–ICCSA 2004, pp. 799–808. Springer, 2004.

[13] Xiaoyong Sun and Eric Dubois, "View morphing and interpolation through triangulation," pp.513–521,2005.

[14] Jiangjian Xiao and Mubarak Shah, "Tri-view morphing," Computer Vision and Image Understanding, vol. 96, no. 3, pp. 345–366, 2004.

[15] Richard Hartley and Andrew Zisserman, Multiple view geometry in computer vision, Cambridge university press, 2003.

[16] Hyeonseob Nam and Bohyung Han, "Learning multidomain convolutional neural networks for visual tracking," CoRR, vol. abs/1510.07945, 2015.

[17] Sam Hare, Amir Saffari, and Philip HS Torr, "Struck: Structured output tracking with kernels," pp. 263–270, 2011.

[18] Daniel Scharstein and Richard Szeliski, "High-accuracy stereo depth maps using structured light," vol. 1, pp. I– 195, 2003.