2007－CVIM－157 (29)
2007／1／12

Tracking People in Crowds by Feature Point Cluster Analysis Based on Spatial and Frequency Domain Cues

Jonathan Sahagun, Takahiro Okabe, and Yoichi Sato
The University of Tokyo

The performance of detection and tracking systems greatly depends on the quality of its information sources. For tracking humans in crowded environments, this becomes especially important due to the additional challenges of heavy occlusions and the sheer number of target objects. Recent works based on feature point tracking and clustering have already begun to use motion cues in conjunction with spatial cues, and have been met with varying levels of success and improvement. We now propose the use of a third class of cues based on frequency domain analysis to enable a better understanding of human activity in crowded scenes. We present a system that takes advantage of multiple domains of information by using domain-specific methods such as motion trajectory and gait frequency analysis. The advantages of our method shall be shown as well through presentation of experimental results.

1. Introduction

Detection and tracking is a field with more than 20 years of research and development history [1]. Enabled by the emergence of more powerful computing technology, detection and tracking systems in the present day have potential use in the modern day industries of logistics and security, and other relevant fields.

From a technical standpoint, the specific problem of detecting and tracking humans in highly crowded situations, despite its significance in modern day industries, still poses a high level of complexity since it involves several challenges at once, such as occlusion, multiple tracking, and difficulty of segmentation. It differentiates itself from other detection and tracking tasks in a very important regard: the sheer number of objects. As a direct result, problems that could have been negligible in the case of single or sparse detection become major issues in the context of crowds. Occlusion happens on a constant basis, segmentation techniques and background subtraction become of little use, and the large number of objects necessitates careful consideration with regards to computational load.

Thus, an effective system would be resilient to the inherent complexities posed by the crowd-tracking problem, and at the same time be computationally effective and elegant to simultaneously handle multiple objects with acceptable time complexity.

The techniques that have been proposed thus far to solve this problem cover a wide variety of approaches and design philosophies. Works such as those of Zhao and Nevatia [2,3] and Ramanan and Forsyth [4] rely on detailed geometric models of the human anatomy to effectively identify and track the motion of humans. One disadvantage for works of this nature is their inherent inability to identify humans and human activity that fall outside of the internally defined shape and motion state models. To attempt to empower such a system with such robustness to handle a wide variety of human motions would entail introducing more and more complexity on the internal system model. Furthermore, it would be difficult, to extend these models to identification and motion tracking scenarios that deal with other general types of objects, such as insects, microorganisms, or vehicles, since the main computational modules would be tied to a very specific type of target object.

On the other end of the design philosophy spectrum, systems have been introduced that extract only general spatial and motion information, and base their conclusions and results only on such generic cues. These systems, while not being able to match the potential for detection and tracking accuracy afforded by using target-specific models, have the potential of being able to handle arbitrary motion patterns for any specific target object, and in some cases, even able to handle arbitrary shapes and objects. In most of these works, clustered feature points are used as the basic indicator of an instance of a target object in the scene. Tu and Rittscher in their work [5], for example, performed a geometry-based clustering operation using their own Emergent Labeling technique on a weighted graph constructed from vertices derived from the results of simple background subtraction.

The works of Rabaud and Belongie [6] and Brostow and Cipolla [7], considered to be the main inspirations for this work, fall under the second class of tracking systems as described above. In both of these

works, motion data derived from Optical Flow tracking [8] of Shi-Tomasi-Kanade feature points [9] is used to construct a weighted graph that would describe the relations of points with respect to belonging to the same target object. The concept behind the utilization of weighted graphs shall be discussed in detail in section 2.2.

Rabaud and Belongie in their system assert that feature points sharing the same affine motion have a high likelihood of belonging to the same object, and perform RANSAC clustering based on that assumption. Brostow and Cipolla, on the other hand, use a Bayesian clustering scheme mainly based on the assertion that feature points belonging to the same object would exhibit little variance in their point-to-point distance while tracing their motion trajectories.

Our system adopts the same general methodology of feature point tracking and clustering. Like previous related works, we utilize spatial and motion cues such as bounding box constraints and motion trajectory analysis. However, we introduce a new class of cues based on frequency domain analysis, which we believe will be able to provide better clustering and segmentation performance than with motion-based data alone. For human beings, this is especially true since in casual scenarios, walk cycles differ from person to person, even if they are walking in the same general direction. Two people walking together may be easily mistaken as one entity by virtue of motion analysis alone, but may be successfully separated by applying frequency analysis on their individual walk cycles.

