Tracking People in Crowds Using Feature Point Cluster Analysis Based on Spatiotemporal and Frequency Domain Cues

by

Jonathan Ancheta Sahagun

A dissertation submitted to

The Graduate School of Information Science and Technology

of

The University of Tokyo

in partial fulfillment of the requirements for the degree of

Master of Science

in

Information and Communications Engineering

Advisor:

Associate Professor Yoichi Sato

February 2007

Tracking People in Crowds Using Feature Point Cluster Analysis Based on
Spatiotemporal and Frequency Domain Cues

Copyright © 2007

by

Jonathan Ancheta Sahagun

Abstract

Tracking People in Crowds Using Feature Point Cluster Analysis Based on
Spatiotemporal and Frequency Domain Cues

by

Jonathan Ancheta Sahagun

Master of Science in Information and Communications Engineering
The University of Tokyo, Institute of Industrial Science

Associate Professor Yoichi Sato, Advisor

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. However, this set of cues is still far from complete with regards to representing the characteristics of human movement.

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, as well as a graph-clustering scheme that takes advantage of both weighted elements and graph topological structure.

The advantages of our method shall be discussed and shown through presentation of experimental results.

Table of Contents

List of Figures

1. Introduction

2. Related Research

3. Proposed Technique

4. Experiments, Results and Analysis

List of Equations

3. Proposed Technique

4. Experiments, Results and Analysis

List of Tables

Chapter 1

Introduction

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

Research regarding the characterization of human movement in crowded scenes in particular has seen quite a flurry of activity during the past recent years, mostly due to the availability of cheap yet powerful computing hardware that would be able to reasonably handle the high computational requirements associated with the task. Furthermore, it is a field that has seen quite an unprecedented divergence among researchers with regards to methods and problem solving philosophies.

In this research, we present a novel way of tackling the crowd-tracking problem. While there has been much research in this field, we differentiate our work from most others with our holistic approach to the problem. It is precisely because of this approach that we are able to easily sidestep most of the common roadblocks involved in the analysis of the motion of people in crowds.

We consider our method holistic since it approaches the problem from a top-down perspective. In our research, we consider a crowded scene as a *single unit* on which to apply our analysis and computations to come up with the locations and trajectories of individual humans. This is in contrast to the more common approach of considering a crowded scene as a conglomerate of multiple human units, in which multiple instances of an algorithm or model is required to track each human instance.

As an example of the advantages of our approach, consider a crowded scene with heavy occlusion. One approach would be to initialize multiple models describing the form and motion of people, and attempt to assign them to each human present in the scene. However, if a person is heavily occluded such that the model would no longer recognize his or her physical form, he or she will not be included in the final tracking results. While it could be argued that such models could be upgraded and modified to reach a level of sophistication that is resilient to problems such as occlusion, this performance improvement would come at the cost of added complexity. This is especially considering that multiple instances of such complex models would be required to track a crowd consisting of multiple people, thus resulting in a system whose performance degrades as the number of humans in the scene increases.

We avoid this problem by building a system that considers occlusion *from the start*; i.e., a system that considers a crowd as a single unit, with inter-human occlusion *as one of its fundamental properties*. In particular, our work sidesteps the occlusion problem by making as little assumptions on the human form as possible, and focus instead on the essential components and concepts surrounding human motion.

1.1. Motivation for Research

Pavlidis, et.al. [PMTH-01] have highlighted that the video surveillance research field has experienced a drastic shift in terms of its target audience. While it used to be a military-only field of interest, it now caters to a wide variety of uses and applications, most notably in the commercial security and surveillance field. Though in their work they have highlighted various negative factors that may impede the growth of commercial security and surveillance research, the overall outlook is positive, as illustrated in Figure 1.1.

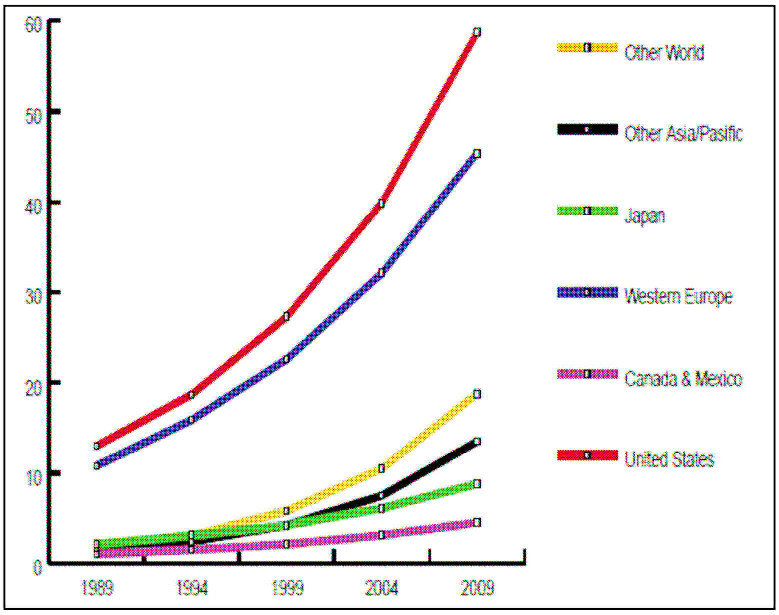


Figure 1.1: The security service market by region in billion US dollars. The numbers after the year 2000 are projections (source: The Freedonia Group)

The market projection figures alone testify to the significance of video surveillance research, with 30 percent growth for Japan from 2004 to 2009 amounting to around 20 billion US dollars (around 2.4 trillion Japanese Yen) according to the figure, with overall global figures amounting to a higher amount and an even higher rate of growth. Furthermore, real-world video surveillance scenarios often deal with multiple people and crowded scenes, thus adding to the significance of our research effort.

1.2. Statement of the Problem

A video surveillance system usually entails two operations: detection and tracking. In the context of image processing, detection means to confirm the presence of a target object, in our case, a human being, in a given image. In most cases, it is also required to report the target's location within the given image. Detection results are often measured in terms of performance using accuracy figures such as hit rates, miss rates, and false positives and negatives.

Tracking, on the other hand, is a motion-related task that involves the observation and monitoring of the motion of a target object as it moves across the space of interest. Object tracking usually implies the prior detection of the target object in question, though it is possible for some systems to perform detection and tracking simultaneously using a single set of computations. The performance of a tracking system is usually measured in comparison to ground truth trajectory data. If ground truth data is not available, one useful metric for performance is the duration of the acquired trajectory before loss of tracking.

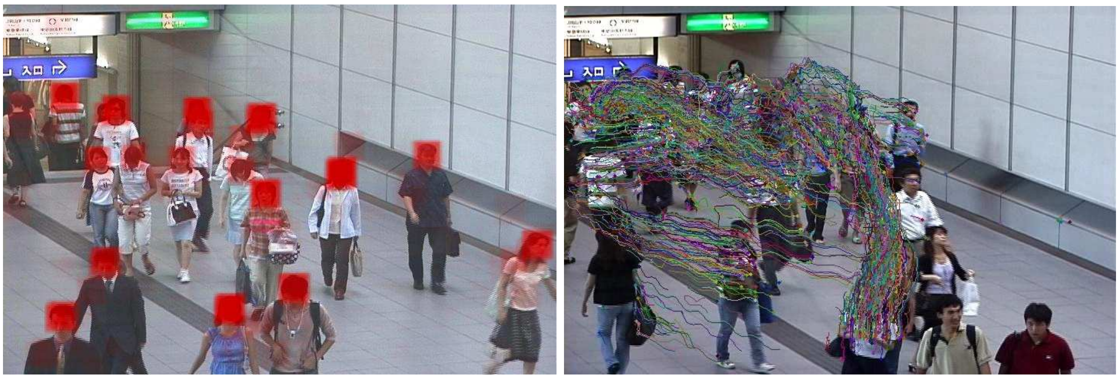


Figure 1.2: Face detection (left) and optical flow (right) are two examples of popular detection and tracking tasks in computer vision.

Thus, our work's main objective is to properly identify human beings in crowds, and successfully characterize their movements as they move across the scene.

1.3. Challenges

From a technical standpoint, the specific problem of detecting and tracking humans in highly crowded situations 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.

We identify three main roadblocks to successful detection and tracking. Probably the primary concern is the number of target objects that are expected to be in the scene. This immediately places a constraint on the choice of algorithms, models, and methods to be used, since a wrong choice of model or algorithm might impose an unreasonable computational complexity that compromise the feasibility of the detection and tracking task. An example for this has already been mentioned, that involves the inefficient practice of using multiple instances of a complex computational block to handle each individual target object.

Mentioned as well in the previous example is the problem of occlusion. Detection and segmentation techniques that rely on human anatomical models would fail in cases when a significant percentage of a target's body is occluded from view. This is especially true for crowded scenes, where inter-occlusion among humans occurs almost at a regular basis. For example, an anatomical model that assumes the presence of arms and legs will fare poorly when faced with an input image similar to Figure 1.3. In this scene, each person occludes parts of other people from view. Furthermore, appendages such as arms and legs enter and leave the occluded state quite often. In this particular frame of time depicted in the image, legs are practically invisible, save for a handful of people walking along the edge of the scene. The only part of the human anatomy that remains visible for a reasonable amount of time is the head, and it alone will not be able to satisfy any human anatomical model.

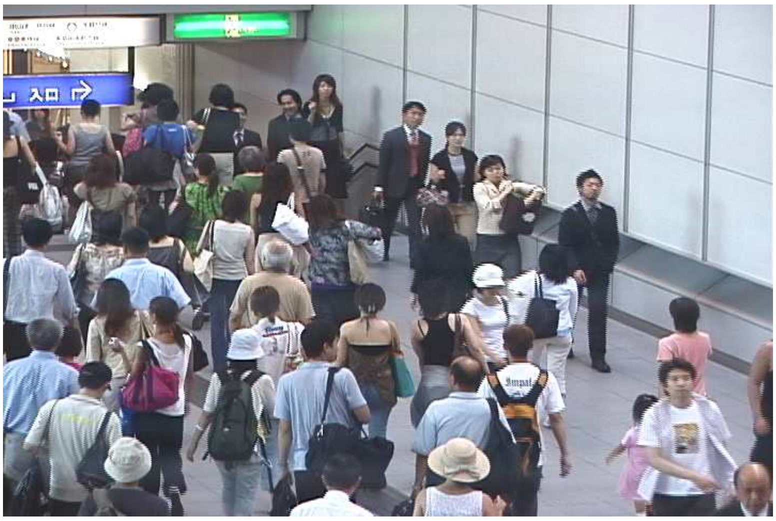


Figure 1.3: Only one out of the tens of people in this scene can be considered as fairly free from occlusion. Otherwise, people cover each other's arms, legs and bodies.

Another challenge when dealing with crowded scenes is the poor performance of standard background subtraction techniques. In computer vision, background subtraction is a very useful tool in reducing the problem space to foreground objects, which is usually the domain of interest for most problems. In object detection, applying background subtraction before the actual detection would not only improve computational performance, but would improve the detector's performance itself by excluding background regions that may be the source of noise that leads to false positives. Unfortunately for crowded scenes, background pixels will rarely reveal themselves due to the presence of people during most of the duration of interest, and would result in a background image result similar to Figure 1.4.

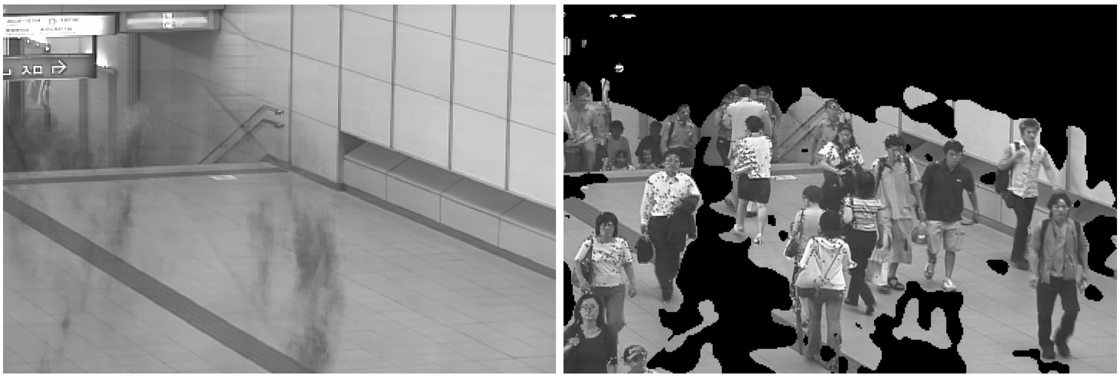


Figure 1.4: (left) Background image extraction result using the Median background model. (right) Background subtraction result using same background image.

As a consequence, background subtraction, which is considered as one of the fundamental components in any image processing engineer’s palette of tools, is not available during crowd analysis, thus potentially leaving the system open to degradations in performance caused by false positive detections, among other things.

Chapter 2

Related Research

The general task of tracking people is a wide encompassing field with an equally wide spectrum of valid problem solving approaches. This is true as well for tracking people in crowds, and the techniques that have been proposed thus far cover a wide variety of design philosophies, with their own advantages and weaknesses.

This chapter focuses on two general types of approaches that are directly related to the task of characterizing the motion of multiple people. Some works rely on detailed geometric models of the human anatomy to effectively identify and track the motion of humans. 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.

2.1. Model-Based Methods: A Bottom-Up Approach

Methods that rely on human anatomical models to achieve multiple tracking are considered to take on a *bottom-up approach*, in which the fundamental detection and tracking unit is a single human being, with the crowd being a complex structure consisting of multiple component units. It is said to be bottom-up, since the main modules and models during computations describe a single human unit, instead of considering the crowded scene as the single encompassing entity. We classify a detection and tracking scheme as bottom-up when it involves multiple instantiations of the main computational unit in order to characterize the positions and movements of an equal number of target objects.

Works of this nature have been presented with varying levels of success. Ramanan and Forsyth [RF-03] in their work, “Finding and Tracking People from the Bottom Up,” model 2D views of the human body as a puppet consisting of textured rectangular segments, as depicted in Figure 2.1.

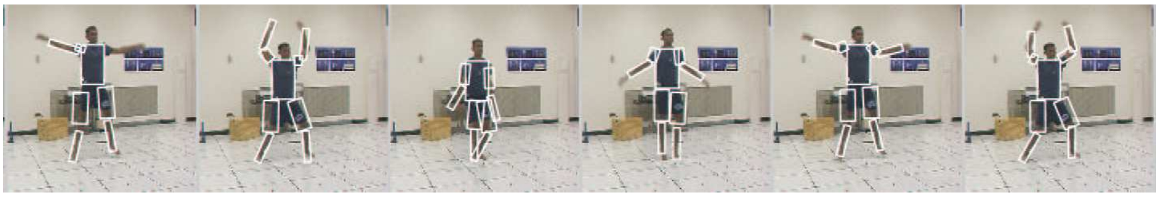


Figure 2.1: [RF-03] The human body is modeled as an assembly of individual body segments.

In their work, the basic unit is the *body segment*, which could be the torso, or any of the appendages. These segments are extracted and grouped together into individual humans using assumptions on human appearance and expected body articulation movements, as shown in Figure 2.2.

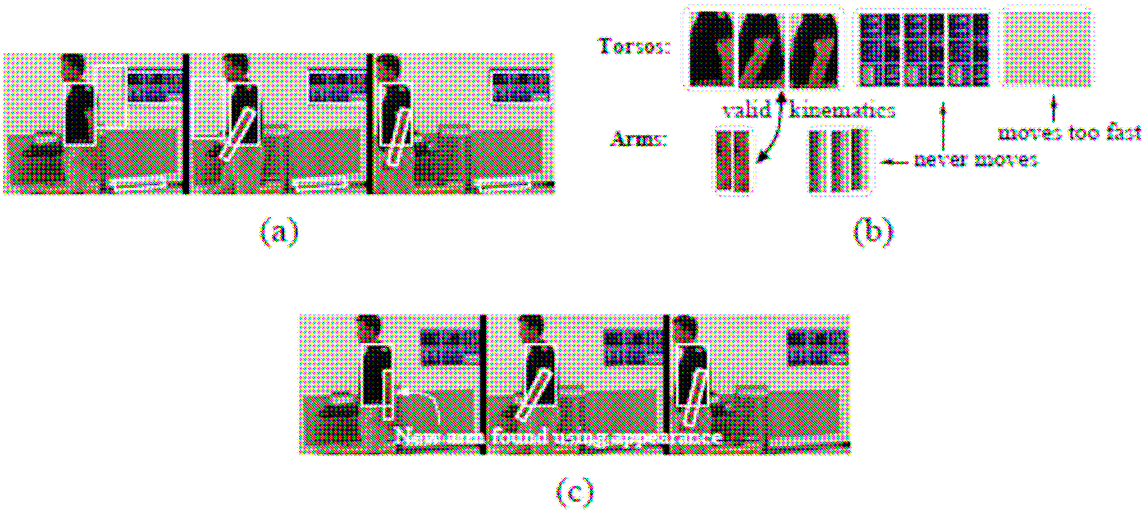


Figure 2.2: [RF-03] (a) Initial body segment extraction. (b) Articulation and movement assumptions and constraints. (c) Refining the human form based on results in step b.

