UNIVERSITÉ DE MONTRÉAL

SYSTEMATIC PARAMETER OPTIMIZATION AND APPLICATION OF AUTOMATED TRACKING IN PEDESTRIAN-DOMINANT SITUATIONS

DARIUSH ETTEHADIEH
DÉPARTEMENT DES GÉNIES CIVIL, GÉOLOGIQUE ET DES MINES
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE CIVIL)
DÉCEMBRE 2014

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

SYSTEMATIC PARAMETER OPTIMIZATION AND APPLICATION OF AUTOMATED TRACKING IN PEDESTRIAN-DOMINANT SITUATIONS

présenté par : ETTEHADIEH Dariush
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. BILODEAU Guillaume-Alexandre, Ph. D., président
M. SAUNIER Nicolas, Ph. D., membre et directeur de recherche
M. FAROOQ Bilal, Ph. D., membre et codirecteur de recherche
M. MIRANDA-MORENO Luis F., Ph. D., membre

RÉSUMÉ

Les mouvements des piétons et leur modélisation constituent un domaine de recherche de plus en plus actif. Bien qu'encore souvent appliqué à la sécurité par l'élaboration de plans d'évacuation en cas d'urgence, comprendre le mouvement des piétons est un enjeu économique de plus en plus important, notamment pour améliorer l'efficacité des aménagements de transport et des grands centres commerciaux.

Cependant, les données existantes — particulièrement au niveau individuel, ou microscopique — sont majoritairement collectées dans des situations expérimentales contrôlées. Elles ne sont donc pas nécessairement représentatives du comportement des piétons dans des situations réelles, particulièrement en tenant compte de la susceptibilité de leur comportement aux facteurs démographiques, psychologiques et environnementaux. Cette lacune est due principalement à l'absence de méthodes prouvées pour la détection et le suivi de piétons dans des cas réels, absence qui résulte de la complexité des mouvements piétons et qui persiste malgré l'avancement continu des méthodes automatique d'analyse.

De ces méthodes, la plus prometteuse est peut-être la détection et le suivi automatisé de piétons à partir de données vidéo. De tels outils ont non seulement démontré une capacité de suivi excellente (une précision atteignant 85 % dans certain cas), mais permettent aussi d'analyser des données vidéo enregistrées par les caméras de surveillance déjà installées.

De tels résultats sont toutefois assujettis à deux limitations importantes, présentes dans la vaste majorité de la littérature. Premièrement, ces outils sont généralement testés sur une seule scène de mouvements piétons, ou sinon sur des scènes très similaires. Leur capacité à reproduire les performances publiées lorsqu'appliqués à une plus grande variété de cas est ainsi difficile à vérifier. Ceci est problématique car de nombreux problèmes affectent la performance des outils d'analyse vidéo et ces problèmes peuvent varier de manière importante entre scènes. Notamment, un piéton change de forme de manière continue lors de son mouvement, peut facilement être partiellement ou entièrement caché par un obstacle ou d'autres individus, et — contrairement aux véhicules routiers — n'est sujet à aucune contrainte concernant son trajet ou sa vitesse en dehors des obstacles physiques de son environnement.

La seconde limitation notée de ces outils est la manière dont ils sont calibrés. En effet, bien que la conception et la fusion de méthodes de détection sont communes, les paramètres sous-jacents semblent être le plus souvent choisis manuellement. De plus, avec une seule exception, ils n'ont jamais explicitement été le sujet d'optimisation rigoureuse. Il est ainsi probable que la performance optimale de ces outils n'a pas encore été révélée.

L'objectif du travail réalisé ici est donc la conception d'un algorithme générique d'optimisation des outils de suivi de piétons. Acceptant comme entrée des trajectoires extraites manuellement d'une courte séquence vidéo (vérité terrain) et les paramètres de l'outil à optimiser, cet algorithme (nommée TrOPed, ou *Tracker Optimizer for Pedestrians*) produit des paramètres calibrés pour produire les trajectoires les plus proches de la vérité terrain pour une scène particulière.

TrOPed est composé de trois fonctions principales. Au coeur est l'algorithme d'optimisation du recuit simulé (*simulated annealing*). Sélectionné pour son efficacité d'optimisation dans un domaine de recherche a priori inconnu (un facteur important vu la généralité désirée de l'algorithme ainsi que le temps important nécessaire à l'obtention de trajectoires dans la majorité des outils de suivi), cet algorithme régit de manière probabiliste le choix des paramètres de l'outil d'optimisation. Comme il permet le recul vers une solution inférieure, le recuit simulé peut s'échapper aux optima locaux et peut ainsi mieux localiser l'optimum global recherché.

Les paramètres sont déterminés à chaque itération par la seconde fonction, celle de mutation stochastique. Utilisant initialement des paramètres définis par l'utilisateur en fonction de leur distribution attendue, cette fonction réduit graduellement l'amplitude des ajustements des paramètres selon l'avancement de l'algorithme afin d'accélérer la convergence tout en assurant une solution finale précise.

La performance de chaque itération est évaluée par la troisième fonction, utilisant les métriques CLEAR MOT : MOTA (*Multiple Object Tracking Accuracy*, mesurant l'exactitude des trajectoires produites en fonction du nombre de piétons non détectés, de surdétections et d'associations fautives, avec une valeur optimale de 1) et MOTP (*Multiple Object Tracking Precision*, l'erreur spatiale moyenne, en mètres). Ces deux mesures sont combinées en une seule selon leurs poids relatifs, définis par l'utilisateur. Toutefois, les essais effectués ont démontré que les solutions optimales sont atteintes en utilisant MOTA uniquement.

Finalement, TrOPed inclut des mécanismes et paramètres additionnels, optimisés en parallèle à ceux de l'outil calibré, qui permettent d'optimiser la méthode typique de projection des trajectoires de l'espace image des vidéos vers les coordonnées réelles. Cette transformation est communément effectuée vers des coordonnées définies au niveau du sol, ce qui est peu problématique lors d'enregistrement fait à longue distance ou de véhicules. Cependant, dans le cas de piétons (qui sont notamment plus grands que larges) et la proximité typique des caméras utilisées pour les enregistrer, la différence entre le plan du sol et le plan parallèle dans lequel les piétons sont détectés devient une source importante d'erreur. Des paramètres régissant l'élévation de ce second plan ont donc été inclus.

Les essais de TrOPed ont été effectués sur deux outils de suivis : *Traffic Intelligence* (TI) et*Urban Tracker* (UT). Ces deux outils utilisent des méthodes différentes de détection, permettant de vérifier la généralisabilité de l'algorithme : TI identifie le mouvement de groupes de pixels et les regroupe en piétons, et UT distingue les "blobs" de mouvement par comparaison avec l'arrière-plan statique.

Afin de traiter différents niveaux de complexité, trois scènes ont été étudiées : un corridor central à l'université Polytechnique Montréal, l'entrée d'une station de métro, également à Montréal, et un passage piéton au centre-ville de la ville de New York localisé en face de la station de train centrale. Les deux premiers cas ont été enregistrés pendant l'heure de pointe matinale, avec des caméras installées à un angle et une distance approximant ceux typiques des caméras de surveillance. Le troisième cas, quant à lui, a été enregistré entre 10h et 13h un jour de semaine, et la caméra installée verticalement directement au dessus du passage. Pour toutes les scènes, deux séquences de une minute chacune — choisies pour leur représentativité des scènes en général — ont été extraites et les trajectoires des piétons individuels extraites manuellement. De ces séquences, l'une a servi à la calibration par TrOPed, et la seconde à vérifier si les paramètres calibrés étaient adaptés aux scènes en entier ou étaient trop optimisés pour la première séquence (suroptimisation).

Dans tous les cas, les paramètres optimisés par TrOPed ont produit des trajectoires d'exactitude et de précision supérieures à celles obtenues par calibration manuelle des outils ; en moyenne, cette amélioration s'est traduite par une réduction de 50 % (+/- 15 %) des erreurs commises. Cette amélioration s'est maintenue lors des essais sur les séquences tests, malgré une légère baisse de performance attribuable à la suroptimisation. L'amélioration a aussi été maintenue peu importe les paramètres initiaux, confirmant que la solution finale représente très probablement un optimum global.

Lors des initialisations sur des paramètres choisis arbitrairement, ces résultats ont été obtenus après une centaine d'itérations pour UT, et approximativement 2000 pour TI, une différence attribuable au plus grand nombre de paramètres et une plus grande gamme de valeurs pour ces paramètres. Cependant, comme TI produit des trajectoires près de 50 fois plus rapidement que UT, dans les deux cas la procédure a été complétée en moins de 24 heures.

Des essais comparant l'optimisation avec et sans les paramètres affectant l'homographie ont montré que, comme attendu, la modification de l'élévation du plan de projection permet d'améliorer la précision des trajectoires. De plus, lorsqu'appliqué à TI (qui utilise les coordonnées projetées au cours du suivi des piétons), ces paramètres ont aussi permis une amélioration de MOTA d'entre 5 et 20 % selon la scène.