Also, in this work we demonstrate the usability of object detection as a suitable replacement for background subtraction in cases where the latter is difficult, if not impossible, to implement.

2. Proposed Method

2.1. A Clustering-Based Approach

Our system uses weighted graphs as the main abstract representation of humans in a clip of motion video. We define two graphs, $G_I = \{V, E_I\}$ and $G_F = \{V, E_F\}$, as the initial hypothesis and final configuration that describe the distribution of humans in any single frame of motion video. In this notation, V represents the set of labeled vertices that correspond to and contain information on each tracked feature point as described in section 2.2. The set of weighted edges E_I, on the other hand, is the set of point-to-point relationships during the initial hypothesis with respect to their likelihoods of belonging to the same human, while E_F represents the non-weighted edges defining the final configuration. G_I and G_F are related by an overall clustering function C, which shall be defined in later sections. Each connected component in the result G_F is then considered to correspond to a single instance of a human being in that particular video frame.

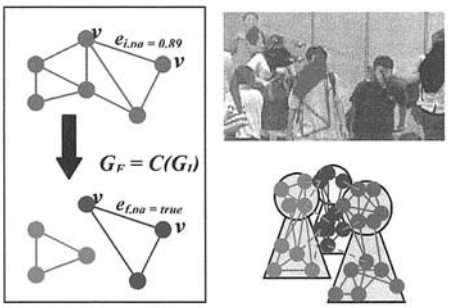


***Figure 1**: (Left) The clustering process. (Right) Figurative and actual illustrations*

2.2. Features and Trajectory Generation

Before the actual clustering operation, needed information to construct the initial hypothesis G_I is obtained. Each element v_i of the vertex set V corresponds to a feature point and its relevant parameters, notably, but not limited to, its spatial X and Y locations. Feature points in this work are defined as synonymous to Tomasi-Kanade features, and are obtained using the technique described in [9]. Aside from spatial information, motion-related parameters are also obtained through tracking and trajectory tracing, using a standard implementation of Lucas-Kanade optical flow [8]. Occlusion and drifting during the tracking of each feature are then handled by a template matching based consistency check between the image window around the feature for the current frame and the corresponding image window in the frame in which the feature was first detected, as illustrated in Figure 2.

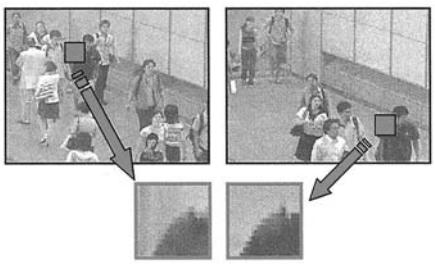


***Figure 2**: Each feature point's flow computation is further validated by comparing it with its initial state. Top left: Start frame. Top right: Current frame. Bottom row: image windows subject to normalized crosscorrelation.*

Each feature point's trajectory is traced forwards and backwards in time for 2N+1 frames, e.g., 129 frames in our experiments. The availability of future frames for forward tracing is emulated by implementing a delay buffer of length N for the video frames, and performing clustering operations on a delayed frame rather than the present frame, as is shown in Figure 3.

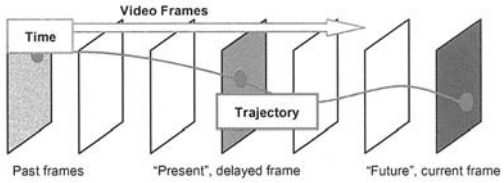


***Figure 3**: Emulation of future frames through the use of a delay buffer.*

We use a trace length of 129 frames as a balance of having a long-enough length for improved analysis and having a short-enough length to avoid unnecessarily discarding shorter yet perfectly suitable trajectories.

2.3. Object Detection Foreground Mask

To reduce the noise involved in selecting irrelevant features from the background, we make use of a foreground-background mask during the feature detection process. For this study, we make use of an object detector as a replacement for traditional background subtraction in constructing the selection mask. This is to compensate for the difficulty in obtaining background images from crowded scenes. We use an implementation of the hierarchical cascade of boosted Haar-like classifiers as described in [10,11], which in turn uses the Adaptive Boosting framework first introduced in [12]. A cascaded classifier trained for human heads from all views was synthesized, with the detector output interpreted as a mask image. This result is modified to include a region of pixels below every detected instance of a human head; i.e., the pixels occupying the corresponding human torso. While this method could not be compared with traditional methods in terms of background-foreground segmentation accuracy, it has been seen to be more suitable for our specific purpose.

