2010 International Symposium on Nonlinear Theory and its Applications
NOLTA2010, Krakow, Poland, September 5-8, 2010

Hierarchical Feature Extraction for Dynamic Feature and Signature Tracking

Vilmos Szabo† and Csaba Rekeczky‡

†Dept. of Information Technology, Pazmany Peter Catholic University
Prater 50/A, Budapest, Hungary
‡Eutecus Inc.
Berkeley, California, USA
Email: szavi@digitus.itk.ppke.hu, rcsaba@eutecus.com

Abstract— The goal of this paper is to introduce an improved tracking framework, which exploits dynamic feature and signature selection techniques for data association models. It performs robust multiple object tracking in a noisy, cluttered environment with closely spaced targets. This method extends the back-end processing capabilities of tracking systems by creating a two-level hierarchy between the parallelly extracted features. These features are dynamically selected based on a spatio-temporal consistency weight function, which maximizes the robustness of data association, and reduces the overall complexity of the algorithm.

1. Introduction

Multiple object or target tracking is an important task in computer vision applications. However, it can become a challenging problem, especially if the object is in a dynamically changing environment. A number of computer vision applications could be characterized by two complex stages of processing. The first stage is the topographic image acquisition, which may include pre-processing, image segmentation, and post-processing. The second stage is a non-topographic sensing which includes feature-signature extraction, data assignment, and state-prediction. High resolution spatio-temporal detection can be accomplished using topographic or cellular processing hardware, such as the Cellular Neural Network (CNN) [1]. The multiple object tracking back-end is usually accomplished using serial Digital Signal Processors (DSP). Therefore, the numerical complexity of the tracking algorithm is crucial in order to meet the systems real-time demand. This paper focuses on object tracking using dynamic data association and its spatio-temporal signature analysis. Application areas may include traffic monitoring, vehicle navigation, automated surveillance and biological applications.

2. Dynamic Multiple Target Tracking Framework

Multiple target tracking can be defined as estimating the trajectory of objects in the image plane as they move around in the scene [2]. Generally, an object segmentation algorithm runs on each frame of the video flow in order to detect objects. This can be done on a CNN-like massively parallel topographic hardware to achieve high spatio-temporal resolution video flow processing. The detected objects are then assigned to consistent labels, called *tracks* [3]. The temporal analysis of tracks can be used to identify and select features that best represent each object. The final goal of target tracking is to determine the position of an object or a bounding box on each frame of the video sequence. Our algorithm follows a bottom-up approach:

1. Pre-processing of input video flow
2. Parallel image segmentation algorithm
3. Post-processing of segmented video frame
4. Image labeling
5. Object shape and appearance representation
6. Parallel image feature extraction
7. Image feature normalization and selection
8. Assignment of object to tracks based on dynamic feature selection
9. Feature signature analysis

Steps 1–3 can be implemented on CNN-type hardware. Pre-processing of each video frame is an important step to eliminate unwanted noise, and to condition the signal for further analysis. Throughout the evaluation, Gaussian filtering was employed that can be approximated on the CNNs resistive grid. The time or scale parameter depends on the amount of noise in the scene. The range of pixel intensity values was converted to $I \in \{-1;1\}^{NM}$ where N and M are the width and height of the image. In case of color processing, each chromatic channel is processed separately (see subsection 2.3 Hierarchical Feature Extraction).

For post-processing, basic mathematical binary morphological [4] operators were used. The aim was to connect fragmented objects with the closing operation, and to clear individual pixels created by the "non-perfect" segmentation algorithm.

Steps 4–9 are typical serial DSP-like processing. Each of the connected components is labeled on the binary image. A number of features are extracted from the connected components for the dynamic tracking. The results of feature signature analysis (8) provide a feedback to the

dynamic feature analysis (6) in order to calculate the track consistency metric (see subsection 2.4 for details).