The grouping of body segments can be done simultaneously using a single group of computations, this making the method suitable for the tracking of multiple humans. Segment grouping is done by first considering the color histograms of each segment, and performing a 30-dimensional modified mean shift based clustering step, which considers both the spatial and temporal dimensions. After this main grouping step, further refinements are done by discarding parts that do not follow expected human movement and articulation. Human anatomical constraints, e.g., a segment group must have only two arms, are also applied.

Since the segment clustering algorithm considers the temporal dimension during clustering, it has good performance against short bursts of occlusions, as shown in Figure 2.3. This is since the segment clustering block would presumably be able to be tolerant of a few disappearances of the segment in question, as long as the segment exists for a significant amount of time in the input sequence.

Though the study implied the capability to track multiple people, it focused on scenarios mainly for two people. While such experiments are essential to test the system's basic functionality, it is far from being a crowded scene. Unless further testing has been done since the publication of the work, the feasibility of the use of a 30-dimensional mean shift based clustering algorithm for scenes with crowds 10 or more people has yet to be verified.

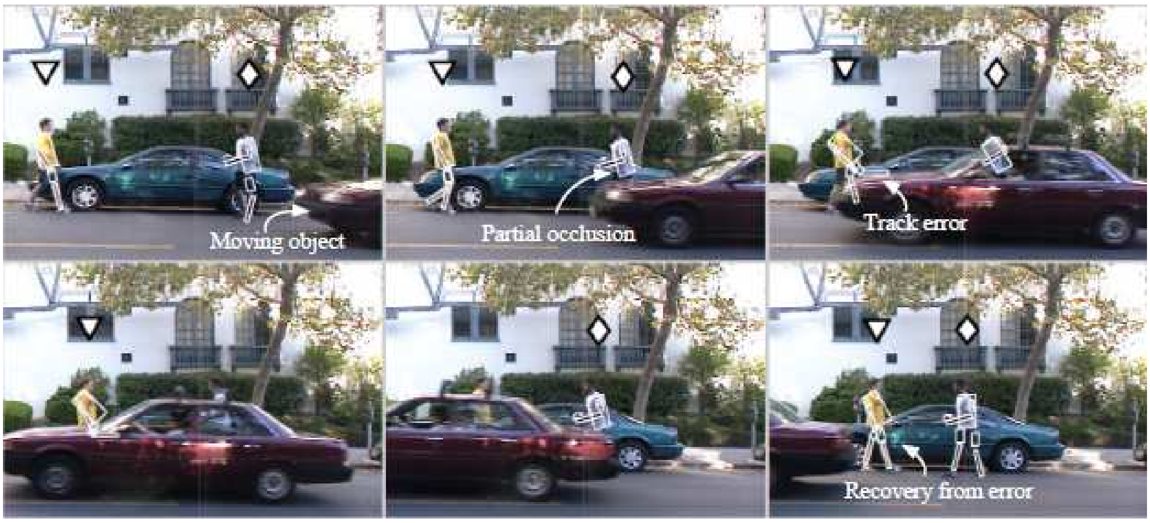
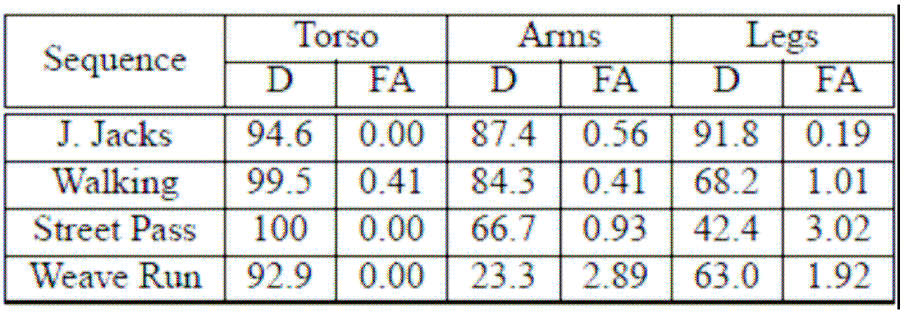


Figure 2.3: [RF-03] Tracking 2 people, with a demonstration of the system's resilience to occlusion.



Sequence	Torso		Arms		Legs	
	D	FA	D	FA	D	FA
J. Jacks	94.6	0.00	87.4	0.56	91.8	0.19
Walking	99.5	0.41	84.3	0.41	68.2	1.01
Street Pass	100	0.00	66.7	0.93	42.4	3.02
Weave Run	92.9	0.00	23.3	2.89	63.0	1.92

Table 2.1: [RF-03] Per-segment detection performance for different scenarios. "D" denotes detection rate, while "FA" denotes false alarm rate.

Furthermore, this scheme puts a premium on the form of the human body, and this is made evident in the need to evaluate segment-related detection rates in the paper, as in Table 2.1. Crowded scenes rarely depict human forms in their entirety; often an arm or a leg would be obscured from view, and it is not uncommon for human beings to be shown with only one arm visible, or with only half the torso and the head in clear view. Evidently, this would cripple any constraints that make assumptions on the composition of a human body, such as that used in their work.

Another notable work that uses a bottom-up approach is that of Zhao and Nevatia [ZN-CVPR04][ZN-PAMI04], which uses model fitting to detect and track the movements of humans in the scene, as depicted in its results in Figure 2.4.

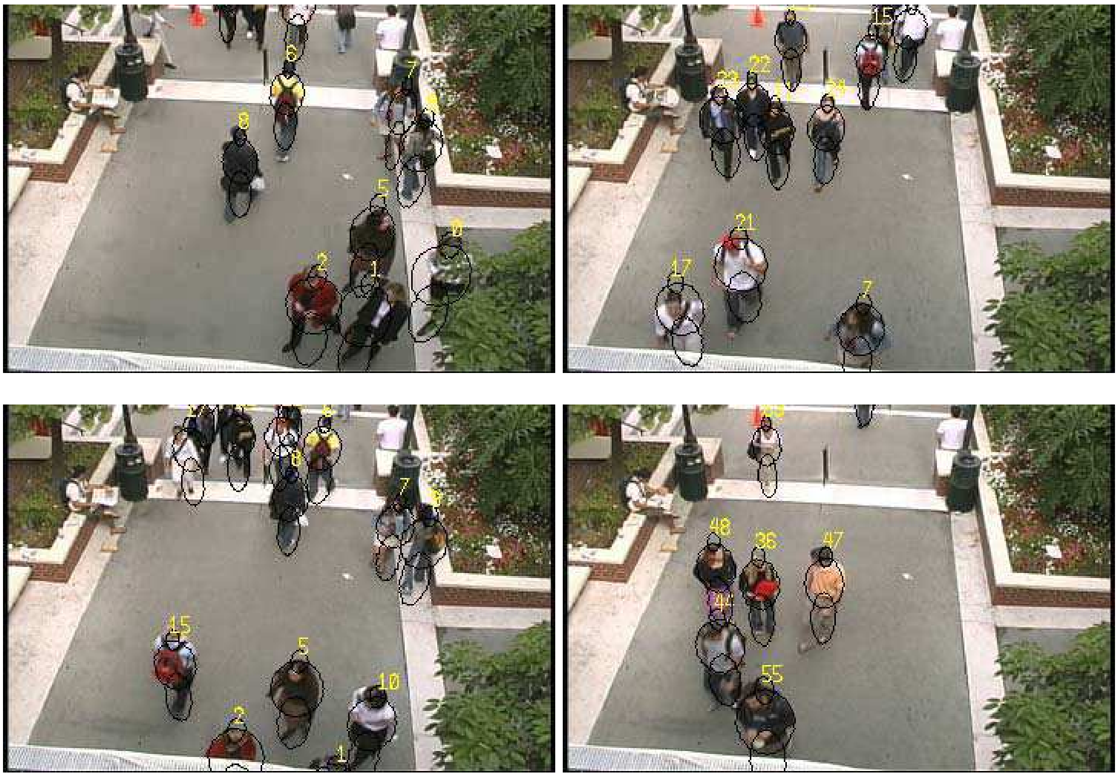


Figure 2.4: Results of [ZN-CVPR04][ZN-PAMI04], which indicate the use of human ellipsoid models.

Two constraints are evaluated according to a Bayesian problem-solving framework: First, an object (human) should exhibit pixels of a different color than the background. The background computation itself is based on a comprehensive statistical Gaussian background model that is based on the historical mean and covariance of each pixel's color. Second, an object must exhibit correspondence with its instances at different time slices. This is done using a 3D human ellipsoid model to find candidate humans, and a temporal correspondence model using both anatomical model and color-based information.

Again, the basic detection and tracking unit for this work is a human anatomical model, which could possibly fail in the event of excessive occlusion due to reasons previously discussed. Furthermore, background processing is used, and its results depend on statistical pixel data. As was mentioned in an earlier section, the applicability of statistical background models should not be assumed for crowds, since background pixels rarely reveal themselves because of foreground objects (people) occupying the scene for a significant amount of time.

Another disadvantage of using anatomical human body models 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, which would prove to be an impediment to performance when more and more people enter the scene. 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.

2.2. Feature Point Clustering: A Top-Down Approach

Recent years have seen the emergence of a new class of detection and tracking systems based on analysis of motion data. 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. These systems can be considered to take on a *top-down* approach to problem solving, since they avoid individual human models but instead tackle the problem by treating the crowd itself as the main unit for analysis. It is said to be top-down, since the analysis starts at the top level, i.e., the crowd as a singular mass, and then proceeds to break down the top level structure into component units using a set of rules, models, and assumptions that describe how the target objects are differentiated from each other. A scheme can be classified as top-down when it seeks to *separate and segment*, as opposed to bottom-up, whose nature is to *locate and build*.

The main inspirations for this work, and indeed this work itself, belong to this class of approaches. In Rabaud and Belongie's effort [RB-06], the crowded scene is represented as a collection of feature points linked to each other by a connectivity graph, and bound by a single motion-related constraint. The system performs well despite the relatively simple problem-solving premise, and sample experimental results are shown in Figure 2.5, depicting the feature points and the final clustering configurations.

The driving idea behind feature point clustering methods is the exploitation of motion data in order to locate target objects within a scene. For both of the works to be highlighted in this subsection, motion data is obtained by locating and tracing the trajectories of *feature points* as they are tracked in their movement across the scene. Feature points are defined as rudimentary trackable points in a single frame of a video sequence. In Rabaud and Belongie's work, feature points are selected using the Good Features to Track criterion as defined by Shi and Tomasi in [ST-94], which selects features within the image with enough texturedness to facilitate proper tracking. The tracking itself of the points is done using Lucas-Kanade optical flow [LK-81].

Humans are assumed to move as 3-dimensional rigid bodies. It is then said that if several features belong to the same target object and therefore share the same rigid motion in 3D, then the motion of their 2D orthographic projections is affine. Using this model, RANSAC clustering is performed to come up with feature point groupings, with each group corresponding to a single human being.

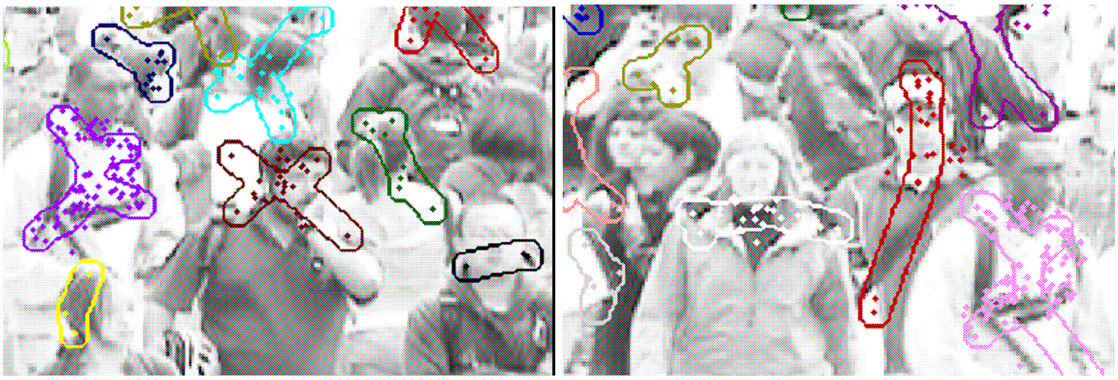


Figure 2.5: [RB-06] Feature points and clusters. Each cluster represents a single target human.

Satisfactory results have been observed during experiments for this system, with the main contributory factor for performance being the frame rate of the input video. This is understandably so since smoother frame rates allow for better optical flow tracking, and thus longer and better quality feature point trajectories. This study differentiates itself from previous crowd tracking works by freeing itself from usual assumptions on the number and form of the targets, as well as the nature of the scene. By freeing itself from traditional assumptions, most notably the assumption on the human form, it has effectively sidestepped the problems involved with them, most notably inter-human occlusion. For this work, as long as even a small fraction of a person's body is visible, he or she could successfully be recognized, as long as good features to track are found on them.

The work by Brostow and Cipolla [BC-06] is similar to Rabaud and Belongie's research in that it also uses a set of relatively simple and fundamental motion-related constraints. However, the general approach differs, as in this method, a two-stage clustering process is adopted, with each stage based on a separate assumption on human behavior and movement. The overall process is shown in Figure 2.6.

As with the previously mentioned work, Brostow and Cipolla's research starts with the extraction of feature point trajectories. This task is mainly done using the same Shi-Tomasi feature points tracked using Lucas-Kanade optical flow. Trajectories, as depicted in Figure 2.7, constitute the most basic computational unit for feature point clustering schemes.

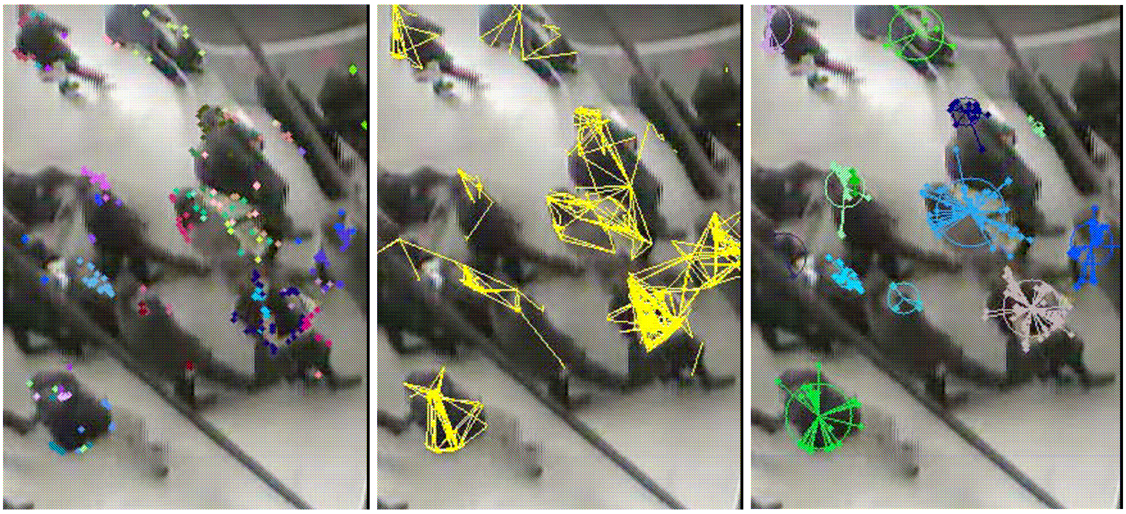


Figure 2.6: [BC-06] Different stages of Brostow and Cipolla's tracker. (left) Results of first clustering stage, with clusters denoted with different point colors. (middle) Intermediate computation based on coherence of motion. (right) Results of second clustering stage.

