Detecting Motion Synchrony by Video Tubes *

Ying Zheng
Duke University
Durham, NC, 27708
yuanqi@cs.duke.edu

Steve Gu †
Duke University
Durham, NC, 27708
steve@cs.duke.edu

Carlo Tomasi
Duke University
Durham, NC, 27708
tomasi@cs.duke.edu

ABSTRACT

Motion synchrony, *i.e.*, the coordinated motion of a group of individuals, is an interesting phenomenon in nature or daily life. Fish swim in schools, birds fly in flocks, soldiers march in platoons, etc. Our goal is to detect motion synchrony that may be present in the video data, and to track the group of moving objects as a whole. This opens the door to novel algorithms and applications. To this end, we model individual motions as *video tubes* in space-time, define motion synchrony by the geometric relation among video tubes, and track a whole set of tubes by dynamic programming. The resulting algorithm is highly efficient in practice. Given a video clip of T frames of resolution $X \times Y$, we show that finding the K spatially correlated video tubes and determining the presence of synchrony can be solved optimally in $O(XYTK)$ time. Preliminary experiments show that our method is both effective and efficient. Typical running times are $30-100$ VGA-resolution frames per second after feature extraction, and the accuracy for the detection of synchrony is more than 90% as evaluated in our annotated data set.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding; I.4.9 [**Image Processing and Computer Vision**]: Applications; I.5 [**Pattern Recognition**]: Design Methodology, Applications

General Terms

Algorithms, Design, Experimentation, Measurement

1. INTRODUCTION

We study the new problem of finding motion synchrony in a video stream. Motion synchrony is an interesting phenomenon in both nature and daily life. For example, fish synchronize their swimming behavior in order to increase their chance for survival. People can synchronize their motions as well. Group dancing or marching often exhibit ordered motion with rhythms and harmonies. In soccer or basketball, people coordinate with each other to either launch an attack or defend against the other team.

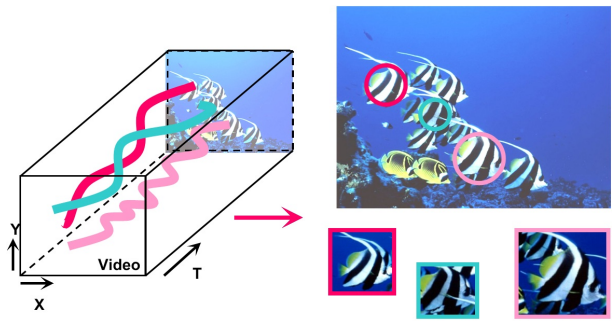

Figure 1: Video tube decomposition for selected fish in a school. These fish move in synchrony.

Finding motion synchrony is challenging, because telling individuals apart is often difficult. For example, each fish in a school looks like the others. How can each be tracked and distinguished reliably? Moreover, given that we know the position of each fish, how to determine if their motions are in synchrony or not? We introduce a few innovations to answer these questions. The general idea is to track the coordinated motions in a collective way and use the geometric relations between individuals to measure the degree of synchrony.

To represent coordinated motion, we decompose an image sequence into a set of disjoint *video tubes* of interest. A video tube is a space-time volume that groups the regions for a single object in consecutive frames into a long, twisting tube. Video tubes are richer than point trajectories. We model each video tube as a generalized cylinder that is equipped with a learnt object model. Different video tubes can share the same object model and hence are not independent of each other. The main novelty of our work is that we allow different video tubes to be spatially correlated and we model their spatial relation explicitly with a minimal spanning tree.

Explicit modeling of synchronous motion thus helps in two ways: First, it keeps tubes of similar objects explicitly distinct from each other, thereby making tracking reliable in the face of severe and widespread ambiguity, whether one is interested in *detecting* synchrony or not. Second, this new

*Area chair: Qi Tian

†Ying Zheng and Steve Gu contribute equally to this paper.

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.

type of motion model allows measuring synchrony when its detection is of interest.

1.1 Literature Review

Synchronization has been studied in both audio and video signals [6, 1, 19, 17, 5, 7]. For instance, synchrony measures have been derived [17, 7] for associating the movement of mouths to the oscillation of sound waves. To the best of our knowledge, detecting motion synchrony, *i.e.*, a coordinated motion of separate individuals within a single video, has not been well explored in the literature. Since synchrony is measured in the visual domain, the challenge is equivalent to that of tracking individual objects well. Multiple-object tracking has been studied in the literature [15, 12, 13, 11, 18, 14, 8] but most of this work focuses on people or vehicle detection. In one exception [8], spatial relations are used to track similar objects. We move a step further by defining the video tubes and using the geometric relations among them to measure and detect motion synchrony.