Finalement, les trajectoires produites lors de l'application des paramètres calibrés par TrO-

Ped sur l'entièreté des données vidéos ont été visualisées et analysées. Les cartes de densités relatives ainsi produites confirment que ces trajectoires représentent adéquatement le comportement piéton de chaque scène. De manière semblable, l'analyse des distributions de vitesses est en accord avec la littérature et avec les phénomènes observés, et les comptages directionnels automatisés — bien qu'erronés sur le même ordre de grandeur que les trajectoires — demeurent représentatifs des volumes relatifs réels.

La limitation principale de ces travaux est l'utilisation de seulement deux outils de suivis, qui n'ont pas été conçus spécifiquement pour le suivi de piétons. Ainsi, bien que les résultats obtenus démontrent une amélioration importante sur la calibration manuelle, la performance demeure inférieure ou égale à ce qui a été publié dans la littérature sur le suivi des piétons. Des travaux futurs devraient donc se concentrer sur l'essai de TrOPed sur des outils plus spécialisés, ce qui permettrait de vérifier si les améliorations obtenues ici se généralisent. Si tel est le cas, et des MOTA de 0.90 ou plus peuvent être régulièrement atteints, la collecte de données automatisées sur les piétons dépasserait enfin la performance de la collecte manuelle à un cout nettement moins élevé.

ABSTRACT

Though a wealth of data exists for the characterization of pedestrian movement, a majority of it originates from experimental settings owing to the current state of trackers for real-world scenarios. While these trackers are steadily improving, they remain insufficiently reliable for the accurate, microscopic tracking of individuals, particularly in cases of occlusion or higher density, complex scenes. In this work, the use of evolution algorithms is proposed for the systematic calibration of the parameters of existing trackers in order to further optimize their performance – evaluated by tracking accuracy and precision metrics – in complex cases, with an initial focus on two tracking methods designed for multimodal analysis. This calibration is further aided by the inclusion of additional parameters regulating homography, or specifically the plane to which tracker detections are projected. Three real test cases were used: a) a confined corridor in a public building, b) a subway station entrance during morning rush hour and c) a crosswalk in downtown New York. Results demonstrate a halving of tracking errors over both default and manually-calibrated parameters, as well as a strong correlation in performance between similar cases. These results were consistent over multiple trials and regardless of the starting parameters, strongly implying that the obtained solutions are indeed the global maxima for each scene. For application and validation of the resultant tracks, flow characterization and directional counting are demonstrated, utilizing tools included in the optimization framework.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

TrOPed	Tracker Optimizer for Pedestrians
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
TI	Traffic Intelligence
UT	Urban Tracker

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

Walking is both the most ancient and the most ubiquitous of transit modes. All trips at the very least begin and end with pedestrian locomotion, and inter- (as well as intra-) modal transfers necessarily implicate additional pedestrian phases when they take place. Consequently, high-density areas - and transit hubs in particular - benefit greatly from designs emphasizing pedestrian movement, both in their role as feeder systems for other modes and to ensure adequate evacuation in the case of an emergency. More broadly, such designs may be applied to increase the attractiveness of walking as a larger part of commutes, and to both evaluate and maximize the profitability in the commercial sector ; indeed, the latter is a substantial and growing area of research, particularly in supermarkets (see Larson *et al.*, 2005).

Optimizing spaces for pedestrian use requires the capacity to accurately and reliably model their behavior, and to predict their movement patterns in response to obstacles, events, distractions, each other, or in the absence of any such factors. Said capacity, in turn, must be built on a thorough understanding of pedestrian behavior in a variety of contexts.

Unfortunately, observing pedestrians' movement in sufficient spatiotemporal resolution to build and calibrate the aforementioned models has proven to be a difficult problem : the most accurate methods for individual - or microscopic - tracking are currently inherently limited to experimental (or at best, very specific) settings. More generalizable methods, in contrast, demonstrate considerably more errors upon visual validation, though their performance has been steadily improving over the last decade.

In an effort to bridge this divide, this work presents a generalizable evolutionary algorithm for the calibration of video-based tracking methods in order to improve both their accuracy and precision, in essence revealing any untapped potential a tracker may possess. Such algorithms have been applied to pedestrian tracking once before : Pérez *et al.* (2006a) applied evolutionary optimization to a single stage of their tracker, attaining a 25% reduction in positional error. In contrast, instead of targeting specific facets of the problem, the method presented herein aims at optimization of the entire tracking method. While the focus here is on extracting trajectories from video data, this method should be applicable to any microscopic trajectory-extraction method.

1.1 Context

The need for improved pedestrian models and design is steadily growing. While historically this has been the result of rapid urbanization since the industrial revolution (with similar trends more recently in developing nations) the 21st century has seen the demand for pedestrian analysis compounded through the global favoring of higher capacity structures and municipal desires to consolidate urban populations around existing transit networks (for example, City of Montreal, 2002).

Of course, existing designs are not bereft of pedestrian modelling, including both macroscopic and microscopic models. The former are advantaged by the fact that data collection for macroscopic behavior (i.e. the movement of pedestrians at the level of corridors and/or other subdivisions of an area, as opposed to that of each individual) can be performed through established methods. These methods include crowd size estimates, counts (manual, by RFID or with turnstiles), infrared detectors, pressure pad sensors and origin-destination surveys. In contrast, microscopic models (at least, those that are not proprietary) lack proven data-collection methodologies, being validated through one of three methods :

- **macroscopic data** : while microscopic models should be capable of reproducing macroscopic observations, using these as the sole means of both calibration and validation defeats the purpose of increased model resolution, making the influence of smaller design elements difficult to distinguish within the studied area as a whole.
- **manually obtained data** : establising pedestrian trajectories manually (habitually from video data, given that doing so on the scene is a very difficult task) is a reliable, accurate, but extremely time consuming process ; model validation with such data is therefore habitually performed with relatively small datasets (for example, Robin *et al.*, 2009).
- **data from experimental settings** : given the added control afforded to researchers in the experimental setting, the extraction of accurate pedestrian trajectories using either of the above methods is greatly facilitated. However, whether the data gathered in these situations is truly representative of real cases is questionable.

A fourth potential validation method exists in microscopic data extracted from real cases. It has however yet to achieve widespread use, a fact which can likely be attributed to its current poor performance in comparison to the methods presented above. Of the potential microscopic methods, those based on typical video data (as opposed to infrared, thermal or binocular video) have several notable advantages. First, the required equipment is already in widespread use for other purposes (e.g. surveillance) ; this ubiquity means data for emergency or other particular situations can be made available. Second, where the equipment is not already in place, its low cost and availability makes installation simple. Third and finally, the nature of the data makes visual validation relatively easy. The latter point is of particular importance in the method presented in the present work, yet all contribute to its generalizability.

1.2 Problem Statement

The primary problem addressed by this work is the need for increased performance in pedestrian tracking, coupled with the current absence of explicit calibration of tracking tools. This problem can be subdivided into the four following difficulties :

1.2.1 Complexity of Pedestrian Movement

Over the last two decades, video tracking of road vehicles has made great strides ; though it cannot yet be considered a solved problem, vehicle trackers have demonstrated the capacity to generate trajectories with both robust accuracy and precision (see Mei and Ling, 2011, for example). However, while one might expect this progress to translate more or less seamlessly to pedestrian tracking due to the superficial similarity of the tasks, several characteristics of pedestrians and their movement render their tracking substantially more difficult.

First, pedestrians make up a particularly heterogeneous group. In contrast to road vehicles, for which vehicle attribute and driving behavior are regulated and enforced, there are no restrictions on who may be a pedestrian and they are, for the most part, free to travel at the speed and via the paths they desire. Their behavior is also subject to a larger number of influent factors. These include those equally influencing motorists, such as age (Himann *et al.*, 1988) and alcohol use (Oxley *et al.*, 2006), as well as additional attributes such as physical fitness (Schlicht *et al.*, 2001) and trip purpose (Hoogendoorn and Bovy, 2004). One famous example perhaps best illustrating pedestrians' sensitvity to outside factors is that of Bargh *et al.* (1996) : exposure to words stereotypically associated with age - for example *prune* instead of the the control word *apple* - was found to significantly slow walking speed immediately afterwards (though it should be noted that the true source of this effect has since been disputed). More obviously, there is the sensitive case of those with visual, physical, or other impairments. Though it would certainly be extremely impractical (and perhaps unnecessary) to include all such factors in a single model - particularely given that some, such as fitness, may be impossible to observe - ensuring their adequate representation in an experimental setting would be a substantial undertaking.

Second, pedestrian movement is far less restricted than that of vehicles. Vehicles are constrained both to a small number of permitted paths at any given moment - as delimited by signals, speed limits, lanes, and safe distance from other vehicles - and to a limited range of motion (notably, a stopped vehicle can begin moving only forwards or backwards). Pedestrians, however, are constrained primarily by their own physical ability and whatever obstacles may be in their path. They may also form groups and move together or interweave when crossing paths, exacerbating both the complexity of their movements and the third problem with their tracking.