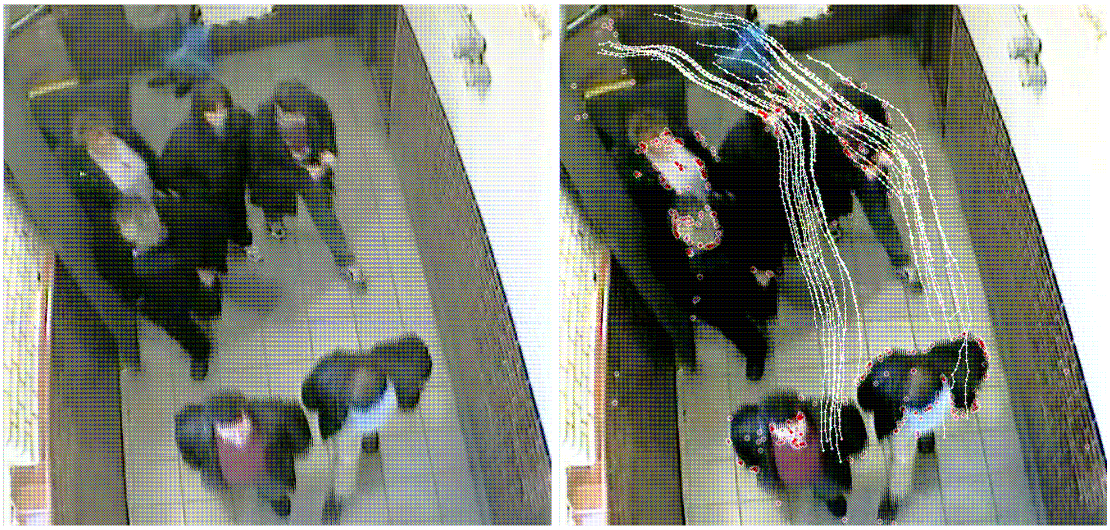


Figure 2.7: [BC-06] Feature points (red dots) and trajectories (white segments).

Brostow and Cipolla's work focuses on two main observations regarding human motion. First is that two points are likely to be part of the same moving body if they are in close spatial proximity of each other. An initial set of clusters is synthesized using spatial distance as a basis – each point is traced forwards and backwards in time, and the furthest historical distance between pairs of points is used as a metric to determine whether pairs of points are to be grouped together in the initial clustering.

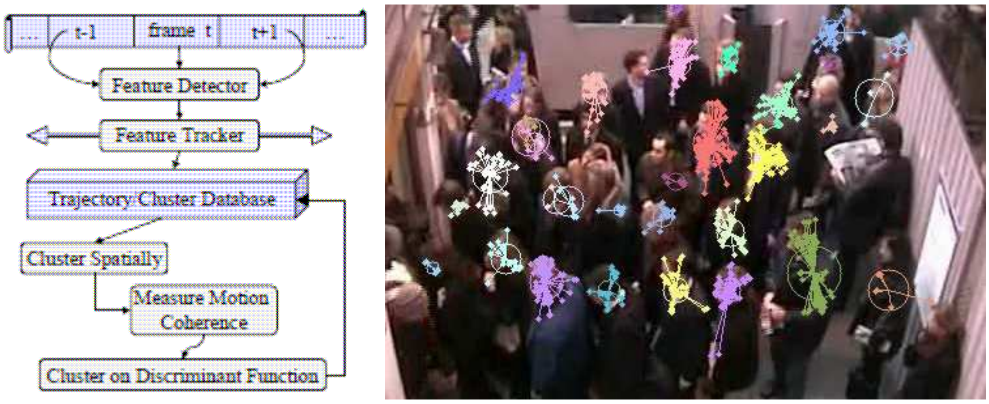


Figure 2.8: [BC-06] (left) Overall system block diagram, clearly showing the two stages of computation. (right) Results for a crowded scene at a train station.

The initial set of clusters is then subject to another grouping process. In this *clustering of clusters*, two clusters are merged together if they share the same coherent motion, which is determined by taking the variances of the spatial distances of the individual feature point trajectories that make up each cluster, and applying a Bayesian framework to decide whether two clusters should be merged based on the trajectory variance data.

The main drive for Brostow and Cipolla's research is to provide a proof-of-concept system that demonstrates the richness of information that can be extracted from motion data alone. It intentionally abandons all other assumptions and models and relies only on motion information to perform the feature point clustering. As with the previously mentioned work, this again effectively sidesteps the complications that may arise from using too detailed models, especially models that pertain to a human's physical form. Indeed, one may even go as far to say that using assumptions on the human form in crowded scenes should not be done, since people in crowds are rarely visible in their complete anatomy – arms, legs, or even full torsos are often occluded by other people.

2.3. Key Points for Crowd Tracking Research

This chapter will now be brought to a close with a brief discussion on the various points and issues that need to be considered when performing research on crowd tracking, as determined through investigation of the relevant research works discussed in earlier sections.

As already mentioned, inter-person occlusion for crowded scenes is a very common occurrence rather than an exception. A direct consequence of this is the failure of most human anatomical models, since occluded humans would lack visible body parts, and would be considered as incomplete as far as the model is concerned. This is also the case for models that deal with 2D geometric forms and silhouettes, since occluded humans generally do not follow a set geometric shape, nor do they even share common geometric properties.

A fix to this problem can be done by adding computational modules to boost the robustness of the human model, such as the addition of temporal consistency and compensation to solve the "disappearing body parts" problem, but again this fix only addresses occlusion if it occurs as a special exception rather than the norm. In any case, the addition of such modules for robustness and exception fixing begs the question of whether the correct approach is being used in the first place. For the task of tracking multiple people in general, such techniques may be suitable, especially when tracking simultaneous, complex gestures and motions of a few people. However, when faced with a crowded scene as in Figure 2.9, one should start taking care in selecting the method that is most suitable for solving the problem.

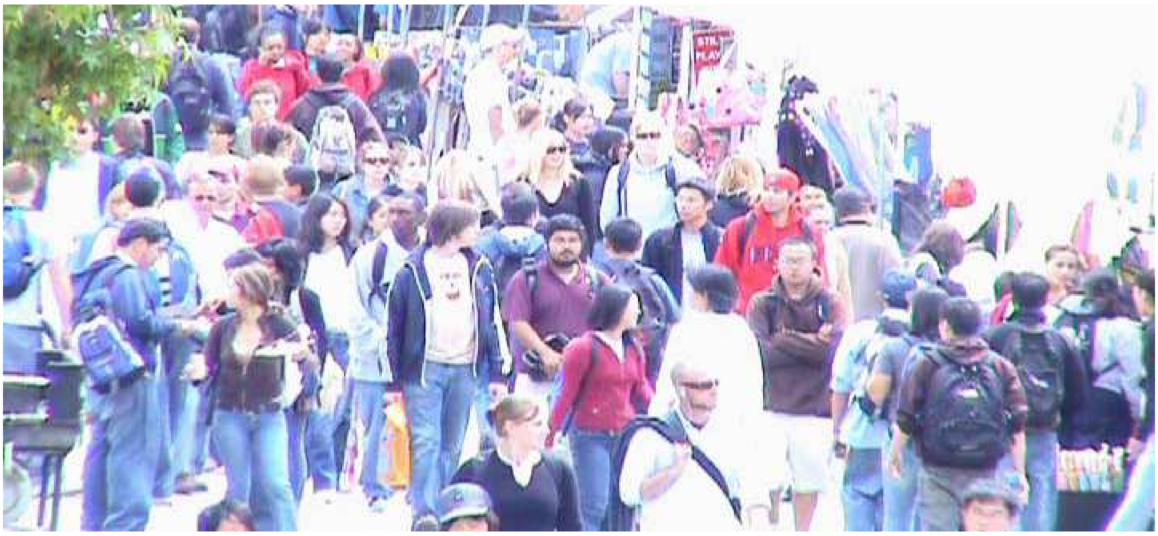


Figure 2.9: [RB-06] At this level of occlusion, it would be difficult to make any assumptions on the human form.

Recent works focusing on treating the crowd as a singular entity defined by a mesh of feature points effectively sidestep the occlusion problem by relinquishing any assumptions based on human anatomy. In these research efforts, humans could take on any visible form, and separating the feature points into clusters that represent individual people is done through an understanding of human motion. In the most basic terms, it is said that *humans in casual crowded scenes move differently*, and humans are therefore distinguished from each other by each individual's own unique motion.

By shifting the focus of computations from the human form to human motion, the constraint imposed by occlusion has been alleviated to some degree, from formerly requiring all body parts to be visible, to requiring at least a small fraction of the human body to be visible.

There are concerns regarding the performance of feature point clustering as well. Rabaud and Belongie highlighted the importance of the input video frame rate as a key concern in system performance. It has been observed that false negatives (untrackable objects) occur more frequently for input video sequences of poor frame rates. The reason for this is almost intuitive – low frame rates cause difficulty in any basic tracking algorithm, and optical flow, which is the basis for trajectory extraction, is no exception. Without trajectories of sufficient length available to the clustering mechanism, proper decisions cannot be made, and trajectories that are deemed too short may even be discarded for their uselessness.

Another important point for crowded scenes is the difficulty of extracting any useful background image for use by background models or background subtraction routines, since background pixels rarely are revealed in busy crowded scenes. While background subtraction may be useful in providing a mask that indicates the regions of interest for crowds, it will be of little use without a reasonably accurate background image or model. Three of the four major related works reviewed in this section admit to this and don't use any manner of background processing, often highlighting this as a feature of their respective systems because of the satisfactory performance despite the lack of any background models. This, however, again begs a question on the choice of technique. It can be argued that what is needed for the task is not necessarily an accurate depiction of the background nor a correct separation of background and foreground, but a means to extract a region of interest with enough information to be usable to the main stages of the tracking system. While the works reviewed here do not have such a provision and operate on the image frame in its entirety, our research utilizes an interest region extractor that aids in the selection of feature points and confine them only to regions with better likelihood of containing a target object. This method based on object detection, along with other novel concepts in our research, shall be discussed in the next chapter, which deals with our proposed method of solving the crowd-tracking problem.

Chapter 3

Proposed Technique

Our system adopts a general methodology of feature point tracking and clustering. Like previous related works, we utilize spatial and motion cues such as bounding box constraints, spatial proximity and motion trajectory analysis. However, we also differentiate our study from others through the introduction and integration of a few novel ideas that we believe would be able to provide enhanced detection and tracking performance.

Our choice of methodology with which to perform the task at hand is largely due to its advantages when faced with a crowd-tracking problem. We select feature point tracking and clustering as our method of choice because:

- **It is resilient against occlusions**

  Occlusion-related problems arise when there are preconceived expectations with regards to a target object's form. With feature point clustering methods, assumptions based on form are relinquished in favor of assumptions based on motion. It can be considered as part of feature point clustering methods' general methodology and principle of operation to include as little computations that are based on geometric forms as possible. Without shape-related assumptions, target objects are allowed to be freeform, free from the occlusion problem, and bound only by the rules of motion.

- **It has elegance in simplicity**

  The mention of simplicity here does not necessarily pertain to the simplicity of program code and functional blocks of the tracking system. Rather, it pertains to the simplicity of the most basic and central concept behind feature point clustering – to group points that move together and consider them as a representative of a single target object. The primary advantage of having such a fundamentally simple main idea lies in its extensibility. Aside from being a concept that is easy to understand, this wide-encompassing concept of motion is open to different interpretations, thus leaving the researcher with many creative ways of coming up with a specific methodology and implementation.

As was discussed in the previous chapter, current works on feature point clustering base their results only on spatial proximity and coherence of motion. With feature point clustering being a relatively new approach to the tracking problem in itself, present works focus on being proof-of-concept research that focus only on a handful of fundamental concepts. In [BC-06], the authors explicitly mention that the purpose of their research is to show that the task can be done using only coherence of motion as the main metric for clustering. The authors of the same work also suggest for future work the use of multiple models, metrics and methods instead of just one, and at the same time express their confidence on the success of such systems.

In our work, we introduce a total of three novel concepts applicable to the main concept of feature point clustering. The main contributions of our study are:

- **The use of frequency domain cues in motion analysis**

  We introduce in our work a new class of cues based on frequency domain analysis, which we believe will be able to provide better clustering and segmentation performance than performing evaluations with motion-based data alone. For human beings, this is especially true since in casual scenarios, the periodic motion in walk cycles differ from person to person, even if they are walking in the same general direction. For example, two people walking together may be easily mistaken as one entity by virtue of motion coherence alone, but may be successfully separated by applying frequency analysis on their individual walk cycles.

- **The use of object detection as an alternative to background subtraction**

  In the feature point selection process, background subtraction may prove to be valuable in distinguishing usable feature points that lie on foreground objects from useless feature points found in the background. Unfortunately for crowded scenes, it is difficult to come up with a usable background image, as discussed in the previous chapter. A small paradigm shift was in order with regards to this, as it should be realized that for our purposes, background subtraction is not needed *per se*; it just so happens to be *one of the many methods* that could be utilized to realize the actual underlying goal of distinguishing background regions from foreground regions for feature point selection. Instead of looking for background pixels using background subtraction to come up with a background mask, we look for foreground objects using object detection to come up with a foreground mask. For our purposes, we use the Cascade of Boosted Haar Classifiers as documented by Viola and Jones in [VJ-01]. It should be noted that our contribution here is not the object detection system itself, but the unique use of it as a tool to address our needs.

- **A clustering scheme sensitive to both edge weights and graph topology**

  For our research, we use an edge-weighted graph to represent the crowd, and a simple yet powerful clustering scheme to handle the separation of target objects. Our hybrid clustering scheme, which directly utilizes edge weight data as well as a graph property called betweenness centrality, is sensitive both to edge weight values and the topological layout of the input graph. This scheme is especially powerful in the removal of bridge-like edges that are the indicators of target objects with clusters in need of separation.

3.1. A Feature Point Clustering Based Approach

Our system uses weighted graphs as the main abstract representation of humans in a clip of motion video. For every frame, we define two graphs, $\boldsymbol{G_I = \{V,E_I\}}$ and $\boldsymbol{G_F = \{V,E_F\}}$, as the initial hypothesis and final configuration that describe the distribution of humans. In this notation, $\boldsymbol{V}$ represents the set of labeled vertices that correspond to and contain information on each tracked feature point. The set of weighted edges $\boldsymbol{E_I}$, 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 $\boldsymbol{E_F}$ represents the non-weighted edges defining the final configuration.

To elaborate, since the weight of an edge $\boldsymbol{e_{i,pq}}$ from set $\boldsymbol{E_I}$ that joins vertices $\boldsymbol{p}$ and $\boldsymbol{q}$ is a number that would define how likely $\boldsymbol{p}$ and $\boldsymbol{q}$ are points on the same person, it is considered as the central piece of information that is the basis of the clustering process. Intuitively, the clustering process should favor pairs of feature points with high likelihoods of belonging to the same person, and sever the links between those with low edge weights, and thus, low likelihoods. The resulting edge set $\boldsymbol{E_F}$ would then be the set of edges that were retained through the clustering process – presumably those with high weights. The existence of an edge $\boldsymbol{e_{f,pq}}$ from set $\boldsymbol{E_F}$ would indicate that the clustering process has decided or confirmed that the feature points that correspond to vertices $\boldsymbol{p}$ and $\boldsymbol{q}$ indeed belong to the same target human.

The computations and process behind the overall clustering function $\boldsymbol{C}$ that relates $\boldsymbol{G_I}$ to $\boldsymbol{G_F}$ shall be defined in later sections. At this point, it is enough to consider that each connected component in the result $\boldsymbol{G_F}$ ideally corresponds to a single instance of a human being for the video frame upon which the computations were carried out.

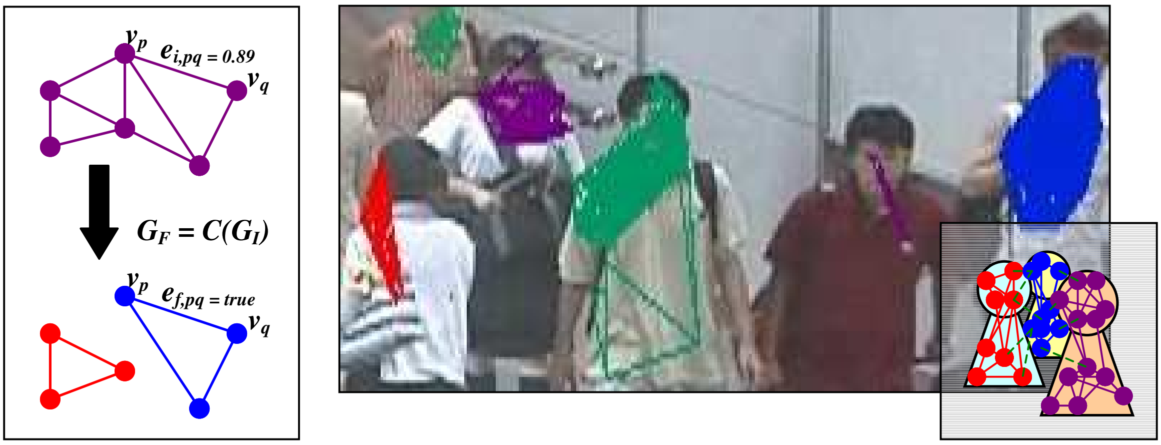


*Figure 3.1: (left) An overview of the clustering process. (right) Sample output with connected components from **G_F** drawn over the source video. (inset) Figurative illustration of connected components.*

3.2. Three Stage System Architecture

We implement our feature point clustering ideas using a system with three distinct functional stages, as illustrated below.