1.2 Our Contributions

Our contributions are as follows: First, we propose to detect the motion synchrony within a single video stream. This opens door to novel algorithms and applications. Second, we introduce the notion of coordinated video tubes to capture and measure motion synchrony. Third, we give an efficient algorithm to compute the K geometrically correlated video tubes in a video block of T frames of resolution $X \times Y$ in $O(XYTK)$ running time with a small constant. The efficiency of the algorithm makes our method highly practical for fast video processing. Fourth, we collect and annotate a data set for evaluating synchrony detection. This data set can serve as a benchmark for further study.

Preliminary experiments show that our method for detecting motion synchrony is both effective and efficient. A set of video tubes in an image sequence can be found at $30-100$ (VGA) frames per second, and the accuracy of synchrony detection is better than 90% in our annotated data set.

2. FINDING VIDEO TUBES

Let V be a $X \times Y \times T$ space-time block where T is the number of frames and each image frame has $X \times Y$ pixels. A tube $U \subset V$ is modeled as a generalized cylinder that passes through each frame exactly once. Specifically, a tube U is represented as a tuple $U = (\mathcal{X}, \mathcal{M}, r)$ where $\mathcal{M}$ is the object model, and the sequence $\mathcal{X} = (x_1, x_2, \cdots, x_T)$ specifies the location of disk centers with radius r in each frame. The model $\mathcal{M}$ is a set of suitably defined appearance features that describes the objects of interest.

2.1 Geometric Relation among Tubes

To specify the geometric relations among multiple video tubes, we compute the Euclidean minimal spanning tree (EMST) among tube centers in the first frame and the tree deforms when tracked through the video sequence. The amount of deformation measures the deviation from synchrony. The reason for working with EMST is to group all tubes into a single component by edges of minimal lengths.

Denote the tree $\mathcal{T}_f = \langle \mathcal{V}_f, \mathcal{E} \rangle$ as the result of tracking the EMST $\mathcal{T}_1$ in the first frame all the way to frame f. Here $\mathcal{V}_f = \left\{ x_1^f, x_2^f, \cdots, x_k^f \right\}$ is the set of centers of tubes $U_1, U_2, \cdots, U_k$ in frame f, and $\mathcal{E}$ specifies the $k-1$ spatial constraints among the k tubes, as captured by the constant topology of the spanning tree computed in the first frame.

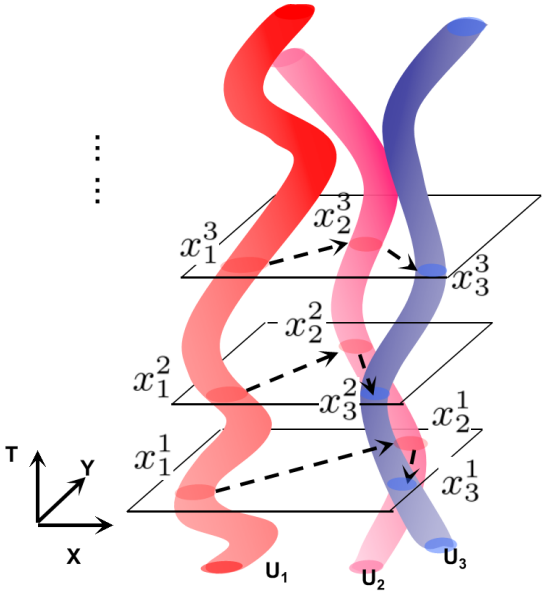

Figure 2: Video tubes and their geometric relations in the spatial and temporal domain.

2.2 The Coordinated Motion Model

We build video tubes frame by frame. Let $\mathcal{M}_1, \cdots, \mathcal{M}_k$ be the object models for tubes $U_1, \cdots, U_k$. Given the set $\mathcal{V}_f$ of positions of the k tube centers in frame $f-1$, the cost of extending these tubes to frame f is:

$$\text{Cost}(\mathcal{V}_f) = \sum_{i=1}^{k} E(x_i^f; \mathcal{M}_i) + \lambda \sum_{(i,j)\in\mathcal{E}} \|\delta(i,j,f)\|^2 \qquad (1)$$

where the *visual inconsistency* term $E(x_i^f; \mathcal{M}_i)$ measures the visual dissimilarity between the disk centered at x_i^f and the object model $\mathcal{M}_i$ associated to the i^{th} tube, as measured in previous work [9, 8], and the norm of the vector

$$\delta(i,j,f) = (x_i^f - x_j^f) - (x_i^{f-1} - x_j^{f-1}) \qquad (2)$$

measures the *pattern deformation* of an edge in the minimal spanning tree between frames $f-1$ and f. The constant λ is a regularization parameter that balances the penalties between visual inconsistency and pattern deformation.