2.1. Motivation

The motivation for employing dynamic feature selection for multiple object tracking emanates from the need to reduce the complexity of data association steps of the the overall algorithm. Let $\mathbf{x}$ and $\mathbf{y}$ be d-dimensional vectors, where each component corresponds to a feature value. The two most widely used distance metrics are the $\mathcal{L}_1$ city block (eq. 1) and $\mathcal{L}_2$ Euclidean (eq. 2) metrics.

$$\mathcal{L}_1 : d_1(\mathbf{x}, \mathbf{y}) = \|\mathbf{x}, \mathbf{y}\|_1 = \sum_{n=1}^{d} |x(n) - y(n)| \qquad (1)$$

$$\mathcal{L}_2 : \ d_2(\mathbf{x}, \mathbf{y}) = \|\mathbf{x}, \mathbf{y}\|_2 = \sqrt{\sum_{n=1}^{d} (x(n) - y(n))^2} \qquad (2)$$

The best feature will provide the maximum interclass distance between objects. Increasing the feature space dimensionality will increase the discriminative power. However, noisy channels can decrease the robustness of the system. Therefore, the algorithm should try to select as few salient features as possible for data association. This decreases the number of features that need to be extracted. There are existing methods for dimensional reduction, such as *Principal Components Analysis* (PCA) [5]. These methods usually require a training set or block processing for dimension reduction. The feature selection method explained in this paper is a recursive one; it has a relatively low computational complexity and is able to successfully select a set of salient features in a changing environment.

2.2. Simulation Videos

The algorithm was evaluated on three computer generated video flows. The first video *Scene 1 (Shapes)* contains five dynamically changing objects. See Figure 1 for a demonstration on three objects. Each object is able to change its location, visibility, orientation, color, shape, noise, and inner-structure according to the following list:

- Location: [0–1]
- Visibility: [0–1]
- Orientation: [0–360°]
- Color: [red, green, cyan, blue]
- Shape: [circle, triangle, square, pentagon]
- Noise: [on, off]
- Inner structure: [dots, lines, concentric circles]

The second video flow is called *Scene 2 (Bipeds)*. This scene contains walking humans with crossing and overlapping paths; they are in partial and full occlusion, entering and exiting the scene. The third video flow

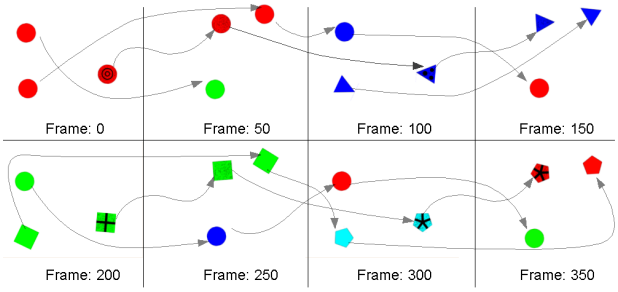

Figure 1: Demonstration of the computer generated simulation video flow containing dynamic feature transformations of the objects. The dynamic transformations include location, color, shape, noise and inner structure changes.

is called *Scene 3 (Cars)*. The first two scenes contain non-rigid objects, while the third scene contains only rigid objects. Figure 2 shows actual frames from all three video flows. The noisy version of the simulation videos had $SNR_{dB} = 15$.

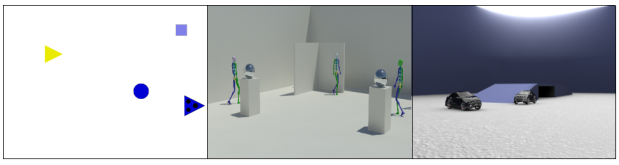

Figure 2: Computer generated video flows that were used in the algorithmic evaluation. From left to right: Scene 1 (5 dynamically changing shapes), Scene 2 (6 Bipeds), Scene 3 (4 Cars)

2.3. Hierarchical Feature Extraction

The input image is highly redundant. The transformation, to reduce the dimensionality of input data while keeping relevant information content, is called *feature extraction*. The result of the segmentation algorithm is a binary mask, where every pixel corresponds to a background or foreground pixel. A set of features were extracted over the foreground that can identify and describe each object in a given frame. These features are grouped into six statistically independent *main feature groups*. The two-level feature extraction is summarized in Table 1. In the current framework, each main feature group can be weighted separately.