Indeed, even if the two preceding factors could somehow be negated, human beings remain difficult targets to track. The safe distance maintained between vehicles, combined with their rigid bodies, both allows their apparent shapes to remain relatively constant and limits instances of occlusion. Both these factors greatly facilitate tracking, as they generally result in more isolated and consistent targets. Pedestrians, on the other hand, implicate their entire bodies in locomotion, changing shape with each stride or during any other action they take as they walk. They also occupy much more space vertically than in the horizontal plane within which their movement occurs. When combined with their greater propensity to move in close proximity to each other, this lends itself both to occlusion of those individuals farther from the camera and to greater difficulty distinguishing individuals within a group.

Fourth and finally, one must consider how the above issues are compounded by the heterogeneity of pedestrian areas themselves. It is highly unlikely that pedestrian attributes in an office building are as varied as within a shopping mall, that their movement is as chaotic as in a secondary school, or that their flow is as dense as in a rush-hour transit hub. As a result, both the specific challenges and overall difficulty faced by a pedestrian tracker can vary wildly between cases ; performance in one case may not be particularly indicative of that in another.

1.2.2 Evaluating Extracted Data

Evaluating said performance is, in itself, problematic. No single, standardized metric exists (see section 2.3) making comparisons between trackers difficult, and yet some basis for comparison is necessary for one to proffer a method as an improvement upon another.

Fundamentally, a video-based tracker must execute two tasks : it must detect and track the objects of interest, as well as locate them within the search space. Tracker performance therefore globally consists of two factors, *accuracy* and *precision*, each relating to one of these tasks. Accuracy refers to the ability to correctly detect targets within the observed space, and to maintain that detection as the target moves. Precision is a function of the error in the targets' location, or how well the tracker locates the objects it detects. Both these measures must be adequate for the resulting tracks to be of use.

Unfortunately, within an optimization framework (such as that proposed here) evaluating performance via two inherently different dimensions is problematic, as the resulting performance measures are mathematically incomparable. Accuracy and precision must therefore somehow be fused into a single measure or, alternatively and if possible, optimization must take place for each one in sequence.

Each metric must also individually be calculated in a robust and consistent manner. What constitutes an accurate detection must be defined, and should exclude nonsensical track-object matching (e.g. associating the movement of branches in the wind to a passing pedestrian) while not implicitly imposing excessive precision requirements by necessitating perfect matches. Both measures should be consistent with the subjects being tracked and the applications being considered, particularly in regards to scale : given the low camera angles and small distances involved in much available pedestrian video data, the effect of perspective renders distances in the video-frame a poor substitute for those in the observed space.

1.2.3 Optimization Methodology

Improving performance as measured by the above metrics has been the subject of research for a number of years ; a brief overview of these efforts is presented in section 2.2. The primary approach, however, has been the development and fusion of novel methods and not the optimization of existing ones, for which calibration (though rarely reported) appears to be performed manually

This suggests that further optimization could be beneficial, but also signifies that the search space for many - if not all - trackers is largely unexplored. Said space is also usually broad and complex, as trackers have a tendency towards having a large number of parameters, many highly sensitive. The selected optimization method must therefore be rigorous. However, the only means by which to test a given set of parameters is through the tracker. The best performing of these run in real time (some an order of magnitude slower) and a test sequence must be sufficiently lengthy in order to be representative of the scene as a whole and so avoid overfitting of the tracker. The optimization method must therefore converge to a solution relatively rapidly lest the process take an inordinate amount of time.

1.2.4 Extracting Meaningful Data

One final consideration, related to that presented in 1.2.2, is that the trajectories output by the tracker must represent the *ground-truth* (or real trajectories of the pedestrians) in such a way as to be meaningful in subsequent analysis. This can, in part, be evaluated by accuracy and precision - indeed, if both are perfect, the data is by definition perfectly representative. In the inevitable case that errors do occur, however, these metrics provide only a partial portrait of their gravity.

Figure 1.1 presents three simple examples : though the two rightmost columns are extremely similar in terms of estimated accuracy and precision, ground-truth in the center column can relatively easily be extrapolated either by human verification or simple heuristics in post-processing. In contrast, the tracks in the column on the right are ambiguous, potentially misleading. And yet, all the presented tracks would perform on par with the leading contemporary metrics, with accuracies ranging between 80 and 90 percent according to most metrics.

This behavior is troublesome, especially during optimization, as the parameter sets leading to two such ostensibly similar solutions are unlikely to be similar themselves, but instead may represent distinct local maxima - a phenomenon equally likely to present itself, albeit more discretely, at the lower performances encountered early in the optimization process.

1.3 Objectives

The objective of this project is the development of an optimization framework for the improvement of video-based pedestrian tracker performance. More specifically, the approach described herein targets the oft-ignored calibration of tracking tools, particularly in regards to optimization for specific scenes. Said framework aimed to fulfill the following criteria :

- **Significant improvement over existing calibration methods** : The optimization can only be deemed successful if it attains marked improvement over previous tracker calibration methods (manual calibration, for the trackers examined herein).
- **Minimal dependence on starting parameter values** : In order to both maximize automation and ensure the provided solutions represent global maxima, optimization should be independent of the input tracker parameters.
- **Efficient convergence time** : Given the problems stated in 1.2.3, it is likely that many optimization algorithms result in convergence times best measured in weeks or months - comparable to what would be required to simply extract the data manually, though of course this would be much more time-consuming for the researcher. Minimizing run-time should therefore be a priority.
- **Generalizability** : The framework should be applicable to a large number of different trackers, regardless of their methodologies or characteristics. Said trackers are not limited to pedestrians ; indeed, while pedestrians present a unique combination of tracking difficulties,

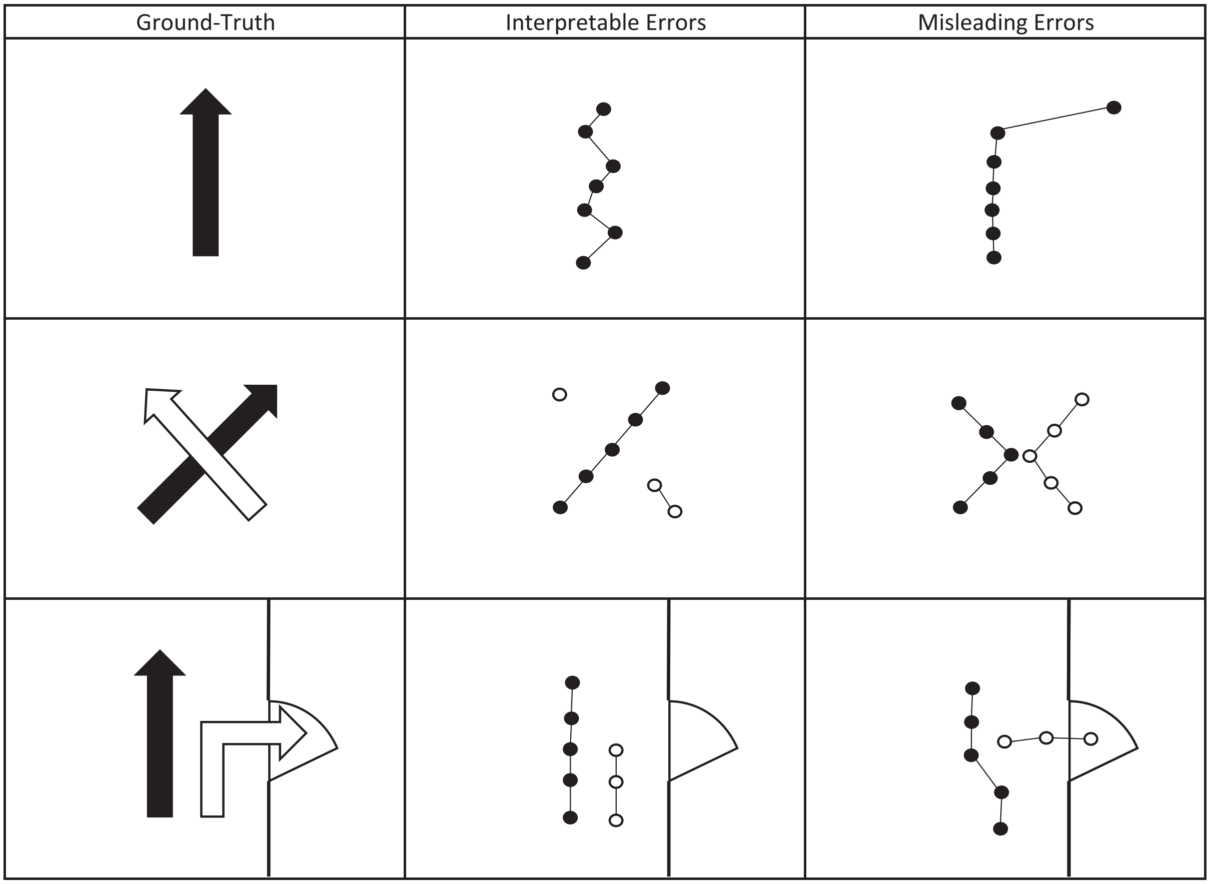


Figure 1.1 Examples of interpretable and misleading errors for near-identical accuracy and precision, as measured by the CLEAR MOT metrics described in section 2.3.2. Each circle represents the detection of an object in a single frame.

tracking methods are nearly universal regardless of the target objects. Furthermore, as all video analysis is to be performed by the optimized tracker, the method presented herein is also not limited to video tracking, but to all automated tracking applications.

