Visual Tracking combining Region Covariance with Gaussian Process Prediction

Humberto P. Marques, Levy G. Chaves, and J. E. Bessa Maia

Ciência da Computação, UECE - Universidade Eestadual do Ceará, Fortaleza, CE.
{humberto.marques,levy.gurgel}@aluno.uece.br,jose.maia@uece.br

Abstract. This work implements a video tracking method which combines region covariance detection with motion estimation by Gaussian Process Regression. The computational and accuracy performance of the proposed method is compared to two other state-of-the-art algorithms available in OpenCV using benchmark videos from the VOT public challenges. The results show that the method is competitive and can be used in the construction of practical applications in several contexts of video surveillance.

Keywords: Visual Tracking, Region Covariance Model, Gaussian Process Prediction.

1 Introduction

Visual Object Tracking is a task in Computer Vision which is the core of applications such as scene understanding, anomaly detection, video surveillance, people counting or human-computer interaction. The goal of object tracking is to determine the position of the object in images sequences in dynamic scenes in a reliable way [1]. Visual tracking is a challenging task due to the presence, among others, of ambiguities, background clutter, object deformation, partial and full occlusions and illumination change.

The modeling and matching of the shape and appearance are critical and interactional two components of object representation, of prime importance for the success of a tracking algorithm [2]. To this end, the covariance matrix of shapes and textures [3], or features [4] appears in several contexts as a powerful model capable of retaining important properties of the represented entity. In addition, covariance matrices are mathematical entities with similarity measures based on established theories [5, 2].

On the other hand, model-based tracking requires a predictive model of object movement. To this end, Gaussian Process (GP) regression and prediction [6] has already been used successfully in trajectory prediction by capturing the trend of motion with better fidelity than linear models [7–11]. And, unlike the ordinary regression prediction whose prediction provides only one point, gaussian process prediction makes a Bayesian prediction that delivers the probability density to the predicted value. This is advantageous in contexts where, for example, it is necessary to calculate risks or propagate belief. In addition, Gaussian Process (GP) modeling is a non-parametric approach for solving regression problems which works by marginalizing distribution functions, providing an automatic tradeoff between model complexity and data fitness.

In this work a Visual Tracking method combining Region Covariance Detection with Gaussian Process Prediction is described and evaluated. The following sections of the paper are thus organized. In Section 2 the method is described in detail. In Section 3 the results of the experiments and their discussion clarifies about the strengths and weaknesses of the algorithm. The work is completed in Section 4. Mathematical notation is given in the text when each symbol appears. To the knowledge of the authors, the composition used in this implementation was not previously proposed.

2 Methods

In this section introduction, a brief description of the tracking algorithm is given in this paragraph. A detailed description is presented in the subsections.

The algorithm uses two memories of the sliding window type: a memory of covariance matrices used to update the model and a memory of the trajectory used to make the GP prediction. After an initialization phase, until the frame i in which the memories are built, the algorithm executes a loop as shown in the Algorithm 1.

In the current frame, the (i+1)-frame, a Gaussian process prediction of the position of the object is performed using the positions estimated in a group of previous frames. This position is used to define the center of the search region. To make the detection [5], at each frame, we construct a feature image of the search region. For a given object region, we compute the covariance matrix of the features as the model of the region. In the current frame, we find the region that has the minimum covariance distance from the model and assign it the estimated location by detection. Finally, the position of the object in each frame is decided by the fusion of two information: the detection by covariance and the Gaussian process prediction. As done in [4], to update the model, we keep a set of previous covariance matrices and extract an intrinsic mean using the Equation 6.

2.1 Region Covariance and Detection

Tracking by Detection based Visual Object Tracking is the composition of three strongly correlated subproblems: object representation, measure of similarity, and update of representation to account for object appearance changes. In this section we will see how each of these subproblems is tackled in the region covariance framework.