The position feature group contains features associated with the location and its derivatives of each connected foreground pixels on the mask image. The second group, describes the size of each object. The shape feature group contain features that represent the shape of an object such as eccentricity (ratio of length and width of an object), extent (ratio of the area of an object to the area of the bounding box) and solidity (ratio of convex area of an

Table 1: Summary of the two level hierarchical feature extraction used for the dynamic tracking framework.

Feature Group	Subgroup
1. Position	Location
	Speed
	Acceleration
3. Scale	Area
	Major Axis Length
	Minor Axis Length
	Bounding Box
4. Shape	Eccentricity
	Solidity
	Extent (Opacity)
5. Structure	Euler Number
6. Texture	Variance
7. Color	Average Y Luminance Component
	Average Cb Color Component
	Average Cr Color Component

object to the area of the object). The texture group contains the variance feature which is extracted from the grayscale image. In order to extract the color information the image is converted to YCbCr color space, but various other color spaces can be used such as Hue, RGB, LUV, Yuv or Lab. Finally, each object is represented by a 17 dimensional feature vector (excluding speed and acceleration features because they are derived from the position). The values in the feature vector are normalized between 0 and 1 in order to make comparable measurements among each frame of the video sequence.

The choice of the feature set should be based on the requirements of a specific application area.

2.4. Dynamic Spatio-Temporal Feature Selection

In a real-time application, the number of features should be minimal to increase the speed of the system, but all relevant information must be kept. This can be done by creating a hierarchy among the features based on their confidence or robustness. The noisy feature channels should be filtered out. The *tracking system* consists of feature selection, data assignment, state space estimation, prediction and error correction.

2.4.1. Feature Selection

The feature selection is done by analyzing the spatial and temporal property of each feature channel. The "good" features are selected based on a spatio–temporal consistency metric. Let $\mathbf{x}_k^i$ and be the feature state space vector at frame k for the i^{th} object. Let $\mathbf{Q}_k^{ij}(n)$ quality matrix (eq. 3) be the minimum of pair wise l_1 (eq. 1) distance of the current state space vector n^{th} component between the i and j^{th} objects.

$$\mathbf{Q}_k(n) = min\{d_1(\mathbf{x}_k^i(n), \mathbf{x}_k^j(n)) \mid (i > j)\} \tag{3}$$

The second term of the consistency metric is the inverse of the residual gradient magnitude of the previous state space estimation. Features that are well separated from each other and do not change much in time are preferred. The final consistency metric (eq. 4) is defined by a linear μ parameter homotopy of the first part and the second parts. (The variable m donates the number of features.)

$$\mathbf{C}_k = (1-\mu)\mathbf{Q}_k + \mu \frac{1}{\frac{1}{m}\sum_{i=1}^{m} |\mathbf{x}_{k-1}^i - \mathbf{x}_k^i|} \tag{4}$$

$\mathbf{C}_k$ vector contains the quality measurement for each feature at a given time. Different feature selection strategies can be considered. A fix number of best features can be selected, or features can be selected above a given threshold level, resulting in a varying number of features for each frame. Section 3 - Performance Evaluation gives detailed comparison of the different feature selection strategies.

2.4.2. Data Assignment

The assignment of measurements to consistent tracks is accomplished using a combinatorial optimization algorithm called the *Hungarian* [6] method. Only selected features selected contribute to the calculation of the distance matrix. The assignment algorithm is used to match the current and predicted states together with minimal cost. The time complexity of the assignment algorithm is low order polynomial. More complex data association models can be applied, such as described in [7].

2.4.3. State Space Estimation

Interacting Multiple Model (IMM) was used as state estimation framework. Recursive steady-state *Kalman* filters [8] are used for the state prediction and correction phases. The prediction filters are also know as alfa, alfa-beta, alfa-beta-gamma filters. Particle filters may be used to improve the accuracy of the results [9].

3. Performance Evaluation

The evaluation of the tracking algorithm was performed on computer simulated videos. The black and white reference masks do not contain information about tracking individual objects. Therefore an object map is synthesized, where a unique color is to each object for track representation (see Figure 3). For each color value, the center of mass coordinates are extracted, which gave the reference tracks for the evaluation. Note that for Scene 1 (Shapes) the object ID mask is multiplied with the corresponding binary mask before evaluation.