1.4 Document Structure

The present document consists of seven sections. The present section, the introduction, serves to present the overall subject matter, as well as the primary research objectives and expected obstacles. The literature review follows, summarizing both the state of the art in pedestrian tracking and modelling, as well as other research topics pertaining to the present subject. The third section presents the fundamental methodology of the developed software, both as it functions as a whole as well as the individual underlying processes. The document continues with the presentation of the results of optimization with the constructed tool in section five, followed by examples of the extracted data and their comparability to data extant in the literature in a sixth section. It concludes with a general overview of the results, as well as discussion of the observed limitations of the project and potential avenues for future work.

CHAPTER 2 LITERATURE REVIEW

This section presents an overview of the state of the art of pedestrian models and data-gathering methods, as well as potential optimization algorithms and video-tracking metrics.

2.1 Pedestrian Models

Pedestrian models serve to represent and predict walking behavior as it occurs in reality. They achieve this by simplifying said behavior into a set of mathematical heuristics, which are subsequently calibrated using available real-world or experimental data. Pedestrian modelling is generally performed on one of three scales (Sahaleh *et al.*, 2012) :

- **microscopic or disaggregate models** : Also sometimes refered to as agent-based models. At this scale, each pedestrian is simulated individually, with the movements and actions estimated independently of other agents. Every individual's position is established according to a preselected unit-time while they are in the simulated area.
- **macroscopic or aggregate models** : Analysis of pedestrians as groups, with individuals existing only as members of an aggregate. System state is generally described by the density, flow and average velocity of groups ; individuals are not distinguishable from each other. As such, these models are particularely reliant on accurate fundamental diagrams, which define velocity as a function of density (Schadschneider *et al.*, 2009).
- **mesoscopic models** : A compromise between the previous two scales, mesoscopic models simulate pedestrians in aggregate terms. They can, however, render a microscopic portrait of an area, albeit probabilistically (Teknomo and Gerilla, 2008).

Regardless of scale, a complete theory of pedestrian dynamics usually takes into account three levels of behavior (defined in Hoogendoorn *et al.*, 2002) :

- **Strategic level** : At the highest level, pedestrians decide what activities to perform, as well as where and when, with no knowledge of the network or potential routes. This information resembles (and is possibly best represented by) origin-destination surveys. Indeed, within a given simulation, this data serves as an input parameter, and tends to be either observed in or extracted from such surveys before being utilized in the following stage.
- **Tactical level** : At the tactical level, individuals take the network into consideration so as to decide upon a particular route. They make decisions based upon factors including geometry, obstacles, signs, and the general macroscopic behavior of other users (velocities, densities, etc.) in order to select an ideal, optimal path through the network. Given the

typically restrained scales of pedestrian simulations, there may be some interplay between this and the previous level : the tactical decision to take the subway, for instance, may lead to the strategic decision to use the station entrance closest to one's home, and to purchase a snack before heading directly to the train.

– **Operational level** : This level describes the actual walking behavior of pedestrians : acceleration, avoidance of obstacles and of other individuals, and potential distractions (e.g. window shopping in a shopping mall). In essence, it is at this level that pedestrians make the immediate decisions and movements to accomplish the objectives set previously.

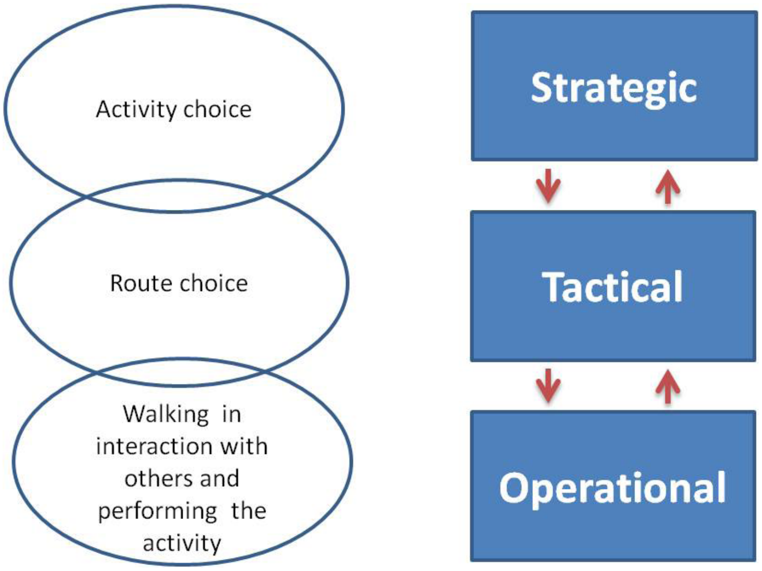

Activity choice
Route choice
Walking in interaction with others and performing the activity
Strategic
Tactical
Operational

Figure 2.1 Different levels in pedestrian walking behavior. Source : Sahaleh *et al.* (2012).

Most pedestrian models discussed below are principally focused on the operational level, with shortest-path calculations taking the place of the more complex thought processes implicated at the tactical level, and strategic-level information being input from observed or extrapolated data, as stated above. While the ability to fully model the higher levels of behavior would require an understanding of human decision involving additional disciplines (e.g. psychology and sociology) a sufficiently large pedestrian dataset would be required to either simulate or truly begin their fuller integration into current models.

2.1.1 Gas particle model

Perhaps the earliest model of pedestrian dynamics is Henderson's gas particle model (Henderson, 1974). Based on analysis of the pedestrian velocities of college students and children on a playground, this model equates pedestrian motion through restricted passageways to that of an ideal gas. It was built upon the observation that human velocity distributions in both studied cases fit the Maxwell-Boltzmann distribution for the velocities of ideal gases for a given temperature, and therefore a Gaussian distribution within a given crowd whose "temperature" is assumed uniform.

Like the ideal gas laws on which it is based, this model is macroscopic, as it focuses on the general movement of and within a mass of particles/pedestrians. It was validated by the author, though (unsurprisingly) only in cases where the studied group's homogeneity most resembled that of an ideal gas : individuals predominantly of the same sex and similar activity (running or walking), age, size and fitness (the latter three attributes, though not explicitly controlled for, are ensured by the chosen cases). The model is also constrained to the specific cases of restricted, high-density movement in channels, within which the analogy's limitations - gas particles do not exhibit agency, and pedestrians are not obligated to obey the laws of conservation of energy or momentum - are less evident.

2.1.2 Fluid dynamics model

A later fluid-mechanics approach to pedestrian modelling (Helbing, 1998) attempted to reconcile the noted differences between particles and pedestrians while still highlighting the similarities observed in their flow patterns. Fundamentally, it consists of the same macroscopic fluid dynamics laws and equations as utilized by Henderson, however modified so as to account for certain microscopic pedestrian phenomena :

- Interactions between "colliding" pedestrians are anisotropic : both individuals will not necessarily be affected in the same manner.
- Pedestrians tend to approach their desired velocity, outside forces permitting.
- Individuals also have a preferred direction : towards their destination.
- Systems can lose or gain density, via entrances and exits.
- Reaction time of individuals plays an important role in their interactions, particularly in propagation of movement within crowds.

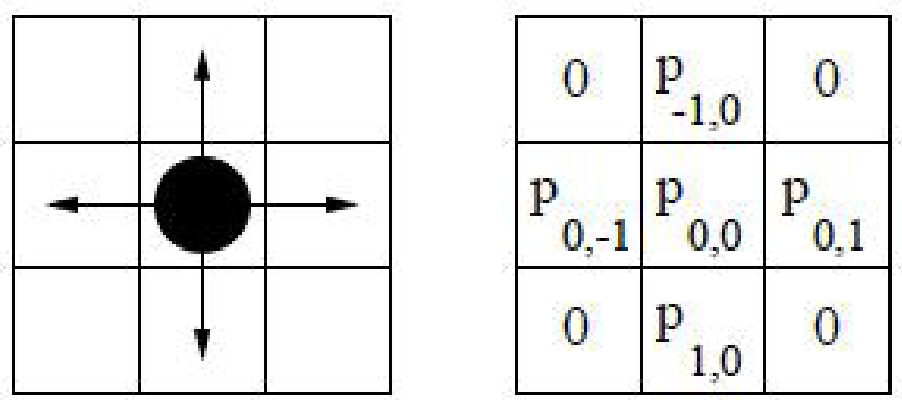
Though the resulting equations greatly resemble those of ordinary fluids, they result in some interesting and realistic emergent behavior. For instance, a crowd's prevailing tendency to avoid obstacles to either the left or right (itself accounted for by the probability density function of their preference) leads rapidly to the development of lanes of opposing flow. Similarly, the inclusion of crowd heterogeneity and reactions times lends itself to realistic depictions of pedestrian jams and increased chances of collisions in critical situations, respectively.