Given a observed image I with height H and width W, we can extract an image of features F, $W \times H \times d$-dimensional, from I, through a transformation [4]

$$F(x, y) = \Phi(x, y, I), \qquad (1)$$

Algorithm 1 Tracking algorithm based on Region Covariance and Gaussian Process Prediction.

```
Initialization(); */Until the processing of $Frame = i$.
repeat
    $Frame = Frame + 1$;
    Captures_the_next_frame();
    Gaussian_Process_Prediction();
    Feature_Matching();
    Fusion();
    Model_Updating();
    Memory_Updating();
    Tracking_Output();
until $Frame = N$
```

where the function Φ can be any mapping of the positions of the pixels and of the image I in properties derived from image I. In this paper, given an object image, we use pixel locations (x, y), color (RGB) values and the norm of the first derivatives of the intensities with respect to x and y. Each pixel of the image is converted to a d-dimensional feature vector, with $d = 7$,

$$F(x, y) = \left[x \; y \; R(x, y) \; G(x, y) \; B(x, y) \; \left| \frac{\partial I(x, y)}{\partial x} \right| \; \left| \frac{\partial I(x, y)}{\partial y} \right| \right]^T (,) \qquad (2)$$

where R, G, B are the RGB color values, and I is the intensity [5]. The image derivatives are calculated through the filters $[-1 \; 0 \; 1]^T$. Note that we construct the feature vector using two types of mappings: spatial attributes that are obtained from pixel coordinate values, and appearance attributes, such as color and gradient. The covariance of a region is a 7x7 matrix. Thus, for a given $M \times N$ rectangular window $R \subset F$, $\{\mathbf{f}_k = F(x, y)\}_{k=1,\ldots,n}$ is a d-dimensional feature vectors inside R, with $n = N \times M$. We represent an $M \times N$ rectangular region R with a $d \times d$ covariance matrix C_R of the feature points as

$$C_R = \frac{1}{NM} \sum_{k=1}^{NM} (\mathbf{f}_k - \mu_R)(\mathbf{f}_k - \mu_R)^T \qquad (3)$$

where μ_R is the vector of the means of the corresponding features for the points within the region R. The covariance matrix is a symmetric matrix where its diagonal entries represent the variance of each feature and the non-diagonal entries represent their respective covariances.

To perform the tracking, besides representing the region of the object, we need to find the region in the test image which is most similar to the given object. Thus, we need to compute distances between the covariance matrices corresponding to the target object window and the candidate regions. Caution must be taken because the space of covariance matrices is not a vector space. Even so, considering the fact that the covariance matrices are symmetric positive definite matrices, several distance measures have already been proposed. In this work the measure defined in [4, 5] was used:

$$\rho\left(C_i, C_j\right) = \sqrt{\sum_{k=1}^{d} ln^2 \lambda_k \left(C_i, C_j\right)} \tag{4}$$

where $\left\{\lambda_k \left(C_i, C_j\right)\right\}$ are the generalized eigenvalues of C_i and C_j, computed from

$$\lambda_k C_i x_k - C_j x_k = 0, \quad k = 1, ..., d \tag{5}$$

and x_k are the generalized eigenvectors.

To complete a tracking by detection framework, note that the image of the object in the video undergoes transformations in shape, size and appearance as it moves. Thus, it is necessary to adapt the model to these variations. For this, we compute the sample mean covariance matrix that blends all the previous T covariance matrices. In case all previously detected regions and the corresponding features are stored, an aggregated covariance matrix $\tilde{C} = \left\{\sigma^2_{u,v}\right\}$ can be obtained whose entries are given by [4]

$$\sigma^2_{u,v} = \frac{1}{NMT} \sum_{t=1}^{T} \sum_{k=1}^{NM} \left[f^t_k(u) - \mu(u)\right] \left[f^t_k(v) - \mu(v)\right] \tag{6}$$

and $f^t_k \in R_t$. The mean μ is computed over all regions $R_1, ..., R_T$. This work adopted, after tests, $T = 6$. This tracking by detection framework will be combined with the object motion prediction algorithm described in the next subsection.

2.2 Gaussian Process Prediction

We treat the task of trajectory forecasting of the object as a regression problem. Given a time series of values of the previous positions as the training set, we aim to predict the object movement, extrapolating to the subsequent position. We chose to use regression because it is computationally lighter than dynamic model whereas tracking is a task to be performed online.