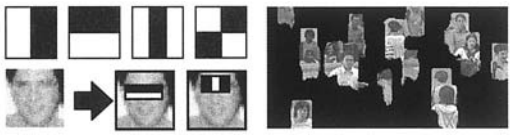


***Figure 4**: (Left) Set of Haar-like features for object detection (top row), and an example of its applications. (Right) Human head detection results as selection mask.*

2.4. Weighted Edges and Scoring

Each edge weight $e_{i,pq}$ corresponding to the edge with endpoints v_p and v_q in graph G_I is computed as the result of an overall weighting function of the properties of each vertex. In this work, we denote each edge weight as a product of different weighting components (interchangeably referred to as "scores") as follows:

$$e_{i,pq} = s_{pq,space} \cdot s_{pq,motion} \cdot s_{pq,fourier} \qquad (1)$$

Each score s_{pq} represents the relation of vertices v_p and v_q by virtue of spatial, motion, and Fourier analysis, respectively, and is computed using related properties such as spatial distance, frequency, and phase.

For both sets E_I and E_F, a nonzero value for e_{pq} denotes the existence of the edge, while zero would denote its nonexistence. For E_I, each component score and edge weight is real-valued from 0 to 1.

Spatial Score and Initial Graph

The spatial score $s_{pq,space}$ is implemented as an initialization for the hypothesis graph G_I. It is a simple constraint similar to the bounding box defined in [6], and serves to reduce the computational load of later stages by eliminating highly unlikely feature pairs. The bounding box criterion states that two points are possibly connected if their distances from each other along the X and Y axes (which are denoted by $|v_{p,x}-v_{p,x}|$ and $|v_{p,y}-v_{p,y}|$) are less than the target object's expected dimensions, here denoted by x_{bound} and y_{bound}.

$$s_{pq,space} = \begin{cases} 1 & ,|v_{p,x} - v_{q,x}| \le x_{bound}, |v_{p,y} - v_{q,y}| \le y_{bound} \\ 0 & ,otherwise \end{cases} \qquad (2)$$

Motion Analysis

The base concept for motion analysis is that any two points on the same rigid object undergoing pure translational motion would maintain a constant separation from each other at any given frame of time. This definition is analogous to that of parallel lines, and this similarity indeed manifests itself in during trajectory visualization. We consider two main arguments in the computation of the motion analysis score.

Coherent Motion

We measure coherence of motion by utilizing the set of feature point trajectories as exemplified by Figure 5. We take the standard deviation σ_{pq} in pixels of the distance l_{pq} between the two feature points in question as they move across the scene (Equation 3), during the time window from t_i to t_f in which they have coexisted (illustrated in Figures 6 and 7). This is a direct measure of the points' deviation from the previously mentioned ideal description of constant separation.

$$\sigma_{pq} = \sqrt{\frac{\sum_{t=t_i}^{t_f} l_{pq}^{\,2}}{t_f - t_i} - \left(\frac{\sum_{t=t_i}^{t_f} l_{pq}}{t_f - t_i}\right)^2} \qquad (3)$$

The standard deviation σ_{pq} is then assigned to a likelihood score $s_{pq,coherence}$ using a simple mapping function (Appendix A).

***Figure 5**: Extracted trajectories from motion video.*

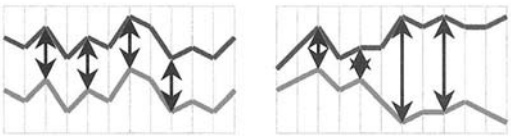


***Figure 6**: Left: Constant distances imply high likelihood of belonging to the same object. Right: high deviation implies different objects.*

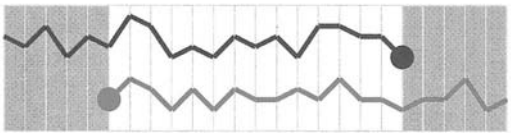


***Figure 7**: The standard deviation is computed only for the time window in which the two feature points have coexisted (highlighted region).*

Coexistence

Unlike related works that artificially extend the length of trajectories through extrapolation techniques [6,7], the coherent motion score in our study is computed using trajectory data only for the frames in which the two feature points have coexisted. This was done to eliminate any errors or discrepancies that may arise from the data estimation involved in extrapolation.