Despite the added emphasis on individual interactions between pedestrians, crowds are still defined by densities and average velocity, much like the previous model. As such, the fluid dynamics model remains purely macroscopic, and shares the problems involved in modelling low pedestrian densities.

2.1.3 Cellular automata

Cellular automata models implicate two titular entities : cells, which are pre-delimited two-dimensional spaces subdividing the modelled area and which have rules defining occupancy and possible directions of travel, and automata, entities which seek to move between cells according to preset instructions. Through these rules, the models attempt to include the psychological factors regulating pedestrian behavior in a more seamless manner than fitting them to existing equations, as done in the previous models.

The use of cells forcibly discretizes both space and time : pedestrians can only be located within a cell (which in turn can only fit a single individual) and the smallest meaningful unit of time is therefore the shortest time required to moving between two adjacent cells. At each time step, the new location of each pedestrian is calculated in parallel, the probability of entering a given adjacent cell being a function of the rules within a von Neumann neighborhood (Blue and Adler, 2001, see figure 2.2).



0	$P_{-1,0}$	0
$P_{0,-1}$	$P_{0,0}$	$P_{0,1}$
0	$P_{1,0}$	0

Figure 2.2 A pedestrian, its possible directions of motion, and corresponding probabilities for the case of a von Neumann neighborhood. Source : Sahaleh *et al.* (2012).

Such models originally had relatively simple rules, largely defining impossible motion (e.g. pedestrians may not walk through each other unimpeded, though they may exchange places) and some fundamental elements of pedestrian motion (e.g. side stepping, preferred speed and collision avoidance). It has since been expanded to include additional behaviors, most notably the ability of particles to move opposite their preferred direction if necessary (Weifeng *et al.*, 2003), to move obliquely (Yamamoto *et al.*, 2007) and to consider data beyond their immediate vicinity (Burstedde *et al.*, 2001).

It should be noted that while these models are built solely on microscopic interaction and movement and are near-universally considered microscopic models, they have only been validated on macroscopic scales.

2.1.4 Mesoscopic models

Mesoscopic models stem primarily from the desire to improve upon the computation times involved in microscopic simulation, without sacrificing the ability to account for individual pedestrians (vital in the accurate evaluation of evacuation times). Indeed, early mesoscopic models (such as Hanisch *et al.*, 2003) were aimed at short-term planning and safety in public buildings.

Such early efforts achieved their goals by first simplifying the simulation area into a network of links and nodes (see figure 2.3). The nodes represent either entrances, exits, stations (where pedestrians must wait to be processed, e.g. a ticket booth) and storage areas, which may be stairways, elevators, hallway intersections, or simply rooms. Pedestrians exist individually within nodes, but are joined into homogeneous groups in order to move between them ; they are again free to leave and join new groups at subsequent nodes. In short, obstacles are modelled by limiting flow out of nodes, and travel times are represented by link length.

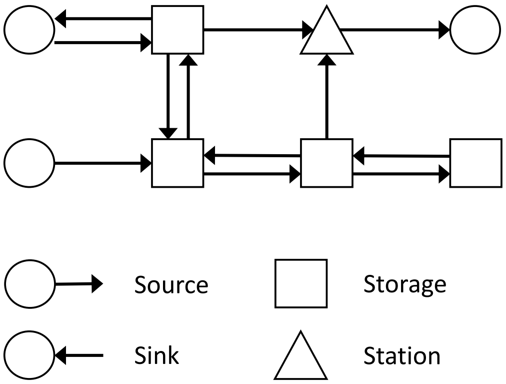


Figure 2.3 Example network used in early mesoscopic models. Inspired by Hanisch *et al.* (2003).

Hanisch's model is adequate if pedestrian flow is controlled primarily by bottlenecks connected by higher capacity corridors, as is the case when movement is predominantly unidirectional. It however neglects interpedestrian interactions beyond those in queues, and so cannot readily be applied to more complex cases.

A more generalizable mesoscopic model is that of Teknomo and Gerilla (2008). It fundamentally resemble cellular automata models, in that the modelled area is decomposed into a lattice of cells, with individuals selecting adjacent cells to move to, much as in figure 2.2.

However, the cells are larger - one to three meters wide - and can therefore accommodate multiple pedestrians. Time to traverse a cell depends on both the trajectory (the source and destination cells) and its current population.

The need to identify and avoid individual collisions is thus elegantly circumvented (at the expense of detailed visualization) but is replaced with a heavy reliance on the accuracy of the fundamental diagram used. A single diagram was utilized in Tekmono *et al.*'s original paper ; however, fundamental diagrams have been observed to vary by culture (Chattaraj *et al.*, 2009), activity, and type of flow (unidirectional, opposing directions, crossing at various angles, merging, etc.) (Zhang *et al.*, 2011). A greater understanding of the fundamental diagram is hence needed if this model is to more accurately reflect real, complex movement.

Qiu and Hu (2013) developed a more fluid, spatial activity-based model. The approach borders on being fully microscopic, as all pedestrians make decisions individually. However, in contrast to the microscopic models presented below, where decisions are made at every time step, decisions in Qui and Hu's model are made only when a pedestrian moves a threshold distance from (or demonstrates significant activity since) the last decision. Between decisions, pedestrians close to one another and with similar directions are grouped together, with characteristics of the group being defined by those of the individuals within it. The model may therefore account for pedestrian heterogeneity in desired speed while maintaining computational performance on par with other mesoscopic approaches.

2.1.5 Discrete choice models

The models discussed thus far were all majoritarily calibrated using macroscopic data ; their microscopic elements (e.g. pedestrian interaction) were designed through observation and adjusted to fit said data. Noting this, Antonini *et al.* (2004) devised the discrete choice model based solely on microscopic data, established manually from video recordings.

In this model, utilized by the SimPed simulation tool (Daamen, 2002), a pedestrian at a given time t is assumed to make two choices regulating his position at $t{+}1$: one regarding speed (accelerate, decelerate, or maintain the current speed) and the other regarding direction (keep the current heading or turn at predefined, discrete angles). The potential positions resulting from these options form a cone, demonstrated in figure 2.4.

Possible positions are described by several attributes, dependent on proximity to the destination, the presence of obstacles, and both the position and direction of other pedestrians. The pedestrian herself is defined by desired speed and its elasticity, or willingness to diverge from the said speed. Finally, a random variable is implemented in order to capture any otherwise

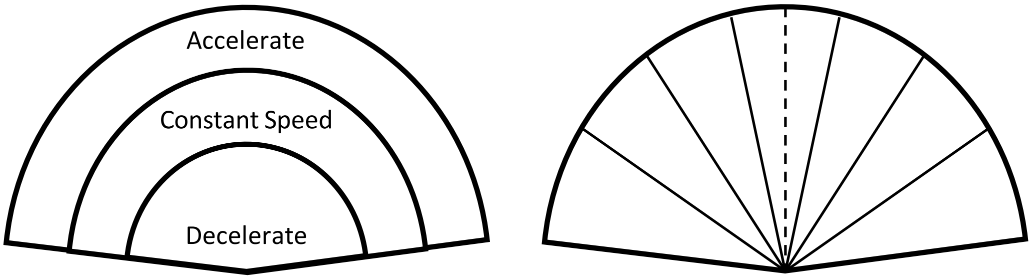


Figure 2.4 Discretization of space based on 3 speed regimes and 7 radial directions. Inspired by Antonini *et al.* (2004).

unconsidered factors. The probability to enter a given space is defined by the utility function of these attributes, forming a behavioral nested logit model for the operational level of pedestrian movement, which is then fitted to the microscopic data. A separate and similar - albeit simpler - model is used to determine paths at the tactical level.

A notable strength of this model is its capacity to accommodate additional variables, and thus take into account factors beyond those stipulated in the original formulation. Indeed, further work has sought to include variables such as visibility (Guo *et al.*, 2012) and density (Asano *et al.*, 2010) as well as improve the ability of simulated agents to identify optimal routes when impeded by other pedestrians (Kretz *et al.*, 2011, notable for using virtual reality to generate a variety of cases for the single test subject). Unfortunately, the primary weakness of logit models is the reliance on large quantities of accurate data ; most (if not all) work on pedestrian discrete choice models has been built on relatively small datasets, most of them experimental.

2.1.6 Social force model

Where the discrete choice models ascribe route-choice to pedestrian agency, the social force model (first described in Helbing and Molnar, 1995) describes pedestrians as passive entities subject to attractive and repulsive forces in their environment. Consequently, a pedestrian's movement in the model is defined by their acceleration, given by Newton's equation :

$$\frac{d\overrightarrow{\nu_\alpha}}{dt} = \overrightarrow{F}_\alpha(t) + fluctuations \tag{2.1}$$

The *fluctuations* term denotes random, unsystematic variations in behavior, representing the fact that pedestrians rarely move in perfectly straight lines. $\overrightarrow{F}_{\alpha}(t)$ denotes the force affecting the pedestrian at time t and is the sum of the individual, eponymous social forces.