A Gaussian process (GP) is uniquely characterized by multivariate random variables which follow a Gaussian distribution, where the covariance matrix is given by a kernel matrix [6]. Given examples $(x_i, y_i)_{i=1,...,N}$, where x_i is element of some space X and y_i is a real value or vector, and a new point x^* with (unknown) $y^* \in R$, the conditional density function $p(y_1, ..., y_N, y^* | x_1, ..., x_N, x^*)$ is the Gaussian [12]

$$N\left(\theta_1, ..., \theta_N, \theta^*, \begin{bmatrix} K + \tilde{\sigma}^2 I^N & \boldsymbol{k}^T \\ \boldsymbol{k} & k(x^*, x^*) \end{bmatrix}\right) \tag{7}$$

where θ_i is the prior mean for y_i, k a kernel on X, K is the matrix $(k(x_i, x_{i'}))$, $\boldsymbol{k} = (k(x^*, x_1), ..., k(x^*, x_N))$, I^N is the N-dimensional identity matrix and $\tilde{\sigma}^2$ is the variance of the input noise. Marginalisation enables the inference of y^* via its density [12]:

$$p\left(y^*|x^*, x_1, ..., x_N, y_1, ..., y_N\right) = N\left(\mu, \sigma^2\right) \text{ where} \tag{8}$$

$$\mu = \theta^* + \boldsymbol{k} \cdot (K + \tilde{\sigma}^2 \cdot I^N)^{-1} \cdot (Y - \mathit{\Theta}) \tag{9}$$

$$\sigma^2 = k(x^*, x^*) - \boldsymbol{k} \cdot (K + \tilde{\sigma}^2 \cdot I^N)^{-1} \cdot \boldsymbol{k}^T, \tag{10}$$

where $Y = (y_1, ..., y_N)^T$ and $\mathit{\Theta} = (\theta_1, ..., \theta_N)$.

To apply GP regression to time series prediction, we phrased time series prediction as follows [12]: Assume that example time series $\bar{x}^1, ..., \bar{x}^M$ are given, where $\bar{x}^j = (x_1^j, ..., x_{T_j}^j)$ with $x_t^j \in X$. Then the task is to infer the successor x_{t+1}^j from its history $(x_1^j, ..., x_t^j)$ for all j and t. Following the Markov assumption, all but the last history entry become irrelevant. This leads to the regression problem with input-output pairs $\left\{\left(x_t^j, x_{t+1}^j\right)\right\}_{t=1,...,T_{j-1}}^{j=1,...,M}$ which can be modelled by GP regression, provided real vectors x_t^j. For time series models, a natural prior is to stay where you are, that is $\theta_t^j := x_t^j$. This leads to the predictive mean $\mu = \theta^* + \tilde{k} \cdot (K + \sigma^2 \tilde{\cdot} I^N)^{-1} \cdot (Y - X)$ with $X = (x_1^1, ..., x_{T_1-1}^1, ..., x_1^M, ..., x_{T_M-1}^M)^T$ and $Y = (x_1^1, ..., x_{T_1}^1, ..., x_1^M, ..., x_{T_M}^M)^T$ and predictive variance (10).

GP can be seen as placing a probability distribution on a function space. This is known as the function-space view described in [6]. The functional form of the covariance function, or kernel, k, encodes assumptions about the smoothness and generalization properties of a GP and their specific choice depends on the application. The kernel function most widely used is the squared exponential, or Gaussian, with additive noise [13]:

$$k\left(x, x^{'}\right) = \sigma_f^2 e^{-\frac{1}{2}(x-x^{'})W(x-x^{'})^T} + \sigma_n^2 \delta \tag{11}$$

where σ_f^2 denote the signal variance, the diagonal matrix W contains the length scales of the process which reflect the relative smoothness of the process along the different input dimensions and δ is the Dirac delta function. The parameter is σ_n^2 controls the global noise of the process. The function in (11) is used in this work with $\sigma_f^2 = 1.0$, $\sigma_n^2 = 0.01$ and $W = 5.0 \cdot I^2$.

Note that in the problem that we have at hand the input space X is one-dimensional, its variable being the time frame, while the output space Y is two-dimensional, the vector process $Y(t) = (x(t), y(t))$, which are the positions recording the trajectory of the object. The processes x and y are considered here as uncorrelated for simplicity. Predictions are made separately. A more realistic model, considering x and y as correlated processes is left for future implementation.