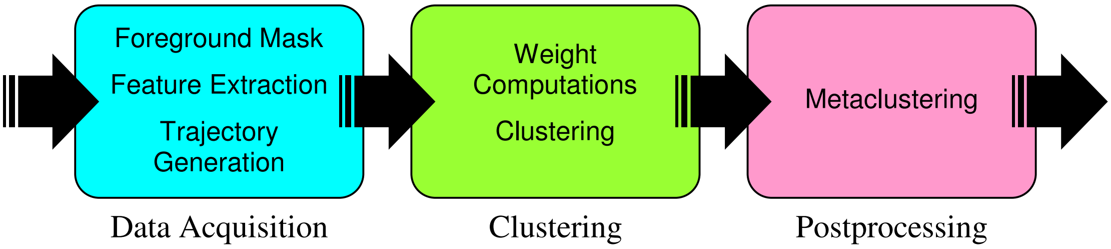


Figure 3.2: Our system's three functional stages.

The definitions of the main stages are fairly straightforward. The first stage centers on the acquisition of data for use by the weighting and clustering modules. It focuses on feature point detection, generation of trajectories using optical flow, and the storage of trajectories in a special data structure. The second stage, which is the main focus of our work, involves constructing the input graph, assigning weights, and clustering. The third and final stage contains any postprocessing routines that would further refine the preceding stage's clustering results.

3.3. Data Acquisition

The data acquisition stage focuses primarily on extracting raw data from the input motion video, and storing the information in especially designed data structures. It should be noted that neither calculation nor interpretation takes place in this stage.

3.3.1. Feature Point Extraction

Feature points are the most rudimentary units in our system, and are essential in constructing the initial hypothesis graph $\boldsymbol{G_I}$. Each element $\boldsymbol{v_i}$ of the vertex set $\boldsymbol{V}$ corresponds to a feature point and its relevant parameters, notably, but not limited to, its spatial x and y axis locations.

Feature points in this work are defined as synonymous to Tomasi-Kanade features, and are obtained using the technique described in [ST-94]. The importance of selecting sufficiently textured feature points for satisfactory tracking performance is especially appreciated by research works such as ours that put a premium on tracked trajectory length, and consider trajectory length as one important measure of information quality.

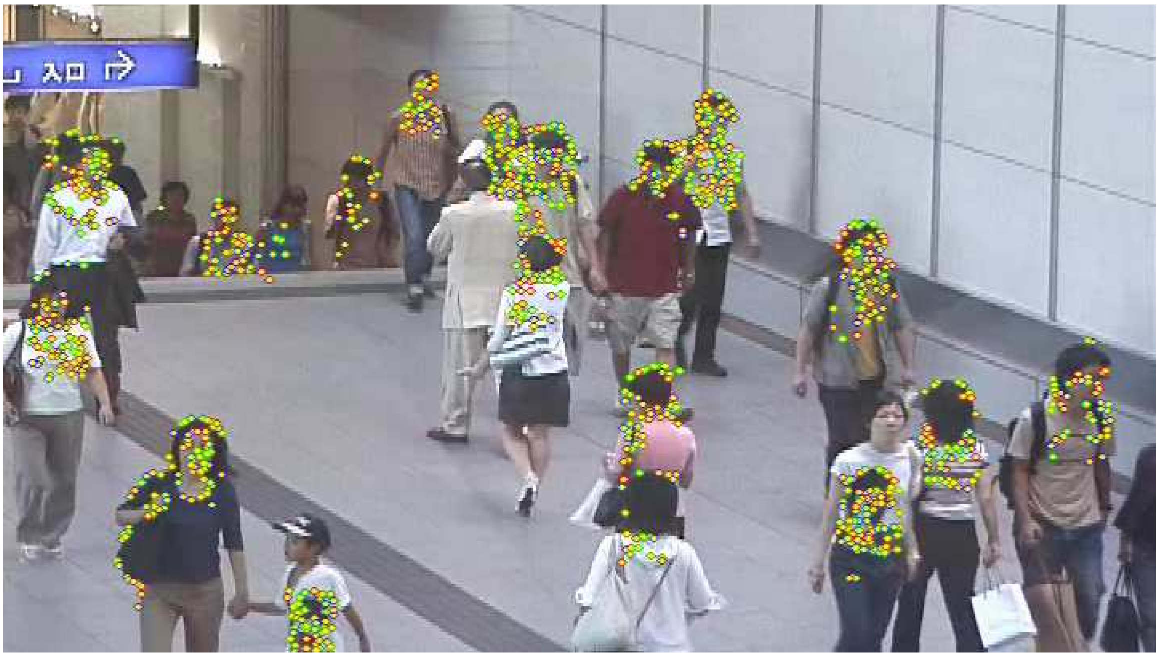


Figure 3.3: "Good Features to Track" [ST-94] – features with high texture and contrast properties – are favored due to their good tracking performance when subject to optical flow.

3.3.2. Trajectory Generation and Storage

Tracking immediately follows feature point detection, using a standard implementation of Lucas-Kanade optical flow [LK-81]. Each feature point's trajectory is then stored using an especially designed data structure that enables the storage and retrieval of the position of each feature point in every frame in time that it has existed.

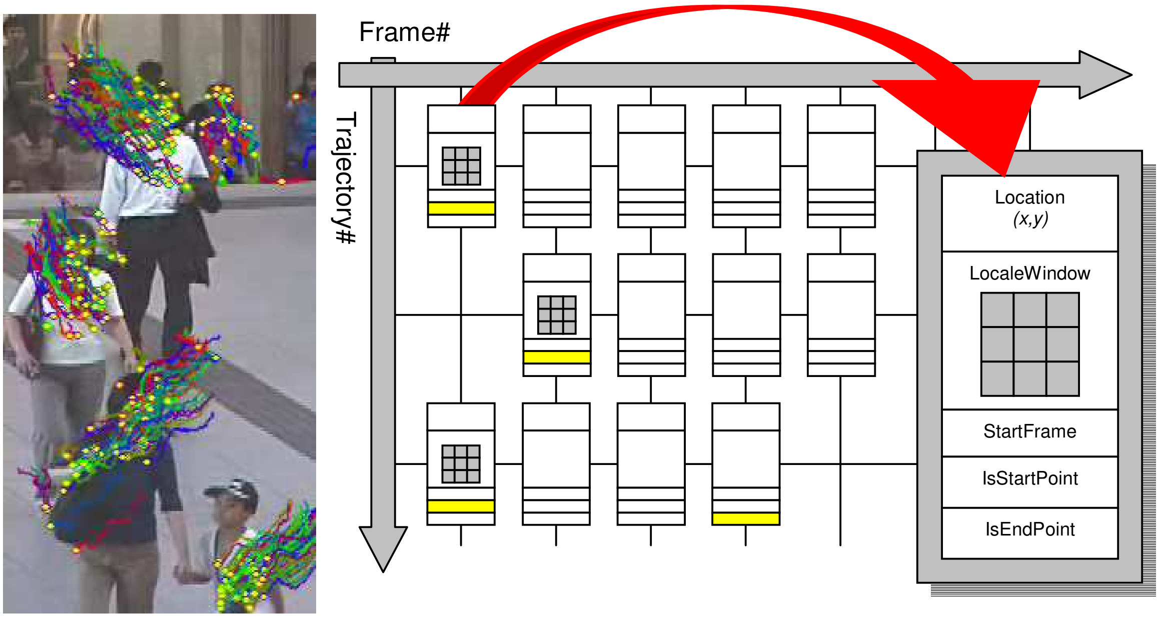


Figure 3.4: (left) Feature point trajectories drawn over input video. (right) Data structure implemented using linked classes in C++. (inset) Details of a single feature point storage unit, containing position data and several navigation tags.

It should be noted that despite the satisfactory level of performance of Lucas-Kanade optical flow, tracking failures still occur and is manifested visually by feature points getting transferred from one moving target to another. To avoid this, a consistency check based on template matching is done for each feature point, in which a small image window around the feature point in question for the current frame is compared to the image window around the feature point at the starting point of its trajectory; i.e., the frame in which the feature point has first appeared. For this reason, as shown in Figure 3.4, image templates for each starting point are obtained and stored for each trajectory. Feature points whose surrounding pixels differ significantly from those of its starting state are assumed to have been transferred to a different object and are duly deleted, thus stopping the corresponding trajectory from further advancement.

3.3.3. 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. The rationale behind selecting object detection as a viable substitute for background subtraction has already been discussed at the beginning of this chapter, and sample results can be seen below.

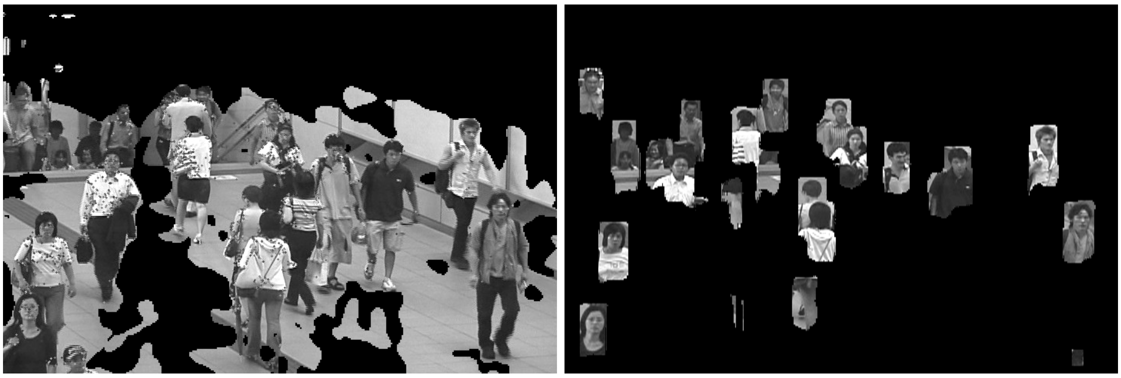


Figure 3.5: (left) Foreground/background masking using background subtraction results. (right) Foreground/background masking using object detection results.

As can be observed in Figure 3.5, while object detection results would not be able stay true to the contours of the foreground objects, it nevertheless does a better job at excluding background regions. For our system, we design the object detector such that only regions of the upper body are captured. We use an implementation of the Viola-Jones hierarchical cascade of boosted Haar-like classifiers as described in [VJ-01] and [VJ-04], which in turn uses Freund and Schapire’s Adaptive Boosting framework first introduced in [FS-97]. A cascaded classifier trained for human heads from all views was synthesized, and the detector output is interpreted as a mask image. This result is dilated to include a region of pixels below every detected instance of a human head; i.e., the pixels occupying the corresponding person’s torso.

The reason selecting this region of interest is that it is this part of the human body that would move in such a way that most resembles rigid motion. Including the arms and legs would only result in unnecessary articulations that may affect clustering results. This is perhaps an additional benefit of using object detection; that one could explicitly include and exclude parts of the foreground objects according to the requirements of the research.

3.4. Clustering

The main idea for clustering has already been discussed in a previous section, in which the initial hypothesis graph $\boldsymbol{G_I = \{V,E_I\}}$ is clustered using our clustering routine $\boldsymbol{C}$ to obtain the final result graph $\boldsymbol{G_F = \{V,E_F\}}$ that defines the distribution of humans within the scene. Before performing the clustering operation, we initialize the edge weights $\boldsymbol{e_{i,pq}}$ for all vertex pairs $\boldsymbol{\{p,q\}}$ such that each weight represents the likelihood of $\boldsymbol{p}$ and $\boldsymbol{q}$ belonging to the same object.

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

$$e_{i,pq} = s_{pq,space} \cdot s_{pq,motion} \cdot s_{pq,fourier} \tag{3.1}$$

As stated by the equation, we base the edge weight on three components based on three characteristics of human motion, namely, spatial distance, coherence of motion, and frequency domain properties. For both sets $\boldsymbol{E_I}$ and $\boldsymbol{E_F}$, a nonzero value for $\boldsymbol{e_{pq}}$ denotes the existence of the edge, while zero would denote its nonexistence. For $\boldsymbol{E_I}$, each component score and edge weight is real-valued from 0 to 1.

3.4.1. Spatial Analysis

The spatial score $\boldsymbol{s_{pq,space}}$ is implemented as an initialization for the hypothesis graph $\boldsymbol{G_I}$. It is based on the observation that the likelihood of two points to belong to the same object increases as the distance between them decreases. Previous works such as [RB-06] have used a somewhat similar measure called the bounding box, but our implementation bears a few notable differences from the simple bounding box constraint. The spatial score is hereby defined as follows:

$$s_{pq,space} = M_F\left[\sqrt{(\Delta x)^2 + (\Delta y \cdot \rho)^2}, W\right] \tag{3.2}$$

In this equation, $\Delta \boldsymbol{x}$ and $\Delta \boldsymbol{y}$ are the distances between points $\boldsymbol{p}$ and $\boldsymbol{q}$ along the horizontal and vertical axes, respectively. We define ρ as the aspect ratio of the bounding box describing the expected dimensions of humans, though our definition slightly deviates from the usual definition. Aspect ratio is usually defined to be the ratio of the longer dimension to the shorter dimension. In our work, we instead fix the numerator and denominator in the computation to specific axes, as follows:

$$\rho = \frac{W}{H} \tag{3.3}$$

This simply states that the aspect ratio ρ is the ratio of the bounding box width $\boldsymbol{W}$ to the bounding box height $\boldsymbol{H}$. Equations 3.2 and 3.3 essentially suggest a normalization operation on the horizontal and vertical components of the distance between the two points when the bounding box is non-square.

The mapping function $\boldsymbol{M_F}$ needs elaboration as well. It is a function that assigns a score ranging from 0 to 1, to some given raw data. In this case, the raw data is the distance between the two points in question. It accepts two parameters: the raw positive value $\boldsymbol{n}$ to be evaluated, and its maximum value $\boldsymbol{n_{max}}$. We define two types of mapping functions, based on the sine squared and cosine squared functions, as in the figures below.

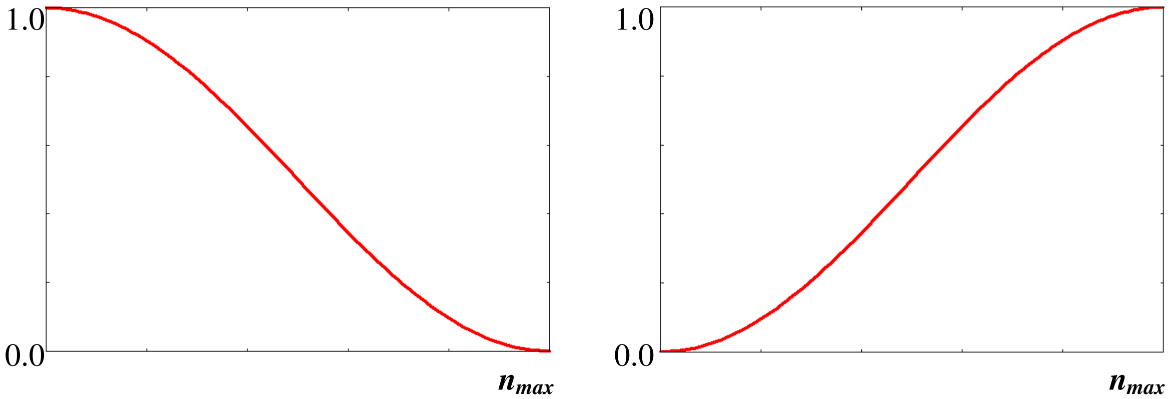


Figure 3.6: (left) Falling mapping function, $\boldsymbol{M_F(n,n_{max})}$. (right) Rising mapping function, $\boldsymbol{M_R(n,n_{max})}$.