As stated earlier, these forces can be either attractive or repulsive. The former are most prominently exerted by the pedestrian's destination, though they may also represent objects that will facilitate their journey (such as escalators or signs) or even elements that simply attract attention (e.g. a store display or vending machine). It is therefore possible for the model to account for operational and tactical decisions simultaneously, potentially capturing a realistic, less mechanical set of behaviors : a simulated agent may decide to purchase a coffee before standing at the portion of the platform closest to a television screen, despite the only input order being "board the next train".

Similarly, the repulsive forces mostly define collision avoidance, being exerted by the nearest walls, by obstacles and by most other pedestrians (friends and street artists being possible exceptions). The use of distance-dependent forces in lieu of physical limits allows for microscopic behavior more in line with common observations : for instance, when crossing another pedestrian in a hallway, one tends to maintain some near-equal distance from both the crossing person and the wall, despite this generally being a slightly longer path than is strictly required.

Evidently, the forces have additional differences. It makes little sense for attraction to the primary destination to vary markedly over time, and it is equally far-fetched to expect an individual to be much distracted by a display too distant to discern. A pedestrian is also less likely to concern herself with the current position of another than with the predicted position at the time of closest approach, and may realistically assume the other will similarly adjust their course (Lakoba *et al.* (2005) and Zanlungo *et al.* (2011)). Together, such considerations call for unique formulations for each types of force, additionally variable between individuals in a given population. Consequently, substantial calibration has been required.

At the macroscopic scale, the social force model has been calibrated and validated in a number of cases (Kretz *et al.* (2008), Beutin (2012)) and is in widespread use for pedestrian simulation, being the core of both VISSIM's VISWalk and SIMWALK. However, beyond confirming a "natural [...] look and feel of the individual agents" (Kretz *et al.*, 2008) it has not been microscopically validated. Hopefully, as with the other models present here, the availability of additional, real microscopic data will allow further calibration.

2.2 Data collection methods

2.2.1 Point and line data-collection methods

As can be infered from section 2.1, the majority of published pedestrian flow data is macroscopic. At this scale, the information of interest is speed, density, and volume, at and between points or linear thesholds in the studied network. The simplest and earliest methods were, unsurprisingly, manual, performed by tally sheet or mechanical or electronic count board. Henderson (1974) collected students' velocities with only a stopwatch and knowledge of the distances travelled. Seyfried *et al.* (2005a) experimentally studied the density-velocity relationship in single-file queue movement, recording when each individual crossed two predefined screenlines (see figure 2.5). Volume is still routinely collected via manual count, as is density provided an aerial view is available. These methods are simple and reliable, particularly when performed with recorded data (field observers have been found to underestimate volumes by up do twenty-five percent ; see Diogenes *et al.*, 2007) but are costly and cannot easily be employed over long periods of time. Unfortunately, in the cases of density and velocity, manual data collection currently remains the only macroscopic measurement method.

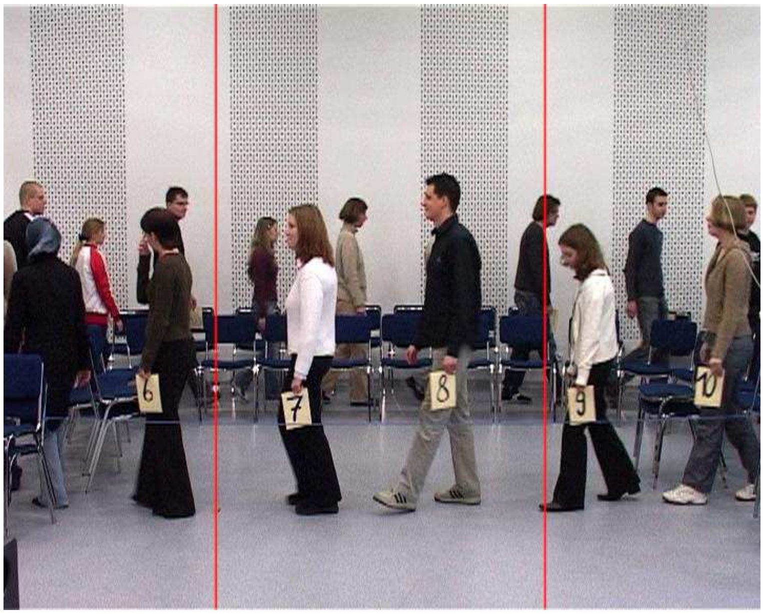


Figure 2.5 Example of a macroscopic pedestrian study, with data collected manually from recorded footage. The time each of the two screenlines (the red vertical lines) is crossed by a pedestrian is recorded manually, allowing both velocity and density measurement. Source : Seyfried *et al.* (2005b).

In contrast to manually recorded data, automated methods can often be left in place near-indefinitely and at little cost beyond the initial investment. The most typically encountered example is the turnstile, increasingly networked and coupled with electronic transit passes. Though the collected data is often limited to the point of entry alone, in certain cases - most notably, transfer hubs in public transit networks - they can provide a near complete picture of flow volume for an area. The primary drawbacks are that this data is most-often privately owned, and thus seldom made publically available, and that the impediment caused by turnstiles makes their implementation solely for the purpose of counts catastrophically impractical.

A number of more workable pedestrian counting methods exist ; they are well summarized in Bu *et al.* (2007) :

- **Infra-red beam counters** : An IR emitter and receiver are placed on opposite sides of a walkway ; pedestrians interrupting the beam between them are counted. While inexpensive, these cannot differentiate between pedestrians and other obstructions, nor can they detect pedestrians occluded by others.
- **Passive infra-red counters** : Based on military technology, these counters passively detects the heat emitted by moving objects within four meters ; models with two detectors can also provide directional counts. They have a particularly high error-rate at higher densities (Kerridge *et al.*, 2004), but these errors are relatively systematic ; with adequate upward adjustments, the resultant counts provide a reasonable estimate of pedestrian volume (Greene-Roesel *et al.*, 2008).
- **Piezoelectric pads** : A piezoelectric pad is a simple sensor that emits a signal when sufficient pressure is applied. At low densities and when installed where direct physical contact is assured (e.g. a building entrance) they can provide excellent results, and some models even include timer systems to ensure a single count even if two steps are detected from the same individual. However, they are much less effective when multiple pedestrians cross simultaneously.
- **Laser scanners** : These detectors consist of infra-red laser range finders, which sweep a horizontal or vertical space to detect obstructions - functionally a 360 degree beam counter where the receiver is any static surface. Installed on a ceiling and used vertically, they can provide directional counts through a threshold as well as classify pedestrians by height. Installed at floor level (so as to minimize occlusion) they can detect and even accurately track pedestrians in a space limited primarily by line-of-sight (Zhao and Shibasaki, 2005). Their main disadvantage is their cost, far higher than the other counters in this list, due in large part to the complex signal processing requiring a dedicated processor.

One final automated counting method listed by Bu *et al.* is using computer vision. If counting is the sole objective, the number of pedestrians can be determined by detection of human-like shapes in still images (extracted or not from video data). While it is also possible (and indeed performed in section 5.1.4) to obtain counts from video data directly by applying video tracking and counting the resultant tracks, this is less a question of macroscopic data collection than one of aggregating microscopic data.

2.2.2 Experimental spatial methods

Experimental settings allow the researcher nearly complete control over the pedestrian environment. In addition to allowing the examination of specific circumstances and their effects on pedestrian flow, such settings greatly facilitate the accurate extraction of pedestrian trajectories by permitting a broader number of tracking methods than are available in the field. Said methods are too numerous and varied to exhaustively list here. However, they can be summarily categorized for the purposes of this study into video-based and non-video-based methods.

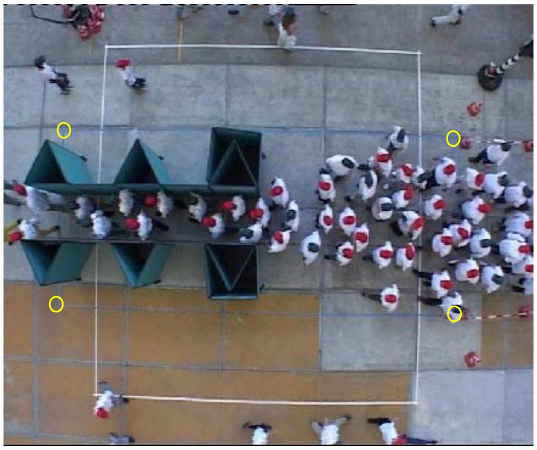


Figure 2.6 Microscopic video tracking using colored uniforms as visual cues. Source : Hoogendoorn *et al.* (2003a).

Generally, the simplest way to track pedestrians microscopically without resorting to video recording is to give them a tracking device. These can range from inertial sensors (e.g. Feliz Alonso *et al.* (2009), Foxlin (2005)) to the augmented-reality headset used by Kretz *et al.* (see section 2.1.5), but such devices are costly to provide to more than a handful of pedestrians. This may be circumvented by the increasing ubiquity of smartphones, and wearable technology may eventually allow more generalizable tracking through triangulation of the emitted wireless signals : Danalet *et al.* (2014) managed to accurately track the activity of a subject throughout a university campus from network traces. At present, however, these methods are limited to very specific applications : despite the increasing ubiquity of personal wireless devices, there is no guarantee any given individual will carry the device through every movement, nor that it will be set to transmit on the frequency used for tracking.