In its final stage the tracking algorithm has two information: the prediction of the position by GP and the detection based on region covariance. The tracking decision is obtained by merging these two information. In this work we have adopted a simple form of fusion that consists of calculating the arithmetic mean between the position returned by the tracking by detection with the mean of the Gaussian prediction given by Equation 9. There are a large number of proposals for probabilistic information fusion [14]. We leave to a later work to test methods of merging information in the context of the proposal of this article.

3 Results and Discussion

3.1 Evaluation and Data Set

For the tests we used 4 videos of the VOT2013 challenge [15], nominally **cup**, **jump**, **juice** and **singer**, 1 video of the VOT2014 challenge [16], **ball**, and 1 video proprietary, denominated **mandog**, totaling 2196 frames. Ground truth for the VOT sequences are available on the homepage and for the proprietary video was done manually by the authors. These sequences contain several challenging situations, for example, occlusion, confusing background, variation of illumination, motion and size changes and some maneuvers. Table 1 shows the video proprieties of the used sequences. Note from this table the various combinations of challenges that are presented to the tracking algorithm.

Following [16], we computed two performance metrics. The detection rate is the ratio of the number of frames in which the object location is accurately estimated to the total number of frames in the sequence. We consider the estimated location accurate if the best match is within the 11x11 neighborhood of the ground truth center object location. Note that detection rate is a measure of accuracy being more demanding than the accuracy defined in [4].

The second metric was the robustness or fails rate, as defined in [16]. The robustness measures how many times the tracker loses the target (fails) during tracking. A failure is indicated when the overlap measure becomes zero.

The research environment is implemented using C ++ and the OpenCV library on a personal computer with CPU Core I7, Clock = 2.4 GHz and MM = 4 GB.

	motion change	size change	occlusion	ilum. change	clutter backgroung	camera motion	num. of Frames
ball	+	+	-	-	-	+	602
cup	+	-	-	-	+	-	303
jogging	+	-	+	-	+	+	307
jump	+	+	-	-	-	+	228
singer	+	+	-	+	+	+	351
mandog	-	-	-	+	+	-	305
Total							2196

Table 1. Video properties: + (-) indicates when the property is present (absent) in the video.

3.2 Results

The tracking in each video was performed 11 times: Once with the initial position given by the ground truth and 10 times at random positions provided by a Gaussian distribution with standard deviation equal to 5 pixels around the initial position of the ground truth. Qualitative results are shown in Figure 1 and quantitative results are shown in Table 2. Only the videos in which the tracking with the initial position given by the ground truth worked were included.

Note, in Figure 1, that video ball includes a maneuver by which one could expect a predicted algorithm to fail. However, Cov + GPP did not present a failure rate lower than tracking-by-detection algorithms such as Cov. On the other hand, for the video with the greatest combination of obstacles to tracking that is the video singer, the comparison shows similar performance between TLD and Cov-GPP, both superior to the other two.

The quantitative evaluation was done by comparing the results of our implementation (Cov+GPP) with a tracking-by-detection implementation using Region Covariance Tracking (Cov), as well as two other state-of-the-art algorithms whose implementations are available in OpenCV.

Median Flow, proposed in [17], is a robust visual tracking algorithm based on the spatio-temporal constraints which tracking a single target using optical flow trajectories together with median filtering.

The Tracking-Learning-Detection (TLD) framework, firstly proposed in [18], explicitly decomposed long-term tracking task into tracking, learning and detection components. For the tracking they used forward-backward errors (Median Flow) to detect tracking failures automatically [17] which is based on Lucas Kanades feature tracker [19]. For the learning component they proposed a iterative procedure, the so-called P-N learning framework which consists of Positive-expert and Negative-expert. For the detection they used cascaded nearest neighbor classifier in order to speed up.