In equation form, the mapping functions are defined as follows:

$$M_F(n, n_{\max}) = \begin{cases} 1 & , n < 0 \\ \cos^2\left(\dfrac{n}{n_{\max}} \cdot \dfrac{\pi}{2}\right) & , 0 \le n \le n_{\max} \\ 0 & , n > n_{\max} \end{cases} \tag{3.4}$$

$$M_R(n, n_{\max}) = \begin{cases} 0 & , n < 0 \\ \sin^2\left(\dfrac{n}{n_{\max}} \cdot \dfrac{\pi}{2}\right) & , 0 \le n \le n_{\max} \\ 1 & , n > n_{\max} \end{cases} \tag{3.5}$$

These mapping functions are ones that were selected to address our need to assign standard score values to raw data. The squared sine and squared cosine functions were specifically selected because first, they provide a smooth transition between 0 and 1, and second, even in the piecewise variations above, they will still exhibit continuity at 0 and $\boldsymbol{n_{max}}$. The input to these mapping functions would be values such as spatial distance, as discussed in this section, as well as values such as frequency and trajectory standard deviation, which will be discussed in the next sections. The output, on the other hand, would be a score ranging from 0 to 1. Finally, the maximum value $\boldsymbol{n_{max}}$ for each value $\boldsymbol{n}$ is defined such that any value outside of the specified range would be mapped to either 0 or 1, as written in Equations 3.4 and 3.5.

3.4.2. 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.

Coherence of Motion

We measure coherence of motion by utilizing the set of feature point trajectories that were collected during the data acquisition stage (section 3.3.2). We take the standard deviation σ_{pq} in pixels of the distance $\boldsymbol{l_{pq}}$ between the two feature points in question as they move across the scene (Equation 3.6), during the time window from $\boldsymbol{t_i}$ to $\boldsymbol{t_f}$ in which they have coexisted. This is a direct measure of the points' deviation from the previously mentioned ideal description of constant separation.

$$\sigma_{pq} = \sqrt{\frac{\sum_{t=t_i}^{t_f} l_{pq}^{\ 2}}{t_f - t_i} - \left(\frac{\sum_{t=t_i}^{t_f} l_{pq}}{t_f - t_i}\right)^2} \qquad (3.6)$$

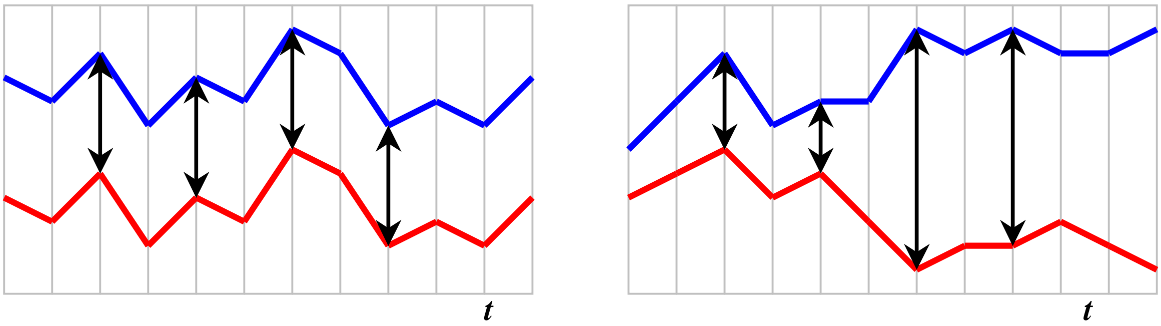


Figure 3.7: (left) Constant distances imply high likelihood of belonging to the same object. (right) High deviation implies different objects.

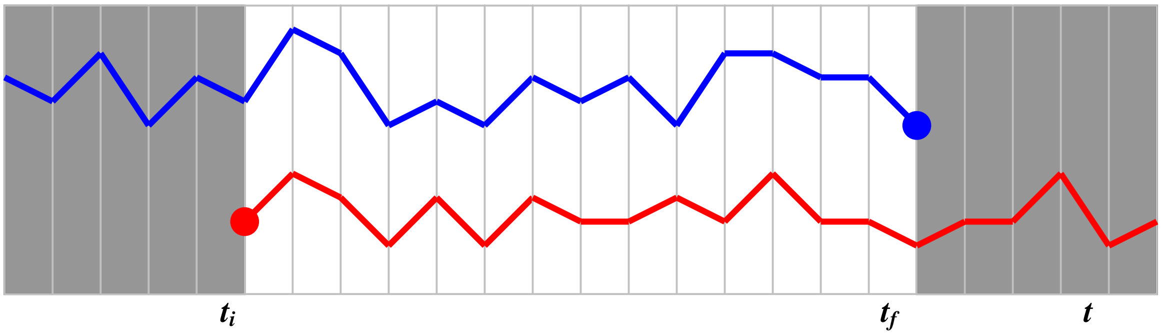


Figure 3.8: The standard deviation is computed only for the time window in which the two feature points have coexisted (middle region).

The standard deviation σ_{pq} is then assigned to a likelihood score $\boldsymbol{s_{pq,coherence}}$ using a falling mapping function with an empirically-selected maximum value σ_{max}, as in Equation 3.7.

$$S_{pq,coherence} = M_F(\sigma_{pq}, \sigma_{\max}) \tag{3.7}$$

Trajectory Coexistence

Unlike related works that artificially extend the length of trajectories through extrapolation techniques [RB-06][BC-06], 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. Furthermore, we would want to operate purely on actual data that was extracted from the input scene, rather than on data with artificially created entries, even if it was through extrapolation.

However, this would imply that trajectories would have varying coexistence intervals $\boldsymbol{t_f}$-$\boldsymbol{t_i}$. Furthermore, smaller values for $\boldsymbol{t_f}$-$\boldsymbol{t_i}$ 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 $\boldsymbol{t_f}$-$\boldsymbol{t_i}$ as a weighting metric. As with the coherent motion score $\boldsymbol{s_{pq,coherence}}$, we assign the coexistence interval $\boldsymbol{t_f}$-$\boldsymbol{t_i}$ to a likelihood score $\boldsymbol{s_{pq,coexistence}}$ using a rising mapping function as in Equation 3.8, such that larger intervals $\boldsymbol{t_f}$-$\boldsymbol{t_i}$ would result in higher scores.

$$S_{pq,coexistence} = M_R(t_f - t_i, \Delta t_{\max}) \tag{3.8}$$

Overall Motion Analysis Score

Finally, the value of $\boldsymbol{s_{pq,motion}}$ is then obtained as a simple product of the two component scores.

$$S_{pq,motion} = S_{pq,coexistence} \cdot S_{pq,coexistence} \tag{3.9}$$

3.4.3. Frequency Domain 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 (e.g., marching bands), humans generally exhibit different frequencies and phase during their walk cycles. The easiest way to observe this phenomenon is to observe a group of people walking together, and to see their heads and bodies bob up and down independently and at different phase angles and frequencies.

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. Using motion analysis alone would manifest the differences as nonzero but negligible standard deviation values, thus keeping the score at a high value despite the differences in movement. Frequency analysis, on the other hand, would be able to reveal any significant differences in the frequencies of their individual and unique walk cycles, and express them in terms of substantially penalized scores.

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

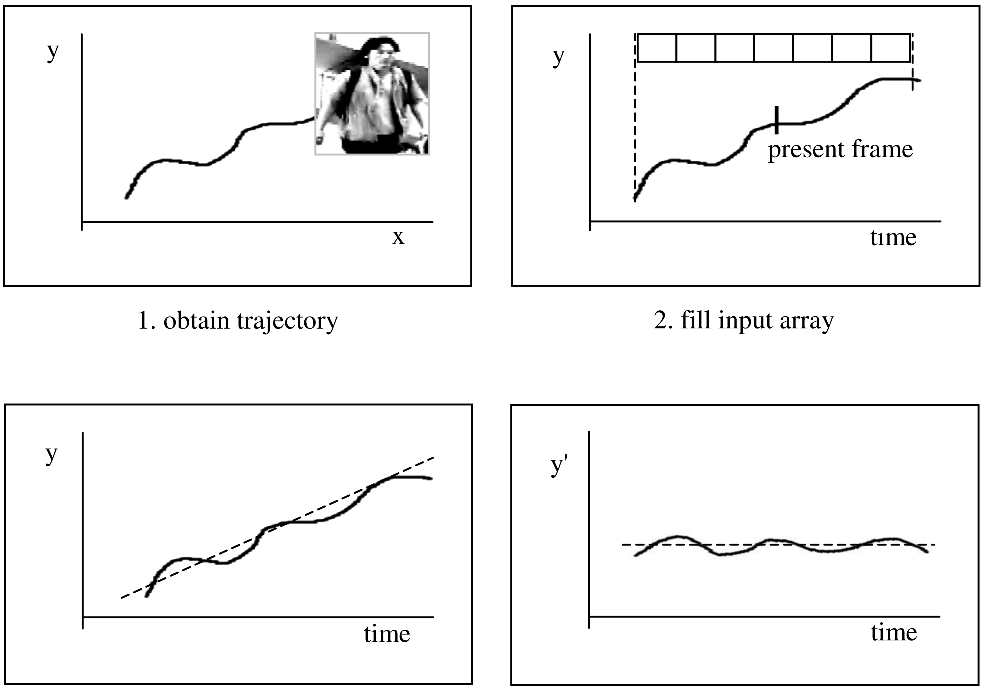
$$S_{pq,fourier} = S_{pq,frequency} \cdot S_{pq,phase}{}^{1} \tag{3.10}$$

Before the computation of the two component scores, each feature point's trajectory data is processed for frequency analysis. Input arrays for Fourier transform computation are filled with each feature point's vertical axis position 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.

[1] *A note on the notation: Since the frequency domain analysis score is composed of a frequency component and a phase component, we refrain from the use of the word "frequency" to refer to the overall frequency analysis score. We use the word, "Fourier" instead.*

We consider only the vertical axis in our computations since for a video depicting ordinary human movement with an upright camera, oscillations due to walking, i.e., the "bobbing of heads", would only be along the Y axis. If there would be oscillations along the horizontal axis, they will not be due to walking. Furthermore, periodic motion may not appear along the X axis at all, as in the case of a person walking directly towards the camera. Since the person would stay at the same horizontal axis location during the duration of his or her movement, the Fourier transform of the X axis position as a function of time would report zero frequency content. Walking oscillations manifest mostly along the vertical axis. On the other hand, the horizontal axis may or may not exhibit periodic motion, and should thus be considered as unreliable with regards to frequency analysis.

Figure 3.9 below illustrates the process for initializing the input arrays for Fourier analysis. We use simple regression line calculations to remove any biases related to the person's large-scale movement, and leave only the periodic component for transformation.



1. obtain trajectory

2. fill input array

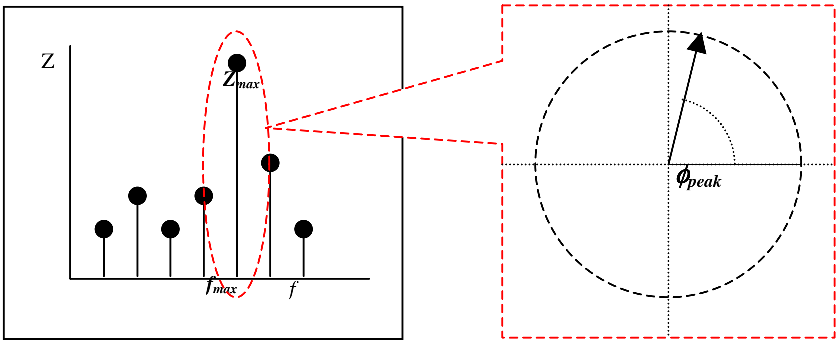
3. obtain regression line

4. remove regression line bias

Figure 3.9: Initializing the Fast Fourier transform input array with trajectory data

An FFT (Fast Fourier Transform) input 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 for an input of 60 frames per second, or about 1.5 to 2 gait oscillations for casual and relaxed walking.

The complex Fourier transform of each feature point's Y coordinate as a function of time is then computed. For each transform result, two values are obtained: the location $\boldsymbol{f_{peak}}$ of the peak complex magnitude $\boldsymbol{Z_{max}}$, and the complex phase at the peak location, denoted by $\boldsymbol{\phi_{peak}}$.



1. obtain transform peak

2. obtain phase

Figure 3.10: Illustration of the frequency analysis process

Scores $\boldsymbol{s_{pq,frequency}}$ and $\boldsymbol{s_{pq,phase}}$ are then computed based on the differences $\boldsymbol{\Delta f_{peak}}$ and $\boldsymbol{\Delta\phi_{peak}}$ of the frequency and phase values respectively for each pair of points. We then apply a standard mapping function such that low differences between frequencies and phases would result in a high score, and use Equation 3.10 to obtain the final frequency analysis score.

$$S_{pq,frequency} = M_F\left(\Delta f, \Delta f_{\max}\right) \qquad (3.11)$$

$$S_{pq,phase} = M_F\left(\Delta\phi, \Delta\phi_{\max}\right) \qquad (3.12)$$

3.4.4. Betweenness Centrality Based Clustering

The betweenness centrality [Br-01] of a vertex or edge of a graph is generally considered and used as a numerical representation of the vertex or edge's performance and importance in the context of flow and connection. The edge betweenness centrality of an edge ***p*** in a graph ***G={V,E}*** with vertex set ***V*** and edge set ***E*** is defined by following equation:

$$c_B(p) = \sum_{s \in V, t \in V} \frac{n_{st}(p)}{n_{st}} \tag{3.13}$$

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***. This makes it a modified counter of shortest paths. In fact, if there is only one shortest path between any two vertices ***s*** and ***t*** for all ***s*** and ***t*** in ***G***, as is usually the case for weighted graphs with real-valued weights, ***c$_B$(p)*** indeed becomes a simple counter of shortest paths that pass through ***p***.

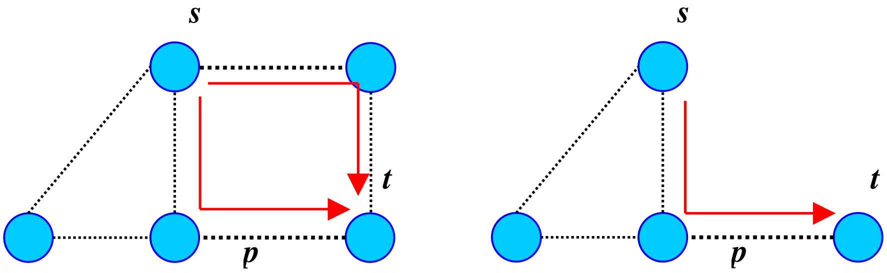


Figure 3.11: A simple example of betweenness centrality computation (discussed below).

Figure 3.11 depicts a single term in the summation computation for betweenness centrality. At the left, two shortest paths exist between vertices ***s*** and ***t***. One of them passes through edge ***p***, so ***[n$_{st}$(p)/n$_{st}$]***, which is the contribution of the vertex pair ***{s,t}*** in the summation, is 0.5. The right hand diagram has only one shortest path between vertices ***s*** and ***t***, with it passing through ***p***. This would give ***[n$_{st}$(p)/n$_{st}$]*** a value of 1.0.

Betweenness centrality is a tool that is used to quantify the performance or importance of a specific graph element. One application is in sociology, where it is used to determine crucial social roles and figures. Another application is in computer networking, where it can be used to identify critical network links. We use betweenness centrality in our work for its capability of identifying two types of edges: *weak links* and *bridge edges*. Weak links are edges with low weights, and the reason for interest in such edges is already obvious. Bridge edges, however, require further explanation.

Bridge edges are manifestations of non-idealities in our basic assumptions on human motion. One of our assumptions is that feature points on humans undergo rigid motion. It is for this reason that we have tuned our feature selection mask to cover only the head and torso regions, thus excluding the limbs, which undergo much articulation. However, this is merely an ideal description, and indeed, humans turn their heads, shift their clothing, and twist their torsos in casual scenes, albeit only to some limited degree. We compensate for these non-idealities by applying a semblance of tolerance via smooth mapping curves instead of hard thresholds. However, despite this, stray edges with fairly high weights still get created from time to time, as in the example below in Figure 3.12.

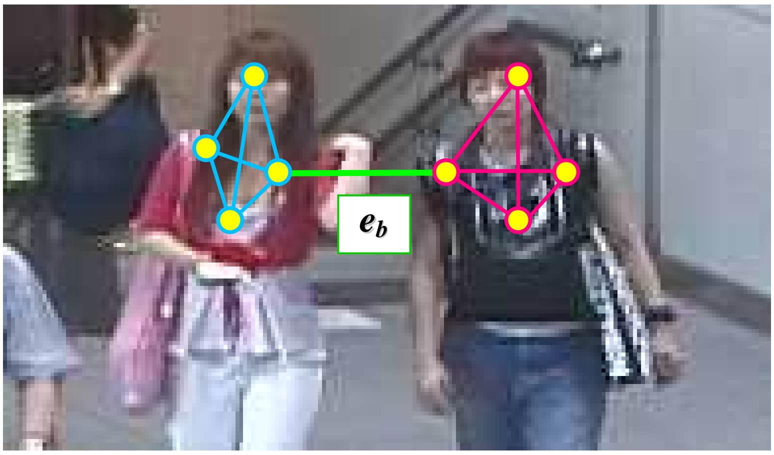


Figure 3.12: Example of a bridge edge manifestation (green edge).