The difficulties of automated video-based pedestrian tracking outlined in section 1.2.1 can be partially mitigated in experimental settings. This is generally accomplished by simplifying the tracker's task, either by providing additional visual cues or by limiting instances of particular difficulty (e.g. occlusion). An example of the both is the experiments carried out by Hoogendoorn *et al.* (2003b), where the participants were provided with solid white shirts and either red or green hats (see figure 2.6) the two colors representing what instructions they were given : walk normally, slowly or aggressively. The tracker was therefore tasked with detecting predefined colors on white backgrounds, rather than the more complicated detection of "a moving object shaped like a human being".

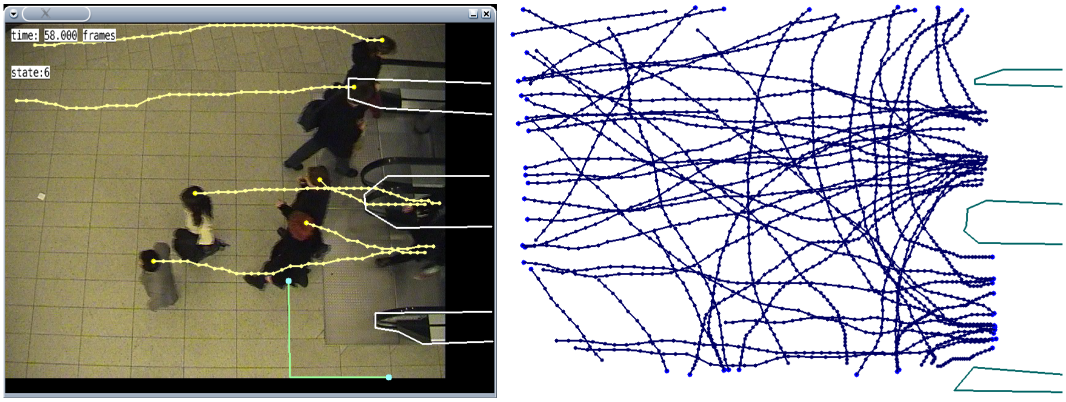


Figure 2.7 Real-world pedestrian tracking, using specially-installed overhead cameras and a close angle. Source : Johansson *et al.* (2007a).

The fact that Hoogendoorn *et al.* recorded from above also facilitates tracking, both by eliminating occlusions and by allowing points in the video frame to be converted directly to real-world coordinates. Indeed, use of this camera angle alone can allow for excellent tracker performance : Johansson *et al.* (2007b) collected ostensibly excellent trajectories (no accuracy or precision metrics were published) in real-world cases, using a relatively simple head-detecting tracker. The studied areas, however, are very small (see figure 2.7) ; while this is likely attributable to the limited vertical space available for camera installation, it provides higher detail for each recorded individual while limiting the effects of perspective.

The advantages afforded by these factors are made clearer when compared to the authors' later study, using this method to examine pilgrim flows towards the Holy Mosque in Makkah, Saudi-Arabia (Johansson *et al.*, 2008). In this case, the camera recorded movement over a far larger area, and was additionally aided by the white clothing and headwear worn by a majority of the pilgrims, contrasting with the darker color of the ground. Yet despite these fortuitous circumstances, the published example frame (figure 2.8) displays multiple instances of both misses and overdetection (again, tracker performance was not explicitly evaluated).

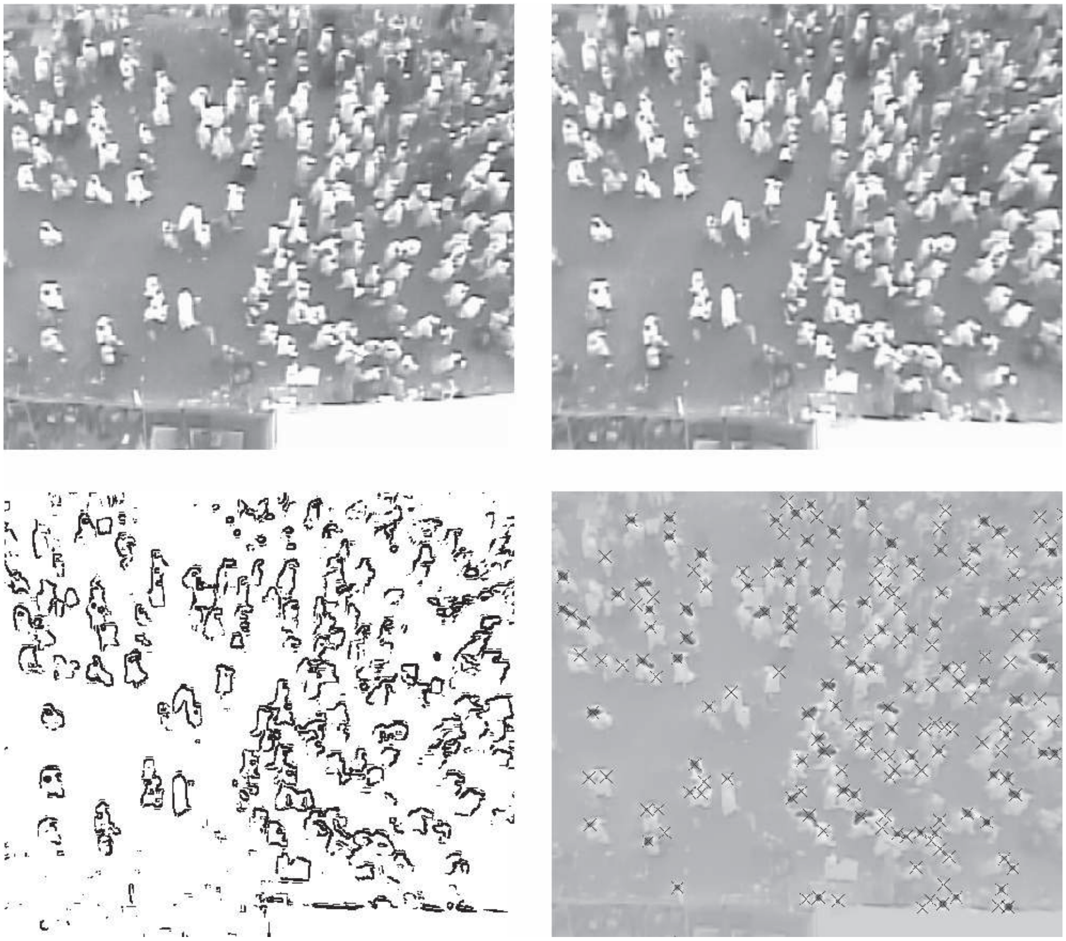


Figure 2.8 Illustration of the processing of an example frame used in the tracking of pilgrims near Makkah, Saudi-Arabia, from the original image (top left) to the resultant detections (bottom right). Source : Johansson and Helbing (2008).

These difficulties - the physical installation of equipment, the development and/or calibration of tracking software, and the tracking errors that remain despite the prior two - help explain why a number of studies utilize manual tracking from recorded video despite taking place in a controlled, experimental setting (e.g. Daamen and Hoogendoorn (2003), Isobe *et al.* (2004), and Kretz *et al.* (2006)). These problems are only compounded when typical real-world restrictions are in place.

2.2.3 Video-based methods applicable to real-world cases

Automated, video-based pedestrian tracking is a difficult problem. It must contend with the inherent complexity of pedestrian movement (described in section 1.2.1). As observed in the previous section, it must also contend with the heterogeneity of pedestrian appearance, and the occlusions associated with typically available camera angles. Furthermore, in real-world cases, there are a myriad of visual effects (e.g. variable lighting, shadows, lens distortion, non-human moving objects) which can confuse an automated tracker (Forsyth and Ponce, 2002).

Video-based tracking originated in the early 1980s. These first trackers relied solely on the analog output of the video camera, searching the signal for voltage spikes indicative of high contrast, which could then be located within the video frame (Noldus *et al.*, 2002). While decidedly clever, such trackers only functioned in cases where the desired target was guaranteed to be the highest source of contrast in the image, largely limiting them to specific applications in controlled settings (e.g. tracking an object through a maze). Due to software and performance limitations, they were also incapable of tracking multiple objects, a constraint which only began to be lifted with the widespread introduction of digitized video in the mid-1990s.

Since then, research interest in the field has greatly expanded, ranging in application from industrial automation to studies of the locomotor behavior of poultry (Sergeant *et al.*, 1998). In pedestrian tracking alone, both CLEAR and PETS (presented in section 2.3) hold annual evaluations of state-of-the-art trackers. The earliest applications to pedestrians attained accuracies ranging from 70 to 80 percent (as estimated from the published results, given that no standard metric then existed) but relied on very specific conditions : providing the tracker with an accurate and continuously updated ground-plane map (Remagnino *et al.*, 1997), manually identifying pedestrians to be tracked (Denzler and Niemann, 1997) and/or limiting the complexity of the tracked area to one similar to that examined by the earliest trackers (Masoud and Papanikolopoulos, 1997).