	MedianFlow		TLD		Cov		Cov+GPP	
	accuracy	robustness	accuracy	robustness	accuracy	robustness	accuracy	robustness
ball	.914	.044	.926	.061	.919	.066	**.928**	.074
cup	.926	.062	.934	.051	.912	.055	**.941**	.055
jogging	.882	.116	**.924**	.046	.918	.062	.922	.054
jump	.964	.031	.965	.028	.964	.022	**.983**	.018
singer	.913	.066	**.932**	.046	.915	.084	**.932**	.061
mandog	.972	.026	**.985**	.010	.962	.024	.984	.012

Table 2. Average accuracy and robustness of the tracking implementation.

Table 2 gives us two results for the tests performed:

1. that the combination of Gaussian process prediction with region covariance substantially improves the performance of the region covariance tracking;

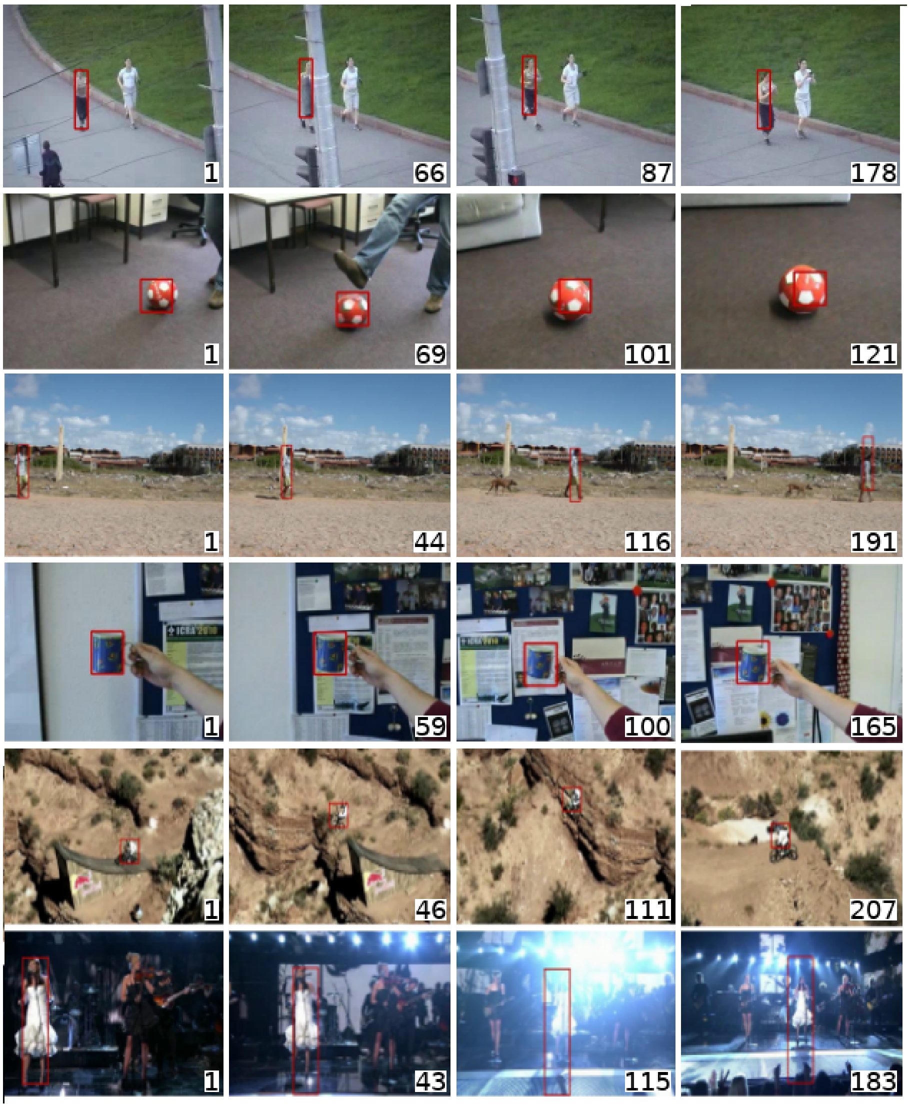

Fig. 1. Qualitative results of the tracking from top to bottom: Jogging (1, 66, 87, 178), Ball (1, 69, 100, 120), ManDog (1, 44, 116, 191), Cup (1, 59, 100, 165), Jump (1, 47, 111, 207), Singer (1, 43, 115, 183).