In this example, consider the right side person to have momentarily moved her right shoulder in a manner that would conform to the movement of the other feature points on her body, but with just enough articulation for it to also coincide with one of the left hand person's feature points. At that exact moment of coincidence, edge $\boldsymbol{e_b}$ would be formed, and it would be one with a sufficiently high figure as its weight. Since such movements are exception rather than norm, only a handful of bridge edges of this nature would ever get formed. However, because of their high weights, they may escape the scrutiny of some clustering routines, such as that of a simple edge weight threshold.

Betweenness centrality, for its simplicity in concept, is able to identify bridge edges in addition to weak links. Figure 3.13 illustrates this through example.

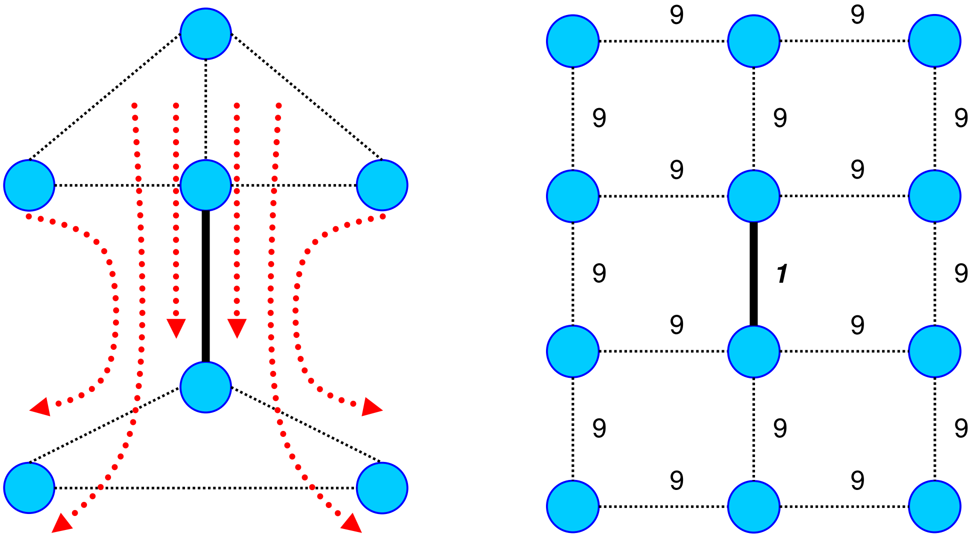


Figure 3.13: Bridges and low-weighted edges have high betweenness centrality.

In this set of illustrations, the thick lines denote edges with high betweenness centrality. On the left hand side, which depicts an unweighted graph, it can be seen that the edge drawn with the bold line serves as the only conduit between the upper and lower parts of the graph. Thus, the shortest paths that connect the points in the upper half to points in the lower half all pass through the middle edge, thus giving it a high betweenness centrality.

In graph theory, the formal definition of a bridge edge is one that would disconnect the graph upon deletion. In the figure above, removing the middle edge would disconnect the graph into two connected components.

The right hand side of Figure 3.13 depicts a weighted graph with the shaded edge having a high betweenness centrality due to its low weight. For weighted graphs, a shortest path between a pair of vertices ***{s,t}*** is defined as one whose total edge weights are minimum. Thus, an edge with a low weight is a likely candidate to have a high value for betweenness centrality.

We implement the clustering function ***C*** as a routine that makes use of two distinct stages, each with specialized purpose, as seen in the simple diagram in Figure 3.14.

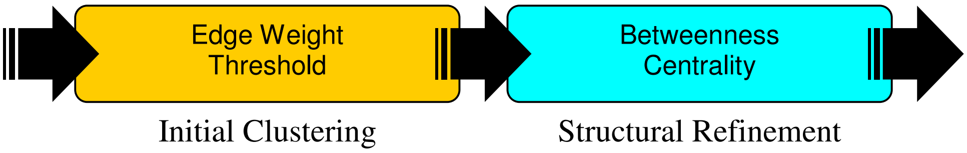


Figure 3.14: Two-stage clustering process..

This architecture performs a simple, initial clustering by deleting negligible edges that are weighted below a threshold that is considered as the noise floor for edge weights. The betweenness centrality computation would then be performed for each of the edges of the resulting graph, which would be one that contained only substantial edge weights. After this computation, edges with high betweenness centralities would then be removed.

The necessity for an initial clustering stage despite the fact that betweenness centrality clustering alone would be able to identify edges with low weights lies in the fact that there exists a failure mode that is seen to occur with considerable frequency.

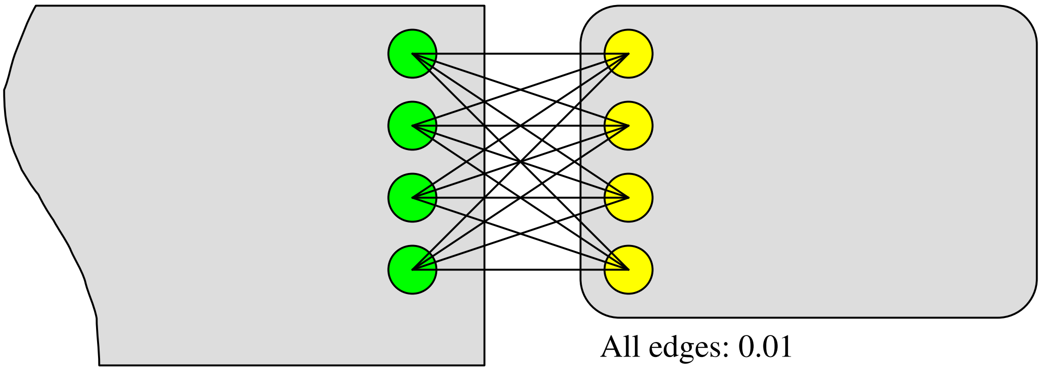


Figure 3.15: A failure mode for betweenness centrality clustering.

Consider the specific graph configuration illustrated in Figure 3.15. In this special case, a set of feature points belonging to two objects are well connected for one reason or another, such as when they satisfy the spatial distance criterion. However, since they belong to different objects with independent motion, the edge weights would garner low scores (0.01 in this example) due to the constraints imposed by the coherence of motion model, as well as frequency domain analysis. Because of the way that betweenness centrality is computed as shown previously in Figure 3.11, the number of parallel paths would cause the denominator $\boldsymbol{n_{st}}$ to increase, thus lowering the overall betweenness centrality value. While it could be argued that the well connectedness of the vertices might suggest a strong likelihood that they belong to the same object, this simply could not be the case if the scores are unreasonably low, such as 0.01 in this example.

The simplest solution to this is to filter out unreasonably low scores even before performing the betweenness centrality computations, and consider them as below a threshold that is not unlike a noise floor. We find this to be a reasonable operation, since low edge scores do denote low likelihoods already, and could be considered as not even worthy of consideration.

The inclusion of a simple edge weight threshold filter before the betweenness centrality operation makes the latter essentially a module geared more towards topological structure refinement rather than one that detects low edge weights. Structural refinement is in fact a more challenging task than simply identifying low edge weights, since while the latter can be done by simple numerical thresholds, figuring out topological entities such as highly connected subgraphs and bridge-like edges are less than straightforward.

Betweenness centrality would be able to identify these two types of entities within a graph due to the nature of its computation. Bridge edges would have numerous shortest paths passing through it simply because there are no other edges to provide the link, thus giving them high betweenness centrality. Highly connected subgraphs on the other hand, which usually indicate a valid target object, would have many parallel edges within them, thus reducing the concentration of shortest paths for each edge and lowering their betweenness centrality.

Thus, our two-stage clustering process proceeds as follows:

1. Delete edges with weights that fall below a certain weight threshold
2. Compute betweenness centralities for all edges
3. Delete edges with centralities that are above a certain centrality threshold

After this clustering operation, the remaining edges would define the final output graph $\boldsymbol{G_F}$, whose connected components would correspond to target human beings, as illustrated in Figure 3.16. A method to interpret this data, experimental results, as well as ideas for the Postprocessing stage mentioned in Section 3.2 shall be discussed in the next chapter.

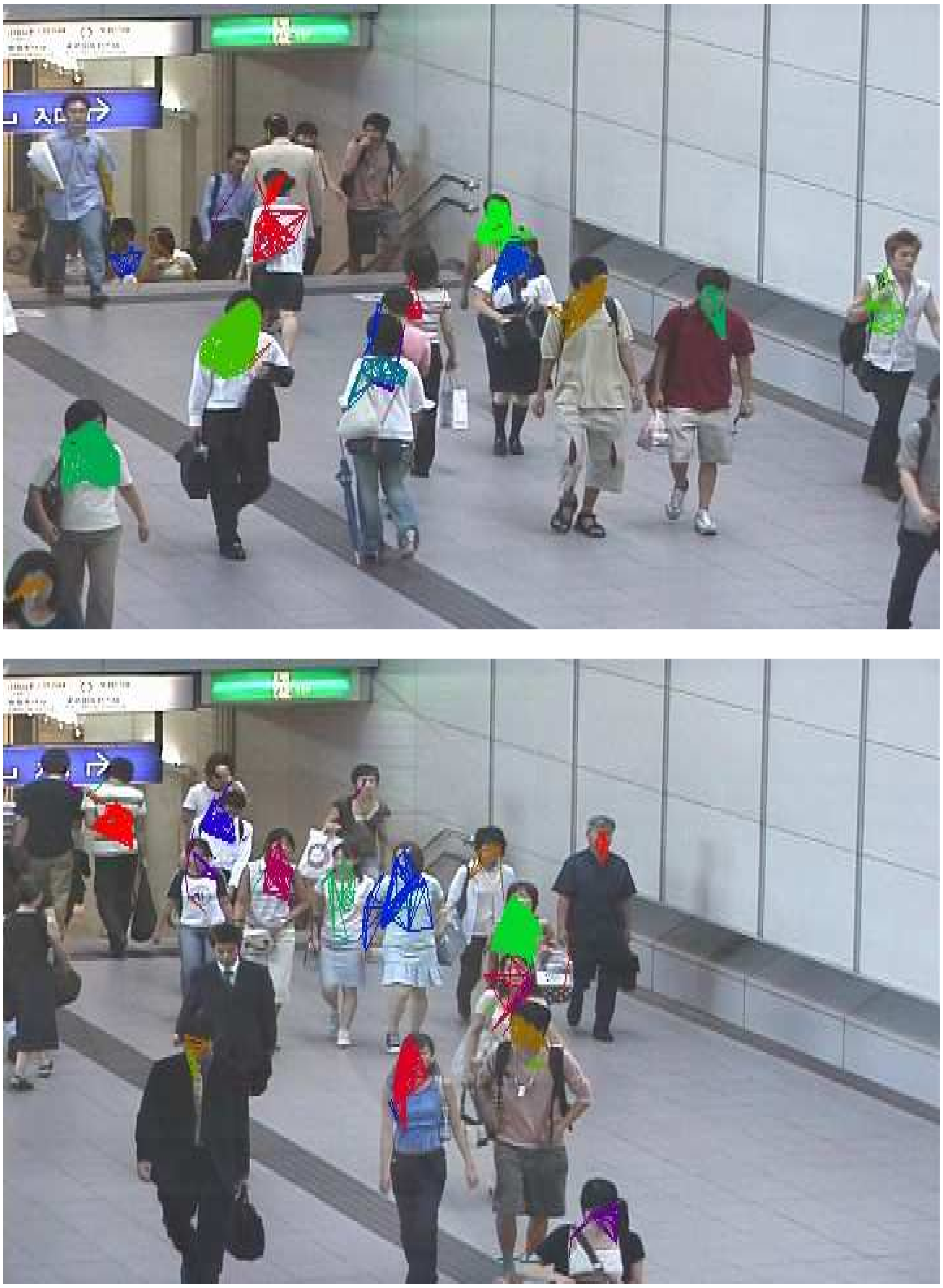


Figure 3.16: Sample outputs: Graph connected components drawn over input frame.

Chapter 4

Experiments, Results and Analysis

This chapter deals with actual experimental results with real data, using the system we have described in the preceding chapter. It is divided into two main divisions with two sections each. The first parts are discussions on how to interpret raw experimental data, while the second part deals with the results themselves, followed by analysis.

4.1. Interpretation of Clustering Output

It is quite common to mistake raw system output as the experimental and testing result that we seek. While there may be cases that the information provided by the test program or test system is indeed synonymous to experimental results, we would like to believe that there is an extra layer of interpretation that transpires between raw system output and experimental results, whether it be done consciously or unconsciously. In our specific case, care must be taken not to readily accept the set of connected components given by the clustering routine, and equate them with humans in the scene. In this section, we show our specific interpretation of the connected components and its vertices with regards to our greater goal of identifying and characterizing the motion of humans in crowded scenes.

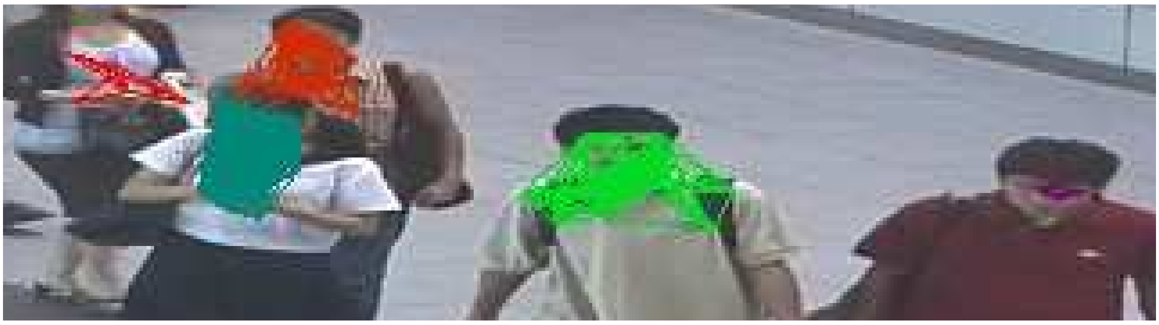


Figure 4.1: Is obtaining the clusters still not enough?

At this point, it is important to note that although the feature points were originally selected as parts of the humans populating the scene, much processing has been done and numerous non-idealities have occurred during the clustering process that it is only naïve to quickly equate vertices and clusters to people without going into much consideration. In Section 3.4.4, for example, particularly in Figure 3.12, we see the possibility of noise-like yet valid edges being formed because of non-idealities and small coincidences in individual feature point motion.

To consider these non-idealities, we abandon the assumption that feature points in a connected component (i.e., a cluster) all lie on the target person. We then replace it with an equally effective yet more tolerant definition that the vertices in a connected component are feature points that are merely localized around the presence of a target human. This description is essentially the same as the first, but relaxes the constraint that the feature points must necessarily lie on the same human being. In our work, we implement this concept by taking the average location for all feature points in a cluster as in Figure 4.2, and consider only the result as the position of the target object. After the computation of this average, the information stored by the individual feature points would then be generally considered as obsolete. By adopting this methodology, the existence of outliers that may arise through non-idealities in the input video is fully taken into consideration. Due to their nature of being sporadic and being exceptions rather than norm, such occurrences are said to have little effect in the computation of the average.

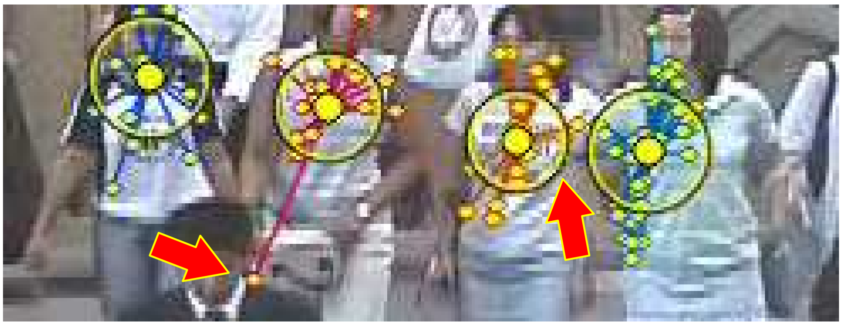


Figure 4.2: The impact in performance of outlier feature points in each cluster (red arrows) is reduced (filtered)by simply considering each cluster's average feature point position.