However, this would imply that trajectories would have varying coexistence intervals $(t_f - t_i)$. Furthermore, smaller values for $(t_f - t_i)$ imply that the source trajectories contain less information, and would therefore entail less reliable results for coherent motion analysis. Thus, we introduce an additional score that is the result of using the coexistence interval $(t_f - t_i)$ as a weighting metric. As with the coherent motion score $s_{pq,coherence}$, we assign the coexistence interval $(t_f - t_i)$ to a likelihood score $s_{pq,coexistence}$ using the common mapping function described in Appendix A. The value of $s_{pq,motion}$ is then obtained as a simple product of the two component scores.

$$s_{pq,motion} = s_{pq,coherence} \cdot s_{pq,coexistence} \qquad (4)$$

Fourier Analysis

Aside from physical separation and differences in general trajectories, one key parameter that can be exploited is the set of frequency domain characteristics of the gaits of individual humans. Outside of coordinated and deliberately synchronized movement, humans generally exhibit different frequencies and phase during their walk cycles. This periodic motion becomes evident in visualizations of feature point trajectories, as was seen previously in Figure 5.

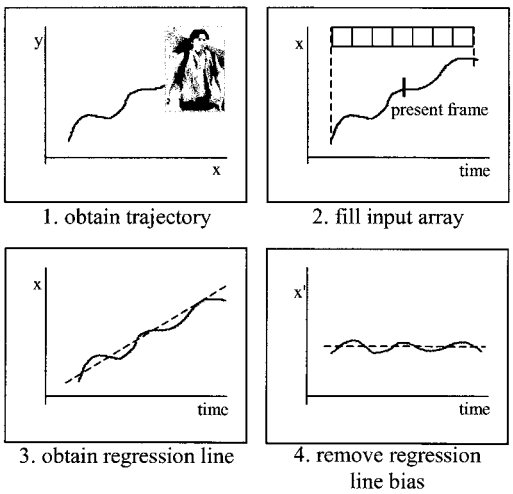
As was explained earlier, the inclusion of a set of scores based on frequency analysis would enable the system to handle the scenario in which two people are walking closely together and towards the same general direction. While motion analysis alone might manifest their differences as nonzero but negligible standard deviation values, frequency analysis would be able to reveal any significant differences in the frequencies of their individual and unique walk cycles despite the fact that they were moving in the same direction.

The Fourier analysis score consists of two components, corresponding to the gait frequency and phase.

$$S_{pq,fourier} = S_{pq,frequency} \cdot S_{pq,phase} \quad (5)$$

Before the computation of the two component scores, each feature point's trajectory data is preprocessed for frequency analysis. Input arrays for Fourier transform computation are filled with each feature point's X and Y positions as functions of time, starting at the current frame assigned to the array's middle position, and stretching forwards and backwards through time up to the array's extents. Feature points with trajectories that could not be able to fill the array are considered too short and are discarded; otherwise their trajectories are conditioned for Fourier analysis as illustrated in Figure 8.

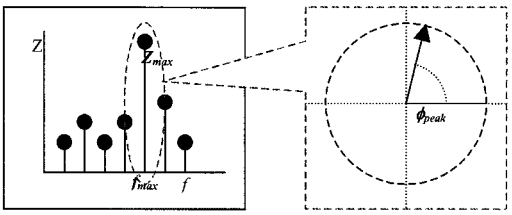
An array length of 64 frames was seen to be a balanced value that is long enough to provide sufficient time frames for analysis while being short enough to accommodate an adequate number of trajectories. Setting the array length too high would lead to unnecessarily deleting shorter but suitable trajectories, so care was taken in choosing this number. This is equivalent to about 1 second of motion video, or about 1.5 to 2 gait oscillations.



1. obtain trajectory
2. fill input array
3. obtain regression line
4. remove regression line bias

***Figure 8**: Preprocessing a trajectory for Fourier analysis*

The complex Fourier transforms for the waveforms for each feature points' X and Y coordinates are then computed. For each transform result, two values are obtained: the location f_{peak} of the peak complex magnitude Z_{max}, and the complex phase at the peak location, denoted by ϕ_{peak}.