2. And that Cov+GPP is competitive both in accuracy and robustness. Note that although favorable results prevail for the TLD algorithm, in some cases the performance of Cov+GPP in robustness is better than that of TLD even when TLD is higher in accuracy.

These results can not be considered conclusive. A large-scale evaluation needs to be carried out for a firm conclusion. However, they show a promising combination.

4 Conclusion

A combination of region covariance detection with Gaussian process prediction was implemented for visual tracking. The concern was only with the accuracy of the tracking, no effort was made to reduce the computational cost in this implementation. This will be one of the focuses in the continuation of this work.

The videos used in the tests present several necessary properties in the evaluation of visual tracking, such as occlusion, abrupt change of lighting and abrupt change to confused background. The combination tested proved to be promising in the initial tests performed. In the next step of this work we will submit the algorithm to large scale systematic tests comparing its performance with that of state of the art algorithms.

References

[1] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, IEEE Trans. Syst. Man Cyber.-C, 34(3), pp. 334-352, (2004).

[2] J. -M. Odobez, D. Gatica-Perez, D. Ba, O. Sileye, Embedding motion in model-based stochastic tracking, IEEE Transactions on Image Processing, 15(11), pp. 3514-3530, (2006).

[3] T. F. Cootes, G. J. Edwards, C. J. Taylor, Active appearance models, IEEE Transactions on pattern analysis and machine intelligence, 23(6), pp. 681-685, (2001).

[4] Oncel Tuzel, Fatih Porikli, Peter Meer, Region covariance: A fast descriptor for detection and classification, Computer Vision-ECCV 2006, pp. 589-600, (2006).

[5] Fatih Porikli, Oncel Tuzel, Peter Meer, Covariance tracking using model update based on lie algebra, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Vol. 1. IEEE, (2006).

[6] C. E. Rasmussen, C. Williams, Gaussian processes for machine learning, MIT press, Cambridge, (2006).

[7] Tim D. Barfoot, Chi Hay Tong, Simo Sarkka, Batch Continuous-Time Trajectory Estimation as Exactly Sparse Gaussian Process Regression, Robotics: Science and Systems, (2014).

[8] Hongwei Li, Yi Wu, Hanqing Lu, Visual tracking using particle filters with gaussian process regression, Pacific-Rim Symposium on Image and Video Technology. Springer, Berlin, Heidelberg, (2009).

[9] Yao Sui, Li Zhang, Visual tracking via locally structured Gaussian process regression, IEEE Signal Processing Letters 22.9, pp. 1331-1335, (2015).

[10] Kihwan Kim, Dongryeol Lee, Irfan Essa, Gaussian process regression flow for analysis of motion trajectories, Computer vision (ICCV), 2011 IEEE international conference on. IEEE, (2011).

[11] M. Tiger, Fredrik Heintz, Online sparse Gaussian process regression for trajectory modeling, Information Fusion (Fusion), 18th International Conference on. IEEE, pp. 782-791, (2015).

[12] B. Paassen, C. Gopfert, B. Hammer, Gaussian process prediction for time series of structured data, In: Proceedings of the ESANN, 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, (2016).

[13] J. Ko, D. Fox, GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models, Autonomous Robots, v. 27, n. 1, p. 75-90, (2009).

[14] J. Kang, I. Cohen, G. Medioni, Continuous tracking within and across camera streams, In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on. IEEE, (2003).

[15] M. Kristan, et al., The vot2013 challenge: overview and additional results, (2014).

[16] M. Kristan, et al., The visual object tracking vot2014 challenge results, (2014).

[17] Z. Kalal, K. MIKOLAJCZYK, J. MATAS, Tracking-learning-detection, IEEE transactions on pattern analysis and machine intelligence, v. 34, n. 7, pp. 1409-1422, (2012).

[18] Z. Kalal, K. MIKOLAJCZYK, J. MATAS, Forward-backward error: Automatic detection of tracking failures, In: Pattern recognition (ICPR), 2010 20th international conference on. IEEE, pp. 2756-2759, (2010).

[19] B. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, IJCAI, 81:674679, (1981).