Another step that was done in our testing was the disregarding of small clusters. We consider cluster sizes of 4 or less vertices to have insufficient information, and are therefore subsequently removed. We find this to be a sound argument as well, since it would be easier for (e.g.) 2 feature points to move together by coincidence and produce a stray cluster, than for (e.g.) 20 points to move together by coincidence.

4.2. Postprocessing

The process of interpreting raw system output isn't limited to computation of simple derived properties. One is also free to perform relatively involved calculations for the sake of having a better understanding of the information that the experimental setup provides. We consider this process of postprocessing as the third stage of the architecture defined in section 3.2. In the case of our research, this stage refines the clustering stage results and provides the user with a refined interpretation of the main clustering block.

Though this stage would not be the main focus of our study, we provide a simple postprocessing block that may be the basis for future work. In our experiments, we perform a *metaclustering* (cluster merging) operation after obtaining the average position of each cluster. This simple metaclustering scheme only relies on spatial data.

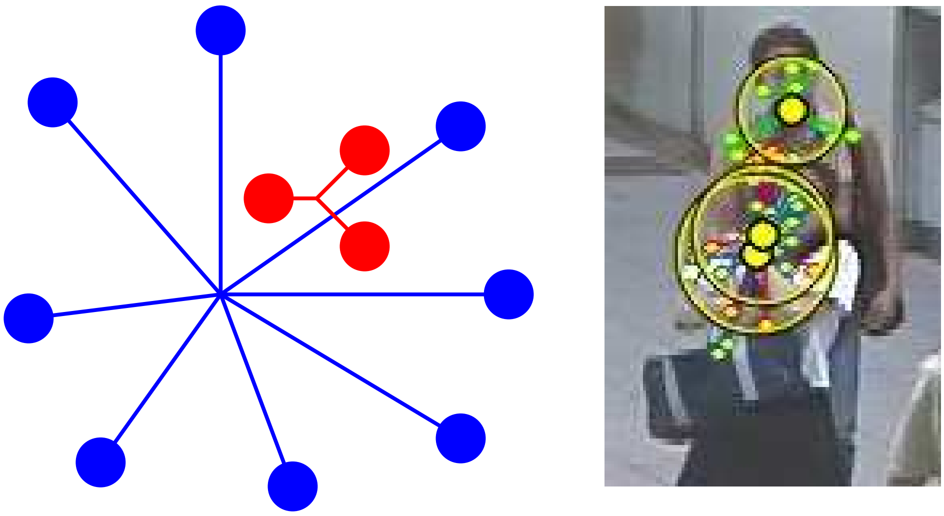


Figure 4.3: Interlaced clusters. (left) Illustration. (right) Actual manifestation.

Consider the case of Figure 4.3, in which two clusters are interlaced within each other in such a way that it would be impossible for them to be interpreted as two separate objects. In the simple illustration at the left, the small connected component is suspended within the larger connected component, such that they cannot be considered as two humans, considering common knowledge of the dimensions of a person and some intuition on occlusion. In the actual snapshot at the right, the same phenomenon is seen, and is likely to be related to the fact that there is much human body articulation at that instant of time (she is seen to be fixing her bag/clothing).

Regardless of the cause, the fact would remain that the two clusters cannot be resolved as a two separate human beings. Rather than ignore this fact and proceed to count the two clusters as two persons, better alternatives would be to either discard one of the clusters, or merge the two clusters into one.

In our work, we go the latter route and perform a cluster merge. The decision to merge two clusters is based on simple collision detection. The average position (the "center") of the feature points within each cluster is computed, followed by the average distance (the "radius") from each feature point to the center. This average radius would serve as a simple figure to quantify the extents of the cluster in space, and other clusters that would penetrate these extents would be candidates for merging. As illustrated in Figure 4.4, for every pair of clusters, it is checked whether the center of one would be inside the circle formed by the center and radius of the other.

It should be noted that this scheme is simplistic and merely a proof of concept. Future researchers may decide to tackle this topic in more detail, as we believe that there is still unlocked potential with regards of interpreting clustering results, such as considering metacluster trajectories, as well as extending the analysis to the temporal dimension.

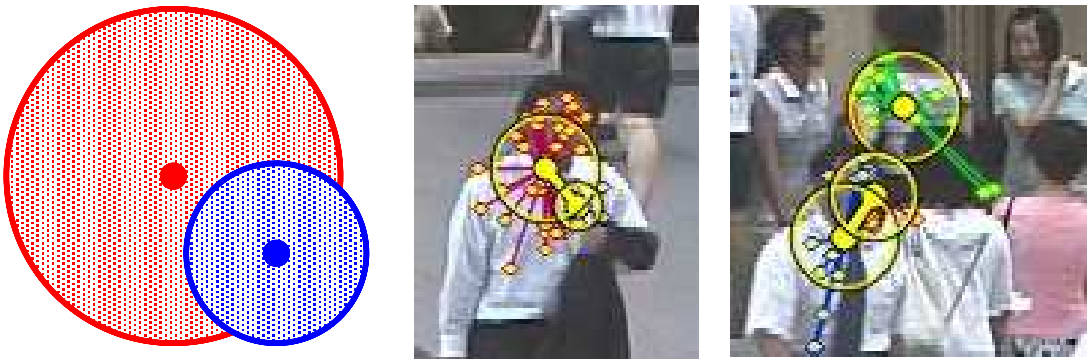


Figure 4.4: (left) A collision occurs when the center of one cluster is within the radius of the other. (right) Sample cluster merges.

With an efficient and principled means of interpreting the raw output graph given by the clustering process, performance evaluations could then be done.

4.3. Design of Experiment

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 4.5. 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, which is subsequently scaled vertically to restore the original resolution of 720x480. The main interest region is populated by 10 to 20 people at a time.

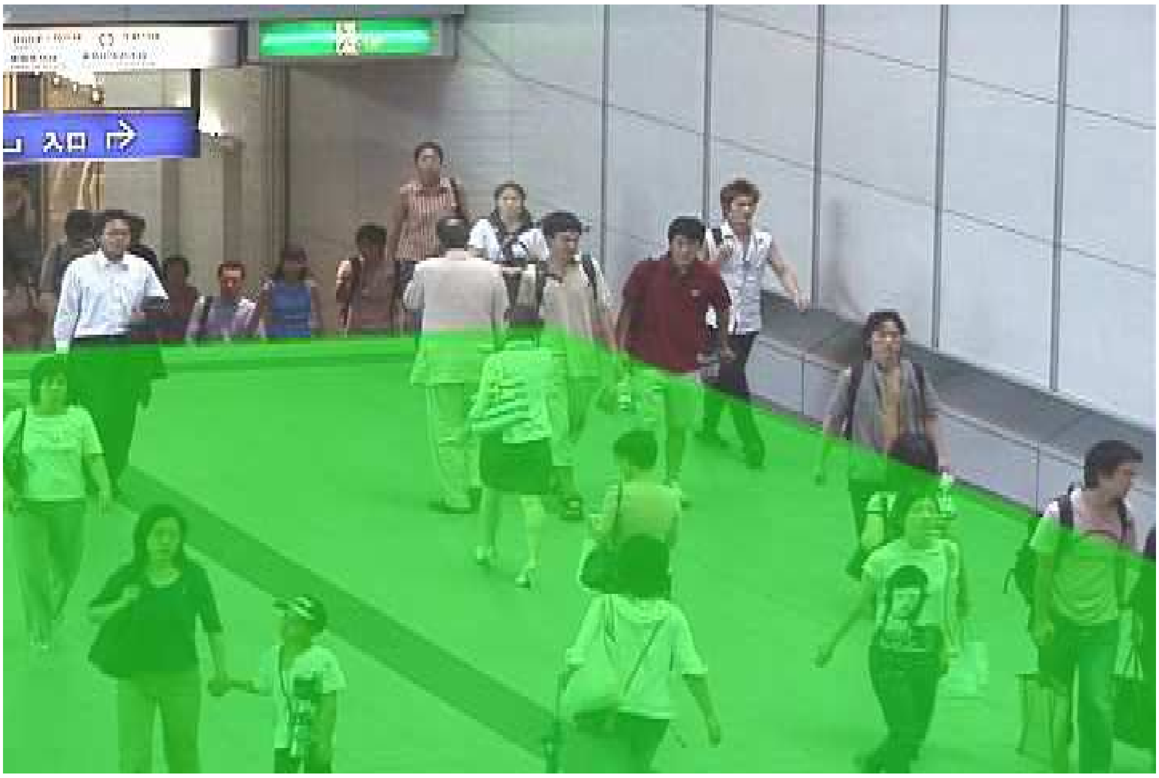


Figure 4.5: Use of the scene's main floor as the main interest region. Humans are counted as long as their feet touch the floor indicated by the shaded region.

We measure system performance using four general parameters: *Hit Rate*, *Cluster/Person Ratio*, *False Positive Rate* and *False Negative Rate*. We perform our experiment by taking random frames and their clustering results, and recording the pertinent data. Although the number of connected components, metaclusters and vertices are computed by the experimental setup, the counting of the number of humans in the scene (i.e., the ground truth) is done by hand.

We define the Cluster/Person Ratio (CPR) 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.

$$CPR = \frac{clusters\_with\_humans}{humans\_with\_clusters} \tag{4.1}$$

A CPR 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 false 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 (fragmentation).

4.4. Results and Analysis

We now present a summary of the final experimental results, listing each performance parameter's average value for 100 randomly selected frames.

Hit Rate	% False Negative	% False Positive	CPR
70.53%	29.47%	0.24%	108.88%

Table 4.1: System Performance Parameters.

Table 4.1 can be considered as a numerical summary of system performance. We see acceptable performance especially in terms of false positives, and the main reason for this is seen to be the foreground selection mask presented in section 3.3.3. 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 [ST-94]. They would therefore have fewer, erratic feature points and thus have edges of lower quality that are prone to deletion during the clustering process.

The CPR also takes on a satisfactory value, close to the ideal 100%. It should be noted, though, that this variable is a figure describing very general cluster assignment performance at best; readers who would be interested in the deeper details of the system's cluster assignment performance would do well to consider the false negative rate above, and the discussion on fragmentation below.

It was noted that fragmentation happens in a fairly regular basis, and almost all cases were related to articulation of the human body, such as articulations of the head and shoulders, the presence and articulation of limbs, and the movements of accessories like handbags. We also suspect less obvious causes such as shifting of clothing, though this remains unfounded due to the difficulty of properly identifying such scenarios.

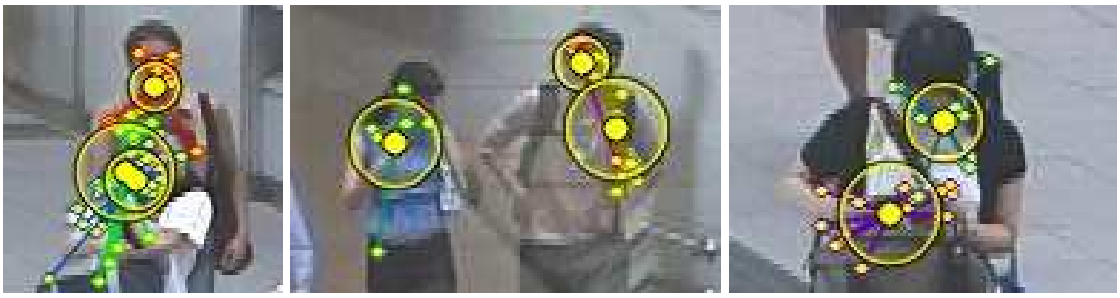


Figure 4.6: Fragmentation scenarios.

Figure 4.6 depicts actual fragmentation scenarios. In the leftmost image, a person moves her arm to sling her bag. In the middle, a person turns his head to speak with his partner. At the rightmost is a person shuffling through her handbag.

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 the figures presented in this chapter thus far, 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.

To further confirm the effectiveness of Fourier analysis, we do a side-by-side comparison of the clustering performance of the system with and without the frequency analysis module. Rather than use numerical figures to compare performance, we compare the output frames. While this is a purely qualitative exercise, we see this as a better way to appreciate the added clustering performance that frequency domain analysis can offer.

Figures 4.7A to 4.7C illustrate the performance benefit of the frequency analysis block. For each set, the top image depicts the output with frequency analysis turned on, while the middle image illustrates the results when the frequency analysis block is turned off. The bottom image depicts the results of a system with frequency analysis turned off and the constraints of the motion analysis block tightened in an attempt to improve performance. It can be seen that simply tightening the constraints in the bottom images will only introduce fragmentation and do little to provide any notable clustering improvement.

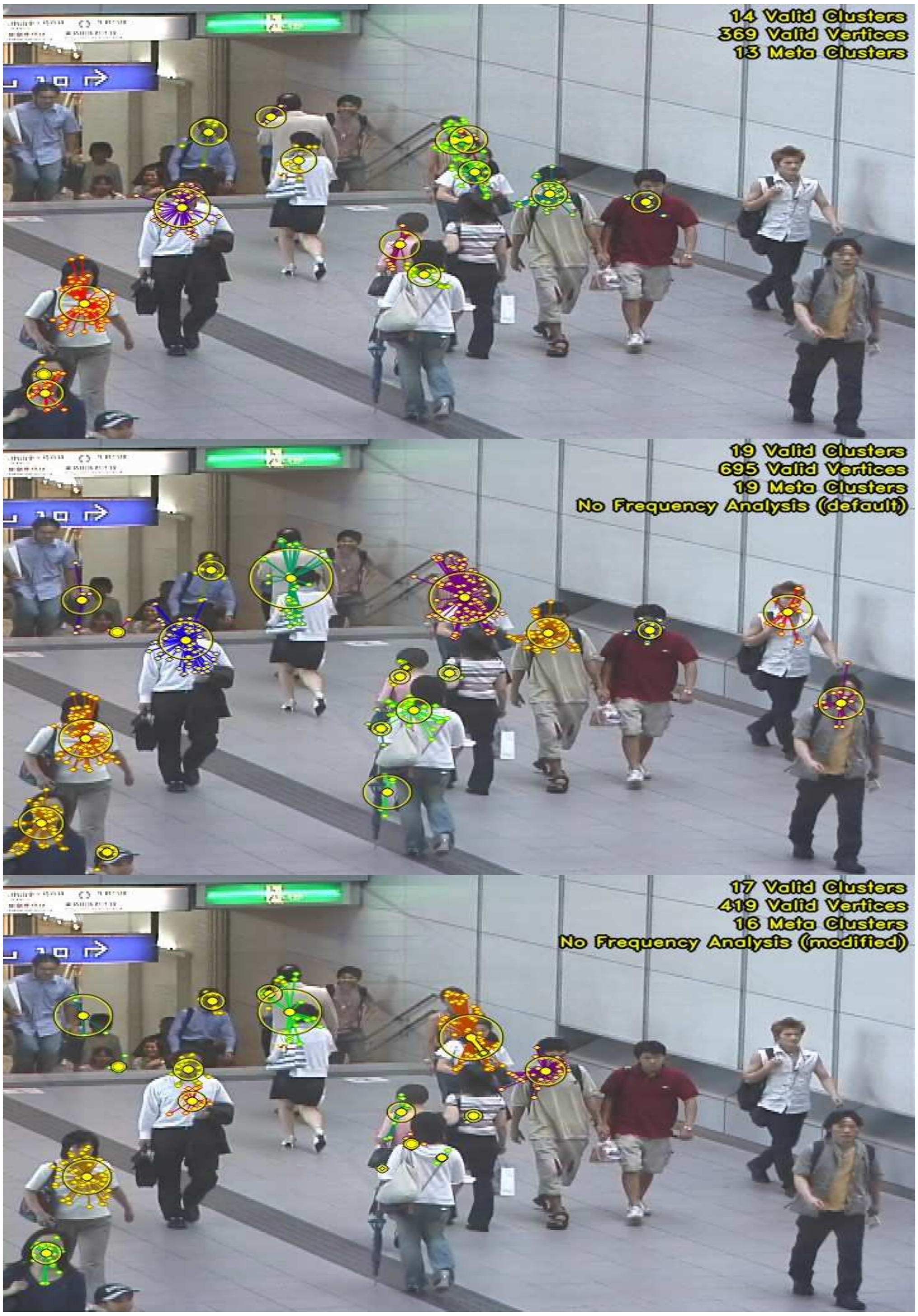


Figure 4.7A: Output comparison for systems with (topmost) and without (lower) Fourier analysis.