1. obtain transform peak
2. obtain phase

***Figure 9**: Illustration of the frequency analysis process*

Scores $s_{pq,frequency}$ and $s_{pq,phase}$ are then computed based on the differences Δf_{peak} and $\Delta\phi_{peak}$ of the frequency and phase values respectively for each waveform pair using a suitable mapping function as described in Appendix A.

These score computations are done for movements in both X and Y components. Thus, the final frequency domain analysis score $s_{pq,fourier}$ is the product of four separate values, as shown in Equation 6.

$$S_{pq,fourier} = S_{pq,frequency,X} \cdot S_{pq,phase,X} \cdot S_{pq,frequency,Y} \cdot S_{pq,phase,Y} \quad (6)$$

2.5. Betweenness Centrality Clustering

After obtaining the scores necessary to construct the initial hypothesis graph $G_I = \{V,E_I\}$, it is clustered based on the measure of betweenness centrality [13]. In general, the centrality of a vertex or an edge in a graph is a numerical representation of its performance or importance in the context of flow and connection.

$$c_B(p) = \sum_{s \in V, t \in V, p \in E} \frac{n_{st}(p)}{n_{st}} \quad (7)$$

The edge betweenness centrality $c_B(p)$ for an edge p in graph $G=\{V,E\}$ in which V and E are vertex and edge sets is defined in equation 8. The value $n_{st}(p)$ is defined as the number of shortest paths from vertices s and t passing through edge p, while n_{st} is the overall number of shortest paths from vertices s and t. Real-valued weighted graphs containing experimental data often offer the simplest case as the shortest path from any vertex pair s and t is usually unique (there are no parallel shortest paths). In this case, $n_{st}(p)$ and n_{st} are both equal to 1, thus making the edge betweenness centrality in this case a counter of shortest paths within the graph passing through edge p.

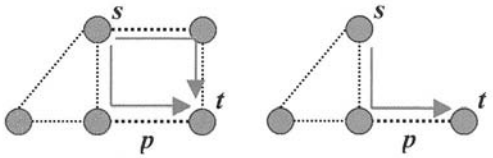


***Figure 10**: A simple example of betweenness centrality computation.*

Figure 10 is a simple explanation of the betweenness centrality computation. At the left, two shortest paths exist between vertices ***s*** and ***t***. One of them passes through edge ***p***, so $[n_{st}(p)/n_{st}]$ in this case is 0.5. The right hand diagram is the usual case for weighted graphs, in which there is only one shortest path between vertices ***s*** and ***t***. This would give $[n_{st}(p)/n_{st}]$ a value of 1.0 if the path passes through ***p***, which would make the summation in Equation 8 a simple counter of all the shortest paths that pass through edge ***p***.

Our use of betweenness centrality is best explained through Figure 11. Among other scenarios, high betweenness centralities would manifest on bridge edges, and edges with low weights, the latter's reason being that the computation of shortest paths is heavily based on edge weights for edge-weighted graphs.

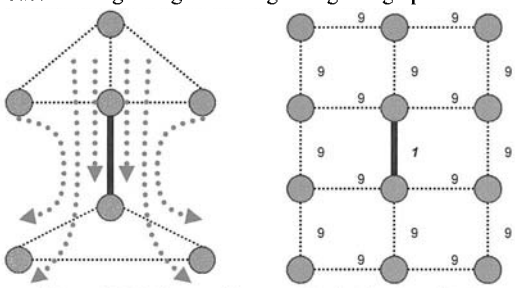


***Figure 11**: Bridges and low-weighted edges have high betweenness centrality.*

In our system, low weights would manifest between points that exhibit low likelihoods of belonging to the same object, as detailed in previous sections. Furthermore, residual bridge edges may also appear, as shown in Figure 12. As shown, betweenness centrality was successful in consolidating valid clusters (colored lines) while discarding weak and bridge edges (black lines).