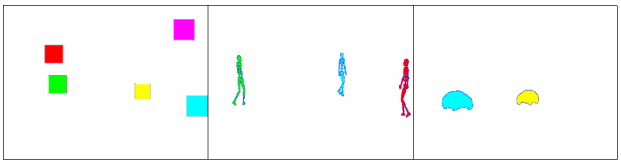

Figure 3: Object ID map flows for the simulation videos. Each object is assigned a unique color. From left to right: Scene 1 (Shapes), Scene 2 (Bipeds), Scene 3 (Cars). The images have been modified for printing.

The *mean square error* (MSE) is calculated between the reference track and measured tracks. The MSE can be calculated according to the following equation:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} d_1(Ref_x, Meas_x) + d_1(Ref_y, Meas_y) \quad (5)$$

A total of three feature selection strategies were evaluated. The first is the *Best-Feature* selection strategy, where only one feature is selected. The second strategy is when all the features (*All-Feature*) are used by the algorithm. The third is the, *K-Dynamic* selection, where a feature is selected above a given threshold level. This threshold level was set such that the quality of tracking approximately achieved the *All-Feature* selection strategy. Table 2 summarizes the MSE comparison measurements for the three feature selection strategies.

Table 2: Summary of MSE measurement between the reference and measured trajectories for the different feature selection strategies.

	Feature Selection Method		
Scene	Best-Feature	K-Dynamic	All-Features
Scene 1	5.4E-3	**8.8114E-4**	6.459E-4
Scene 2	2.9E-3	**2.6E-3**	2.5E-3
Scene 3	1.94E-4	**1.7366E-4**	1.698E-4

The *K-Dynamic* selection used an average of 2.76, 3.01, 3.41 features on average for Scenes 1-3 respectively.

Figure 4 shows the final object detection and tracking result. The images also include the bounding box and trajectory of each tracked objects.

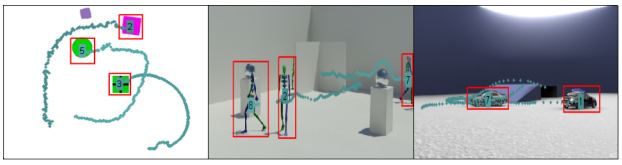

Figure 4: Object detection and tracking results on the three simulation videos. Scene 1 (Shapes) frame: 215, Scene 2 (Bipeds) frame: 250, Scene 3 (Cars) frame: 200

4. Conclusion

The tracking framework uses a dynamic feature and signature selection method for multiple target tracking. This algorithm can be used to track objects in a changing environment after topographic CNN-like segmentation and hierarchical feature extraction. The algorithm arranges the parallelly extracted features into a hierarchy, based on their consistency measurement. The overall complexity is reduced by using only the relevant features for tracking the objects in the scene, which reduces the computational time demand. Performance evaluation on synthesized videos confirmed that instead of using the 17 dimensional feature vector dynamically selecting the best 3-4 features can result in as accurate tracking. Future work will include the development of more complex selection strategies and testing the algorithms on real video sequences.

References

[1] L. O. Chua, T. Roska, T. Kozek, and A. Zarandy, "The CNN paradigm – a short tutorial," in *Cellular Neural Networks*, T. Roska and J. Vandewalle, Eds. New York: Wiley, 1995, pp. 1–14.

[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.

[3] K. Smith, D. Gatica-Perez, J. Odobez, and S. Ba, "Evaluating multi-object tracking," in *In Workshop on Empirical Evaluation Methods in Computer Vision*, 2005.

[4] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 532–550, 1987.

[5] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer, October 2002.

[6] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[7] R. Karlsson and F. Gustafsson, "Monte Carlo data association for multiple target tracking," in *IEEE Target Tracking: Algorithms and Applications*, 2001.

[8] G. Welch and G. Bishop, "An introduction to the kalman filter," Chapel Hill, NC, USA, Tech. Rep., 1995.

[9] X. Wang, S. Wang, and J. Ma, "An improved particle filter for target tracking in sensor system," *Sensors 2007*, vol. 7, pp. 144–156.