The objective is therefore to find the optimal arrangement $\hat{\mathcal{V}}_f$ for which the cost $\text{Cost}(\hat{\mathcal{V}}_f)$ is minimal among all possible choices of configuration $\mathcal{V}_f$:

$$\hat{\mathcal{V}}_f = \arg\min_{\mathcal{V}_f} \text{Cost}(\mathcal{V}_f) \ . \qquad (3)$$

This optimization problem is solved in Section 4 below.

3. MEASURES FOR SYNCHRONY

We use the spatial relation determined by the spanning tree to measure the synchrony and account for the *structural difference* between two consecutive trees. Two measures, μ_1 and μ_2, are used to capture first-order and second-order

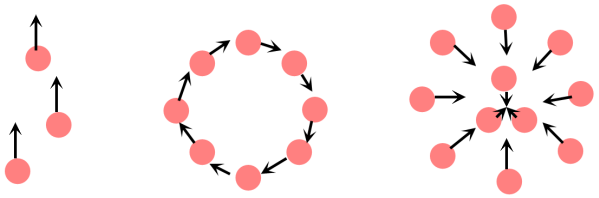

Figure 3: Left: First-order synchrony pattern. Middle and right: Examples of second-order synchrony. Each red dot represents a frame of the video tube and the vector indicates the motion direction.

deformation (lack of synchrony), respectively.

$$\mu_1 = \frac{1}{(K-1)(T-1)} \sum_{f=2}^{T} \sum_{(i,j)\in\mathcal{E}} \|\delta(i,j,f)\| \qquad (4)$$

The measure μ_1 for the degree of first-order deformation of a video clip of T frames accumulates pairwise vector differences of the form (2) in consecutive frames. The actual position of each tube is irrelevant. What matters is the changes of their relative positions. For instance, μ_1 is zero for pure translation. However, μ_1 cannot capture nonlinear motions (see Figure 3 for illustration). We hence introduce the other measure μ_2 for the degree of second-order deformation:

$$\mu_2 = \frac{1}{(K-1)(T-2)} \sum_{f=2}^{T-1} \sum_{(i,j)\in\mathcal{E}} \|\delta(i,j,f+1) - \delta(i,j,f)\| \,. \qquad (5)$$

In practice, both deformation measures are used to determine the synchrony state by comparing μ_1, μ_2 to a fixed threshold. Those video clips with sufficiently low μ_1 or μ_2 are declared to have motion synchrony.

4. THE ALGORITHM

We give an efficient algorithm to solve the optimization in Equation (3). The speedup is significant thanks to the use of the generalized distance transform [2], similarly to what is done for the matching of pictorial structures in human body parsing [3] and object recognition [4] and tracking [10].

The visual inconsistency $E(x_i^f; \mathcal{M}_i)$ can be evaluated in $O(1)$ or constant time [9, 8] with the help of the integral-image structure. In order to optimize the cost model, we start from the leaves in the spanning tree. We compute the optimal position for each leaf $x_l^f \in \mathcal{V}_f$, given each possible position x_p^f of the leaf's parent:

$$\mathrm{opt}_l(x_p^f) = \arg\min_{x_l^f} [E(x_l^f; \mathcal{M}_l) + \|\delta(l,p,f)\|^2] \,. \qquad (6)$$

Then, recursively, we compute the minimum-cost position for each intermediate node x_i^f – tracing back from leaf nodes – for all the possible positions x_p^f of that node's parent:

$$\mathrm{opt}_i(x_p^f) = \arg\min_{x_i^f} [E(x_i^f; \mathcal{M}_i) + \|\delta(i,p,f)\|^2 + \sum_{j\in C_i} \mathrm{opt}_j(x_i^f)] \qquad (7)$$

where C_i is the set of the children of x_i^f in the spanning tree. Finally, we compute the cost of the root x_r^f:

Figure 5: The *Fish* data set contains random fish motion (left) and fish circling in synchrony (right).

$$\mathrm{opt}_r(x_r^f) = \arg\min_{x_r^f} \left\{ E(x_r^f; \mathcal{M}_r) + \sum_{j\in C_r} \mathrm{opt}_j(x_r^f) \right\} \,. \qquad (8)$$

If implemented naively, this dynamic programming scheme would take time $O(X^2Y^2K)$. The complexity is further lowered to $O(XYK)$ by using the generalized distance transform. We refer to the paper [2] for details.