***Figure 12**: Bridge and weak edges..*

In our system, we compute the betweenness centrality of each edge in G_I and discard high centrality values through application of a threshold. The remaining edges would comprise the edges of the output graph G_F.

$$G_F = C(G_I) = \left\{ v \in V, e_f \in E_F \mid e_f = \begin{cases} 1 & , c_B(e_i) \le c_{b,thr} \\ 0 & , c_B(e_i) > c_{b,thr} \end{cases} \right\} \quad (8)$$

After this clustering step, the resultant graph G_F would not contain low-scored edges and bridges, and would ideally consist only of connected components of relatively strong connection.

3. Experimental Results

Our scheme was evaluated by measuring relevant performance parameters while being applied to the main interest region of a video clip of a typical crowded scene.

We select the main interest region for our experiment to be the main floor as illustrated in Figure 13. All evaluations were performed only within this region, and only humans standing on the main floor are counted. We restrict our evaluation within this region since we would want the evaluation environment to have as much constant parameters as possible, most notably lighting and level of occlusion. The source video itself was taken using a prosumer level video camera. Prior to processing, we have deinterlaced it to produce motion video with 720x240 pixels at 60 frames per second. The main interest region is populated by 10 to 20 people at a time.

***Figure 13**: Use of the scene's main floor as the main interest region. Humans are counted as long as their feet touch the floor indicated by the shaded region.*

3.1. Figures of Accuracy

We measure system performance using four general parameters: *Hit Rate*, *Fragmentation*, *False Positive Rate*, and *False Negative Rate*. We perform our experiment by taking random frames and their clustering results, and performing manual, by-hand counting. We use a fully human-supervised data

extraction method as opposed to automated methods to prepare our data in an effort to reduce errors and increase accuracy of results. Furthermore, some parameters, notably the fragmentation rate, can only be measured using manual evaluation.

Hit Rate is defined as the ratio of the number of humans correctly tagged with clusters to the total number of humans in the scene. False Positive Rate is defined as the ratio of the number of incorrectly formed clusters without associated human targets to the total number of clusters in the scene, while the False Negative Rate is defined as the ratio of the number of humans without any associated clusters to the total number of humans in the scene.

We define Fragmentation in this work to be a figure that describes the accuracy of the correspondence of clusters to human targets. It is defined as the ratio of the number of clusters that lie within humans, to the number of humans that have been identified with at least one cluster.

$$Fragmentation = \frac{clusters\_with\_humans}{humans\_with\_clusters} \qquad (9)$$

A fragmentation value of 1.0 indicates overall ideal operation, in which each human being in the scene is assigned to one unique cluster. Values less than 1.0 would be indicative of false negatives and cluster merges (defined as the scenario in which multiple humans are grouped into a single cluster and are erroneously considered as a single entity), while values above 1.0 would indicate the presence of multiple clusters within the same human being. It was also noted that almost all cases of multiple clusters were related to articulation of the human body – i.e., separate clusters were assigned to jointed body parts such as arms, shoulders, and legs.

***Figure 14**: Most fragmentations are attributed to human body articulations.*

Table 1 is a summary of the final experimental results, listing each performance parameter's average value for 100 randomly selected frames. Also, to account for the existence of small-sized clusters – i.e., clusters that have only 4 or fewer vertices, we present two versions of the fragmentation computation: one that considers small-sized clusters as legitimate clusters ("Fragmentation"), and another that treats them as noise ("Modified Fragmentation").

Hit Rate	False Negative Rate	False Positive Rate	Fragment-ation	Modified Fragment-ation
86.57%	13.43%	0.24%	125.51%	89.97%

***Table 1:** System Performance Parameters, averaged over 100 randomly-picked frames*

3.2. Analysis

Table 1 can be considered as a numerical summary of system performance. We see acceptable performance especially in terms of false positives. The main reason for this is seen to be the foreground selection mask presented in section 2.3. The system's false negative rate is acceptable, with the feature point selection scheme seen as the main factor affecting its performance. In particular, humans facing away from the camera and those wearing plain clothing would lack enough texturedness as defined in [9]. They would therefore have fewer, erratic feature points and thus have edges of lower quality that are prone to deletion during the clustering process.

Fragmentation is also satisfactory; with the main cause for most cases of single person multiple clusters being human body articulation.

As for the system's ability to separate individual humans, the use of Fourier analysis as one of the edge metrics would be immediately evident upon observation of individual frames. In Figure 15, we see humans situated close together and moving in the same general direction, but were still separated by virtue of the fact that their gaits were independent of each other, and hence exhibit different frequency domain properties.