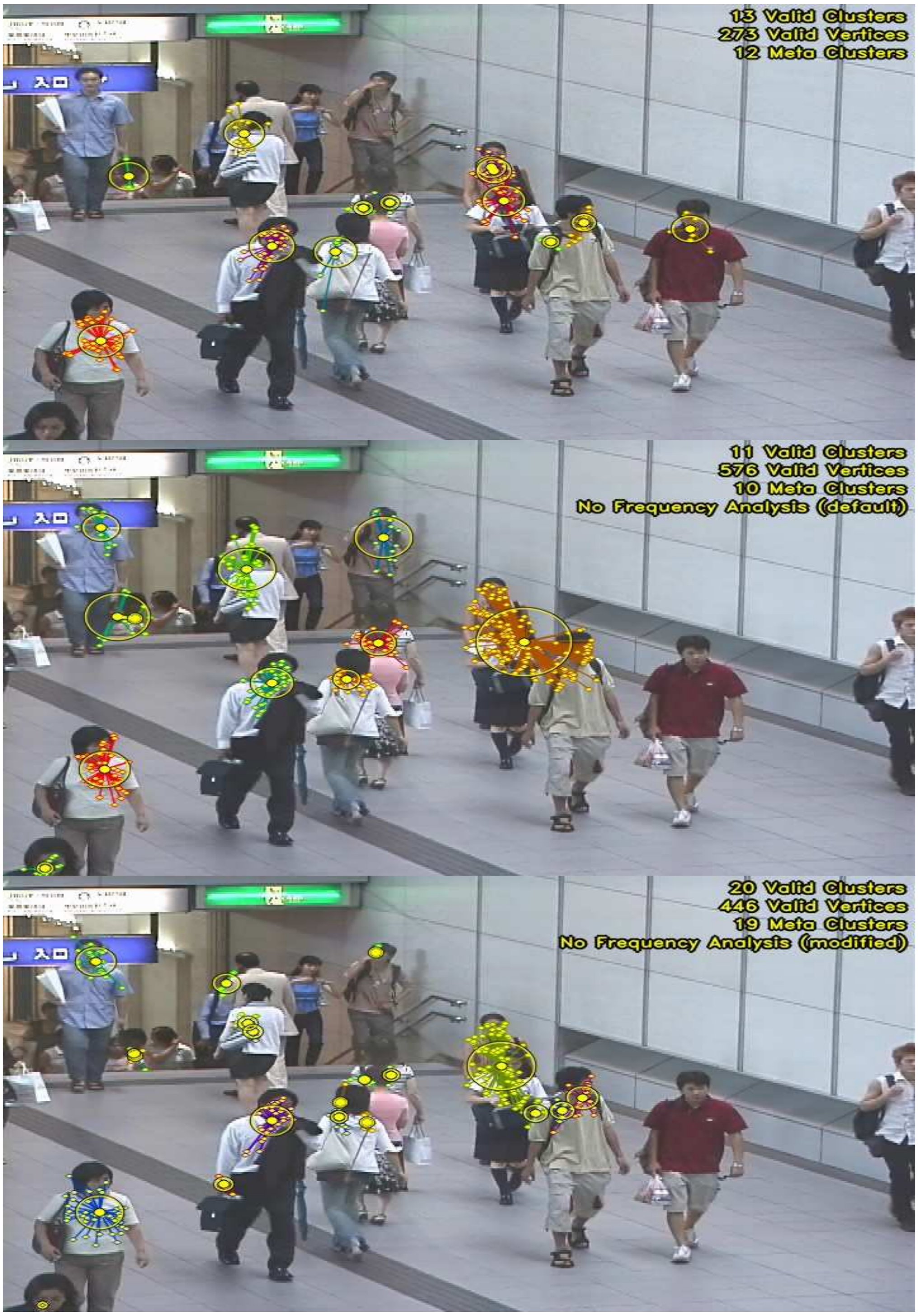


Figure 4.7B: Output comparison for systems with (topmost) and without (lower) Fourier analysis.

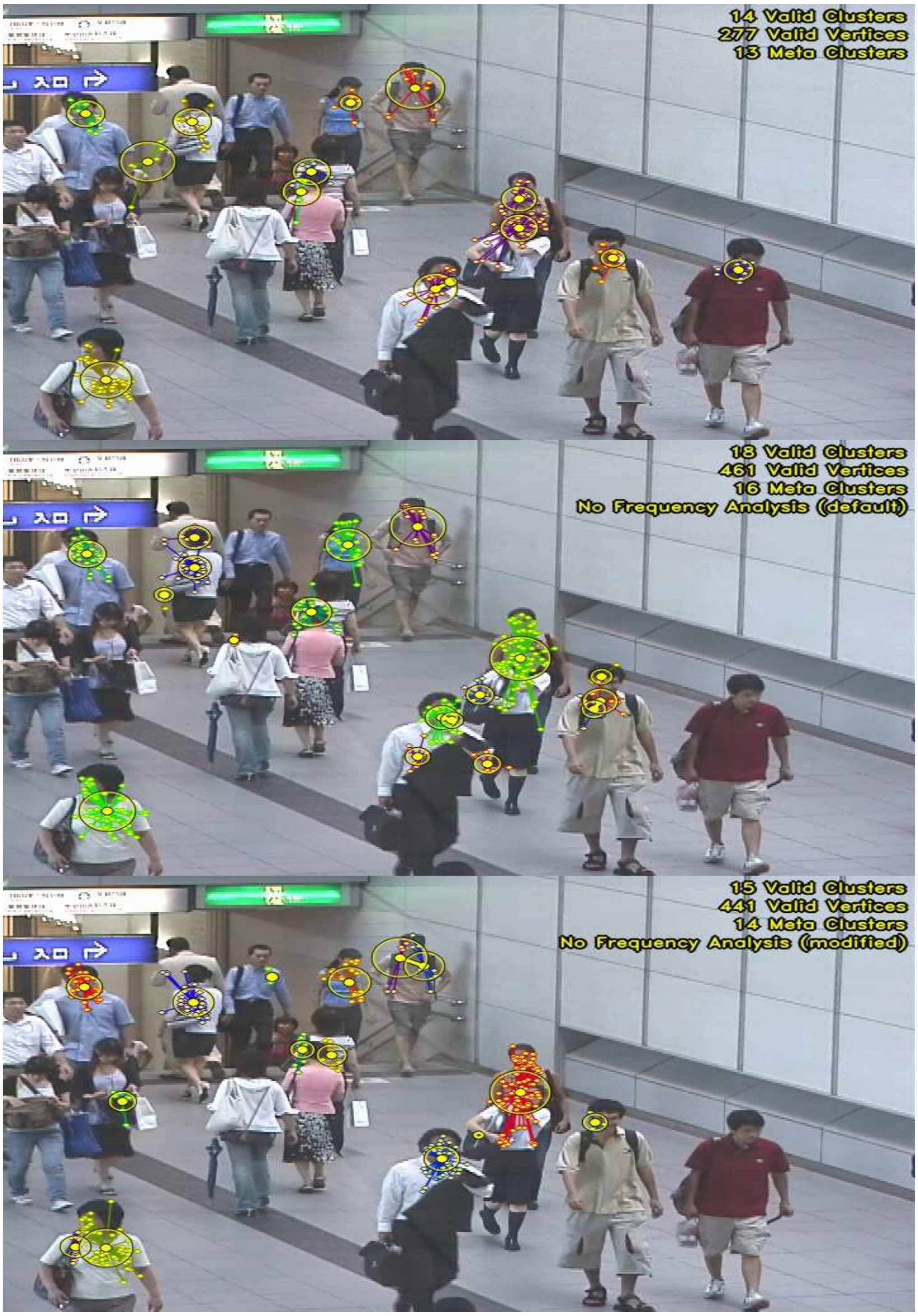


Figure 4.7C: Output comparison for systems with (topmost) and without (lower) Fourier analysis.

Chapter 5

Conclusion

5.1. Summary

We have presented a system that utilizes frequency domain analysis in addition to trajectory analysis during feature point tracking and clustering, to address the limitations that were identified in the use of coherence of motion as a metric for clustering. It also features a few other novel ideas, such as using object detection as a feature selection mask and using a unique two-stage clustering scheme using the concept of betweenness centrality.

The system 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 is largely due to the increased sensitivity provided by frequency domain analysis, which is able to differentiate two humans just by their gait cycles alone.

The use of a foreground-background mask in feature point detection has also helped in keeping the false positive rate at a very low figure. The use of object detection presents many advantages over background subtraction for crowded scenes, due to the nature of crowded scenes to rarely reveal background pixels.

Finally, our “score-and-structure” clustering method, in its simplicity of concept, has shown satisfactory if not good performance in terms of analyzing the weighted graph into valid clusters.

Certain roadblocks do exist that prevent us from realizing a 1:1 person-to-cluster ratio. The lack of texturedness for back view heads and some articles of clothing leads to low feature count for some people, and this ultimately leads to poorly formed clusters, or even the absence of clusters. The fragmentation phenomenon, on the other hand, does pose a significant challenge to system accuracy. For the latter, a simple, proof-of-concept postprocessing routine has been discussed, and we are confident that future researchers who would choose to investigate further into this class of computations would be met with good results.

5.2. Insights and Recommendations for Future Work

We believe it is important to document the insights we have gained through the course of this work and present them as recommendations for future research, for the sake of readers who would want to investigate this particular field in more detail. Below are the most important points we have observed during the course of this study, along with our recommendations, in no particular order:

- **Low Textured Targets**

  For our test video, there have been occurrences of targets that are low in the texturedness requirements specified in [ST-94]. In our case, these people were mainly those facing away from the camera and people in plain, dark-colored clothing. The number of feature points detected for these people as compared to those who have high texture content is substantially less, and this contributes to the formation of weakly formed or even ill formed clusters. In our results, some occurrences of fragmentation are due to the clustering and structural refinement operations on these weak clusters.

  Unfortunately, this is an inherent disadvantage of the feature point selection algorithm. Furthermore, our study focused on the clustering process itself and not the data acquisition process, and only used standard implementations of feature point detection and clustering. Since the quality of the input graph is only equivalent to the quality of feature points of which it is composed, the use of an improved feature point selection algorithm is highly recommended.

- **Frame Rate Over Resolution**

  For our study, we had the opportunity to operate on two types of data sets, one of which is a high resolution, progressive scan video footage. Unfortunately, due to the limitations of our equipment, the progressive scan video could only be taken at a frame rate of 15 frames per second. It was observed that while the standard implementation of optical flow does work for this particular clip of video, the average trajectory lifetimes were significantly shorter and were mostly unusable to our trajectory-centric computational modules.

  Therefore, at this point we echo the concerns documented by [RB-06], which stresses the importance of a high frame rate for studies of this type. It intuitively makes sense as well, since for this study we focus more on motion rather than form. Our system would thus favor video clips with richer motion data (frame rate), than those with rich spatial data (resolution).

- **Sensitivities and Fragmentation**

  Our system utilizes several functional blocks with adjustable sensitivities, such as those of motion coherence and frequency domain analysis. It is probably quite common to initially think that increasing the sensitivity automatically increases accuracy. However, during the early parts of our research, we have seen that there is a limit to how one can set the sensitivity, and any more would just lead to fragmentation.

  For accurate detection and tracking, it has been seen that it was not an issue of sensitivity at all. It has been, from the start, not an issue of "tweaking", but an issue of the accuracy of the models themselves. In our previous testing, for example, while attempting to successfully separate the clusters of two people walking in the same direction using only the model for coherence of motion, we were never able to obtain highly satisfactory results. Overspecifying the sensitivity in this case only led to unpredictable fragmentation. After adding a model for frequency domain analysis, however, the performance greatly improved without even a need for much parameter adjustments.

  Thus, future work in this field should never take parameter tuning as its focus of research. It is instead recommended to identify more characteristics of human motion that could be expressed in terms of a model, or improve the existing models themselves instead of merely tuning their parameters.

- **Temporal Coherence**

  This work, as well as [RB06] and [BC-06] that preceded it, do not resolve temporal coherence of clustering results – i.e., new sets of graphs with new sets of results are computed for every frame. However, we see temporal coherence as a relatively trivial possibility for future works, since feature points and their tracked trajectories persist over time, and this data may be used to draw cluster-to-cluster correspondences across frames of video.

- **Postprocessing**

  One area which we believe holds tremendous potential is in postprocessing; i.e., the routines that take the raw output clusters and perform further computations, usually for the sake of interpreting them into useful data. At this point, we see postprocessing as the most viable solution to the fragmentation problem, particularly fragmentation due to human body articulation. We have presented a very simple cluster merging technique in this work, and it is indeed possible to extend this technique to use more involved models.

Bibliography and References

[Br-01] U. Brandes, “A Faster Algorithm for Betweenness Centrality”, In Journal of Mathematical Sociology, 25(2), pp.163–177, 2001.

[BC-06] 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.

[FS-97] 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.

[FS-99] Y. Freund and R. E. Schapire, “A short introduction to boosting”, In Journal of Japanese Society for Artificial Intelligence, Volume 14, Number 5, p.771-780, September 1999

[GCS-04] C. Gentile, O. Camps and M. Sznaier, “Segmentation for Robust Tracking in the Presence of Severe Occlusion”, In IEEE Transactions on Image Processing, Volume 13, Number 2 February 2004.

[LK-81] 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.

[MR-03] R. Meir and G. Rätsch, “An Introduction to Boosting and Leveraging”, In Advanced Lectures on Machine Learning (LNAI2600), 2003.

[OTdFLL-04] Okuma, Taleghani, de Frietas, Little and Lowe, “A Boosted Particle Filter: Multitarget Detection and Tracking”, In Proceedings of the Eighth European Conference on Computer Vision, Volume 3021 of Lecture Notes in Computer Science, p.28-39, 2004.

[PMTH-01] 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.

[RB-06] V. Rabaud and S. Belongie, "Counting Crowded Moving Objects", In Proc. Conference on Computer Vision and Pattern Recognition, Volume 1, pp.705-711, 2006.

[RD-06] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection", In European Conference on Computer Vision, May 2006.

[RF-03] 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.

[ST-94] J. Shi and C. Tomasi, "Good features to track" In Proc. Conference on Computer Vision and Pattern Recognition, pp.593-600, 1994.

[TR-04] P. Tu and J. Rittscher, "Crowd Segmentation Through Emergent Labeling", ECCV Workshop SMVP, pp.187–198, 2004.

[VDP-03] J. Vermaak, A. Doucet and P. Pérez, "Maintaining Multi-Modality through Mixture Tracking", In Proceedings of the Ninth IEEE International Conference on Computer Vision – ICCV'03, p.1110-1116, 2003.

[VJ-01] 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.

[VJ-04] P. Viola and M. Jones, "Robust Real-Time Face Detection", In International Journal of Computer Vision, Volume 57, Issue 2, pp.137-154, 2004.

[ZN-CVPR04] 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.

[ZN-PAMI04] 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.

[ZT-03] Yue Zhou and Hai Tao, "A Background Layer Model for Object Tracking Through Occlusion", In Proceedings of the International Conference on Computer Vision, pp.1079-1085, 2003.

[Boost] Boost Graph Library.
http://www.boost.org/libs/graph/doc/index.html

[FFTW] FFTW Fast Fourier Transform Library.
http://www.fftw.org/

[OpenCV] Open Source Computer Vision Library.
http://www.intel.com/technology/computing/opencv/index.htm

Appendix

A. Program Settings

As suggested by the text in Chapter 3, our system makes use of a number of numerical settings such as thresholds and buffer lengths. This section lists these settings in detail.

```
FEATURE_COUNT = 1024;
```

The system is capable of tracking 1024 feature points at a time.

```
FFT_ARRAY_SIZE = 64;
TRACE_LENGTH = 128;
```

Trajectories are traced backwards for up to 128 frames, and forwards for another 128 frames, bringing the maximum buffer length to 257. Of these, the data for 64 frames are used are used as input to the Fourier Transform computations.

```
OBJECT_SIZE_X = 80; OBJECT_SIZE _Y = 240;
```

Settings for expected object size in pixels, to be used by the spatial analysis block.

```
MAX_TRAIL_LENGTH = 256;
MAX_PATH_STDDEV = 4;
MAX_FREQ_DIFF = 32;
MAX_PHASE_DIFF = M_PI/4;
```

Settings for maximum values in mapping functions, for coexistence, coherence, frequency, and phase, respectively. M_PI is π, the ratio of a circle's circumference to its diameter. Trajectory standard deviation is measured in pixels, while the frequency difference is in terms of discrete samples in the FFT algorithm's output array.

```
EDGE_WEIGHT_THRESHOLD = 0.2;
```

Edges with weights below this value are discarded during the clustering process.

```
CENTRALITY_THRESHOLD = 8;
```

Edges with betweenness centralities beyond this value are discarded during the second step of the clustering process.