The overall algorithm is simple: First, compute tube models $\mathcal{M}_i$ and their minimum spanning tree $\mathcal{T}_1$ in the first frame. For subsequent frames, do the following:

Step 1: Use the algorithm above to find the optimal $\mathcal{V}_f$ defined in Equation (3)

Step 2: Extend the video tubes to frame f and update deformation measures (4) and (5)

Step 3: Set $f \leftarrow f+1$ and repeat until $f = T$.

Computing EMST takes $O(K \log K)$ time. In applications, $\log K \ll XY$, so the overall running time is $O(XYKT)$.

5. EXPERIMENTS

Typical video sequences are very long and objects often come in and out of the field of view. We therefore chop long sequences into clips of small duration and apply our method for finding correlated video tubes and motion synchrony to these short video clips. We collect 4 video sequences and chop them into 200 short video clips. We annotate each video clip by visual inspection as to whether it contains motion synchrony (See Figure 5 for examples).

We use SIFT features [16] for the tube models $\mathcal{M}_i$ and nearest neighbor matching [9] to evaluate visual inconsistency in Equation (1). Object models are constructed from a few windows specified by a user around some of the object in the first frame. Except for SIFT feature detection, running times range from 30 to 100 frames per second depending on how many video tubes are found. The program is mostly written in C++/MEX with the visual interface implemented in MATLAB.

Sample results are shown in Figure 4 and synchrony detection accuracy is shown in Table 1. The average accuracy for detecting synchrony is more than 90%. The code and data is available at `http://www.cs.duke.edu/~yuanqi/`.

6. CONCLUSIONS

We give an efficient algorithm to track synchronous motions in video, and use variations in the spatial pattern of the objects to measure the degree of motion synchrony. Preliminary results show that our method achieves more than 90% detection accuracy.

Figure 4: The left column show the query images. The rest show detected examples of motion synchrony.

Table 1: Motion synchrony for 4 categories.

Sequences	Airplane	Birds	Fish	Horses
# frames in the video	1025	2000	2200	1875
# chopped clips	50	50	50	50
# synchronous clips	35	21	23	30
# true positive	31	19	15	28
# true negative	14	28	26	20
# correctly classified	45	47	41	48
Detection accuracy	0.90	0.94	0.82	0.96

Video tubes can also be used in other applications such as automatic object highlighting, interactive object retrieval, and efficient memory storage of video segments. Our algorithm is efficient for these tasks. We leave the investigation of these applications for future work.

7. ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under Grant No. IIS-1017017 and by the Army Research Office under Grant No. W911NF-10-1-0387.

8. REFERENCES

[1] M. Beal, N. Jojic, and H. Attias. A graphical model for audiovisual object tracking. *PAMI*, 25(7):828–836, 2003.

[2] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical Report TR2004-1963, Cornell Computing and Information Science, 2004.

[3] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[4] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[5] J. Fisher and T. Darrell. Speaker association with signal-level audiovisual fusion. In *IEEE Transaction on Multimedia*, 2004.

[6] J. Fisher, T. Darrell, W. Freeman, and P. Viola. Learning joint statistical models for audio-visual fusion and segregation. In *NIPS*, 2001.

[7] S. Gu and C. Tomasi. Phase diffusion for the synchronization of heterogenous sensor streams. In *ICASSP*, pages 1841–1844, 2009.

[8] S. Gu and C. Tomasi. Branch and track. In *CVPR*, 2011.

[9] S. Gu, Y. Zheng, and C. Tomasi. Efficient visual object tracking with online nearest neighbor classifier. In *ACCV*, pages 267–277, 2010.

[10] S. Gu, Y. Zheng, and C. Tomasi. Linear time offline tracking and lower envelope algorithms. In *ICCV*, 2011.

[11] R. Hess and A. Fern. Discriminatively trained particle filters for complex multi-object tracking. In *CVPR*, pages 240–247, 2009.

[12] Z. Khan, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *PAMI*, pages 1805–1918, 2005.

[13] B. Leibe, K. Schindler, and L. V. Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, pages 1–8, 2007.

[14] K. Li, E. Miller, M. Chen, T. Kanade, L. Weiss, and P. Campbell. Computer vision tracking of stemness. In *ISBI*, pages 847–850, 2008.

[15] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, pages 2953–2960, 2009.

[16] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[17] H. Nock, G. Iyengar, and C. Neti. Assessing face and speech consistency for monologue detection in video. In *Proc. ACM Multimedia*, pages 303–306, 2002.

[18] A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *CVPR*, pages 666–673, 2006.

[19] M. Sargin, Y. Yemez, E. Erzin, and A. Tekalp. Audiovisual synchronization and fusion using canonical correlation analysis. *IEEE Transactions on Multimedia*, 9(7):1396–1403, 2007.