***Figure 15**: Successful separation of humans by virtue of the Fourier analysis scores*

4. Conclusion and Future Work

We have presented a system that utilizes Fourier analysis in addition to trajectory analysis during feature point tracking and clustering. It has shown acceptable if not good performance, especially in the specific scenario in which two humans are walking close together and towards the same general direction. This sensitivity, however, has also led the system from time to time to incorrectly classify separate human body parts as separate objects. This phenomenon of multiple

clusters, however, has been observed to be relatively minimal and could be improved by further testing and development.

There are numerous areas for improvement for our system, especially in the area of fragmentation reduction, such as by utilizing a higher-level cluster merging scheme not different from the one used by [7]. Of course, adding additional metrics and observations for edge weights would also be a possibility.

Appendix

A. Mapping Functions

The mapping functions described in Section 2.4 are functions that are selected to match our requirements for mapping curve shapes. The input to such functions would be values such as phase, frequency, or trajectory standard deviation, and the output would be a score ranging from 0 to 1. A maximum value for each is also defined, such that any value outside of the range from zero to the maximum would be mapped to a score of 0.

Furthermore, two types of mapping functions are defined. A rising mapping would result in a score of 0 for a given input value of zero, gradually rising to 1 as the value approaches the maximum. The rising mapping function for this work was based on the squared sine function. The coexistence metric uses the rising mapping type.

A falling mapping value, on the other hand, would result in a score of 1 for a given input value of zero, gradually falling to 0 as the value approaches the maximum. The falling mapping function for this work was based on the squared cosine function. Metrics that use this type are the coherent motion metric and the frequency domain analysis metrics.

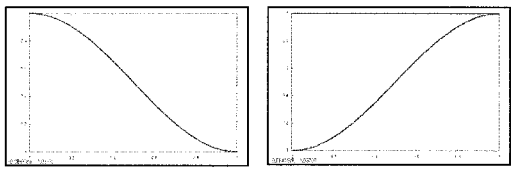


***Figure A-1**: Left: Falling mapping plot based on squared cosine function. Right: Rising mapping plot based on squared sine function.*

References

[1] I. Pavlidis, V. Morellas, P. Tsiamyrtzis, and S. Harp, "Urban Surveillance Systems: From the Laboratory to the Commercial World", In Proc. IEEE. Volume 89, Number. 10, pp.1478-1497, 2001.

[2] T. Zhao and R. Nevatia, "Tracking Multiple Humans in Crowded Environment", In Proc. Conference on Computer Vision and Pattern Recognition, Volume 2, pp.406–413, 2004.

[3] T. Zhao and R. Nevatia, "Tracking Multiple Humans in Complex Situations", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 26, Number 9, pp.1208-1221, 2004.

[4] D. Ramanan and D. A. Forsyth, "Finding and Tracking People from the Bottom Up", In Proc. Conference on Computer Vision and Pattern Recognition, Volume 2, pp.467-475, 2003.

[5] P. Tu and J. Rittscher, "Crowd Segmentation Through Emergent Labeling", ECCV Workshop SMVP, pp.187–198, 2004.

[6] V. Rabaud and S. Belongie, "Counting Crowded Moving Objects", In Proc. Conference on Computer Vision and Pattern Recognition, Volume 1, pp.705-711, 2006.

[7] G. Brostow and R. Cipolla, "Unsupervised Bayesian Detection of Independent Motion in Crowds", In Proc. Conference on Computer Vision and Pattern Recognition, Volume 1, pp.594-601, 2006.

[8] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", In Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81), pp. 674–679, 1981.

[9] J. Shi and C. Tomasi, "Good features to track" In Proc. Conference on Computer Vision and Pattern Recognition, pp.593-600, 1994.

[10] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", In Proc. Conference on Computer Vision and Pattern Recognition, p.511-518, 2001.

[11] P. Viola and M. Jones, "Robust Real-Time Face Detection", In International Journal of Computer Vision, Volume 57, Issue 2, pp.137-154, 2004.

[12] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting", In Journal of Computer and System Sciences (Proceedings of the Second European Conference on Computational Learning Theory - March, 1995), Volume 55, Issue 1, pp.119-139, August 1997.

[13] U. Brandes, "A Faster Algorithm for Betweenness Centrality", In Journal of Mathematical Sociology, 25(2), pp.163–177, 2001.