More recently, trackers have evolved to require little to no manual input, and to be free of the previously pervasive scene constraints. In the current generation of pedestrian trackers, MOTAs of over 0.80 have been attained, though accuracies in the 0.50-0.60 range appear to be more common (Ellis *et al.*, 2009). Still, such measures are difficult to objectively interpret : tracker performance is dependent not only on a tracker's attributes but also on scene complexity, a feature rarely prominent in their published evaluation.

It was noted in section 2.3 that automated tracking requires the performance of two primary tasks : detection and tracking. In addition, some trackers also integrate post-processing as a third phase, fine-tuning the generated trajectories according to hypotheses of pedestrian behavior. However, these phases are only a heuristic simplification ; some trackers perform multiple tasks simultaneously, or subdivide and reorder them according to their own particular methodologies.

Globally, individual tracker may perhaps better be classified by the underlying method utilized at the detection and tracking stages. The most common of these using static, monocular cameras are *feature-based tracking*, *background subtraction*, and *tracking by detection*. They are examined individually in the subsections below.

Feature-based tracking

Feature-based tracking consists of detecting distinct arrangements (or features) that move uniformly through the video frame. Sometimes refered to as corner-detection, it consists of following any group of pixels moving through the image-space with little or no relative change to one another (Maggio and Cavallaro, 2011). Pixels are interpreted not by specific color but by intensity. This reduces the features' sensitivity to changing lighting conditions, as well as making areas of higher contrast (e.g. the titular corners) the most readily detected (Tomasi and Kanade, 1991). In effect, it is therefore similar to the early voltage-spike based methods, albeit applicable to multi-target tracking. Traffic Intelligence (Saunier and Sayed, 2006), a tracker optimized in this research and therefore examined in its own section, is a feature-based tracker. An example of its application is presented in figure 2.9.

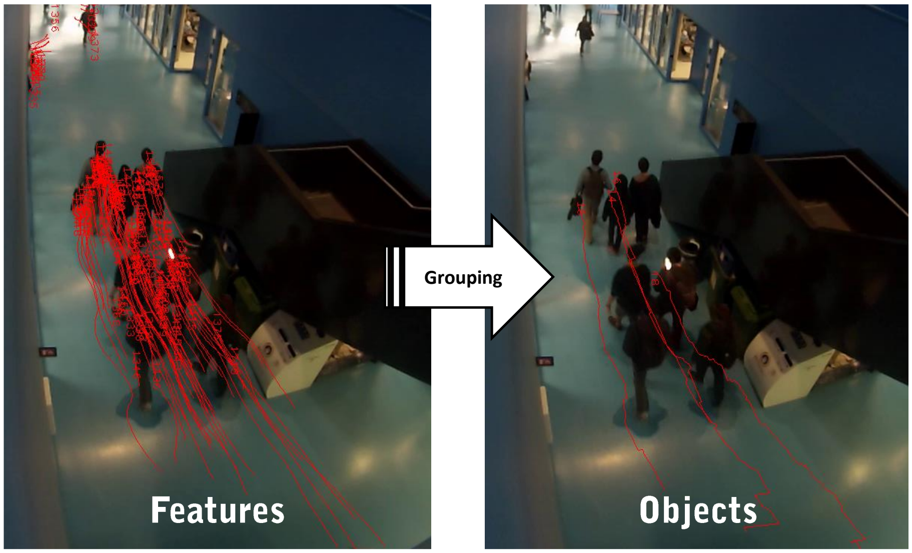


Figure 2.9 Example of feature detection and subsequent (mostly erroneous) grouping, as performed by Traffic Intelligence before optimization.

One weakness of feature-based trackers is their reliance on movement for detection. If a target temporarily ceases moving - for example, a pedestrian stopping to read a map - it will be lost by the tracker, only to be detected once it begins moving again, generally as a new target unless some post-processing intervenes. The small size of individual features also increases the likelihood of false positives, either by detecting small targets (e.g. a bird's shadow or litter carried by the wind) or even signal noise ; such behavior is typically filtered via adequate parameterization of both the grouping phase of the tracker and of what constitutes an acceptable feature. Within corner-detecting trackers, the latter concern is accounted for in the Kanade-Lucas-Tomasi (KLT) feature tracker. The most widely utilized such tracker, it is specifically intended to optimize tracker performance via classification and thresholding of detected features by quality, though at the cost of requiring increased parameterization (Birchfield, 2007).

Recent works have attempted to extend the fundamental tracking methods to better perform pedestrian tracking specifically. Rabaud and Belongie (2006) sought to train the KLT tracker for extremely dense (> 1 $pedestrian/meter^2$) cases. To do so, they optimized (through unreported methods) the tracker's window size, in which features are searched for from one frame to another. They also added a series of post-processing steps both to correct the trajectory fragmentation caused by mismatches and to extrapolate trajectories when targets were lost. This resulted in heavy restrictions to feature grouping, particularly in terms of the acceptable relative motion of features in a group, leading to a tracker which majoritarily detects the relatively static heads of its targets. Though the reported accuracy (measured by counts) was high (see table 2.1) it is likely to be substantially less in cases of more complex movement than the linear, bidirectional case tested.

Background subtraction

Background subtraction, as the name implies, relies on identification of the static background (or background model) of the video, and subsequent subtraction of said background in each frame in order to identify moving objects (Piccardi, 2004). Since it is unrealistic to expect real scenes to be initially empty, and seeing as the background is likely to change over time (e.g. as the sun moves accross the sky, or as vehicles park and depart) the background model must be continuously updated. This task can be performed by a variety of methods, but all consist fundamentally of extrapolating the current state of a pixel from the prior behavior of it or its neighbors.

The remaining pixel groups, or *blobs*, are then evaluated by the tracker in order to distinguish individual targets of interest. This is typically determined by size : pedestrians can be assumed to occupy a number of pixels first exceeding a certain minimum (under which the blob is likely noise or debris) and then, in the segmentation stage, under a given maximum (beyond which the target is either a vehicle or several grouped pedestrians). This process is far more

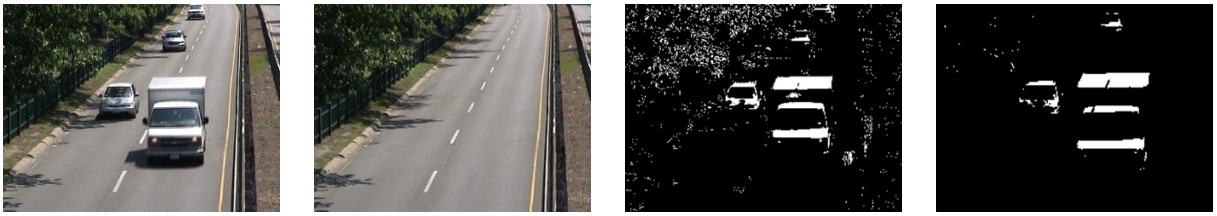


Figure 2.10 The four phases of background subtraction. From left to right : video frame, background model, foreground detection, and final foreground after segmentation and cleaning. Source : Jodoin *et al.* (2014).

computationally intensive than feature-based tracking ; Urban Tracker (Jodoin *et al.*, 2014) - a background subtraction tracker optimized in this work and therefore presented in its own section below - takes approximately sixty times longer than Traffic Intelligence to extract trajectories from a given scene when run on the same computer. Nevertheless, the extracted information is both simplified by the removal of the background, and more complete as most of the spatial information of targets is maintained. As a result, more varied and more taxing post-processing methodologies have been applied to background subtraction trackers than to the other types.

Much like feature-based methods, background subtraction trackers' reliance on movement for detection makes continued tracking of targets that cease movement difficult ; they quite literally fade into the background. This problem was targeted by Berclaz *et al.* (2011), who added a post-processing phase utilizing the k-shortest paths algorithm in order to interpolate highest-likelihood trajectories when a gap was detected. Without modifying (or publishing the calibration of) the underlying tracker, this added phase alone increased MOTA on the 2009 PETS dataset from 0.67 to 0.79.

More in line with the objectives of this work, Pérez *et al.* (2006a) applied evolutionary optimization to the detection stage of a background subtraction tracker (the specific evolutionary strategy utilized is oddly unmentioned, but was likely a genetic algorithm). Tracked objects were outlined by bounding boxes, and fitness was evaluated as a function of the blob-box overlap and aspect ratio (the ratio of width to height). Optimal fitness was predefined, and the tracker was tested on a subset of three videos chosen from forty at each generation, in order to converge to a shared optimum. No accuracy metrics were provided, but figure 2.11 illustrates a clear qualitative improvement in the quality of extracted blobs, and average positional errors were reduced by 25%. Unfortunately, the calibration and test scenes all consisted of a single pedestrian walking a straight path a uniform distance from the tracker, neglecting occlusion, grouping and perspective problems entirely.

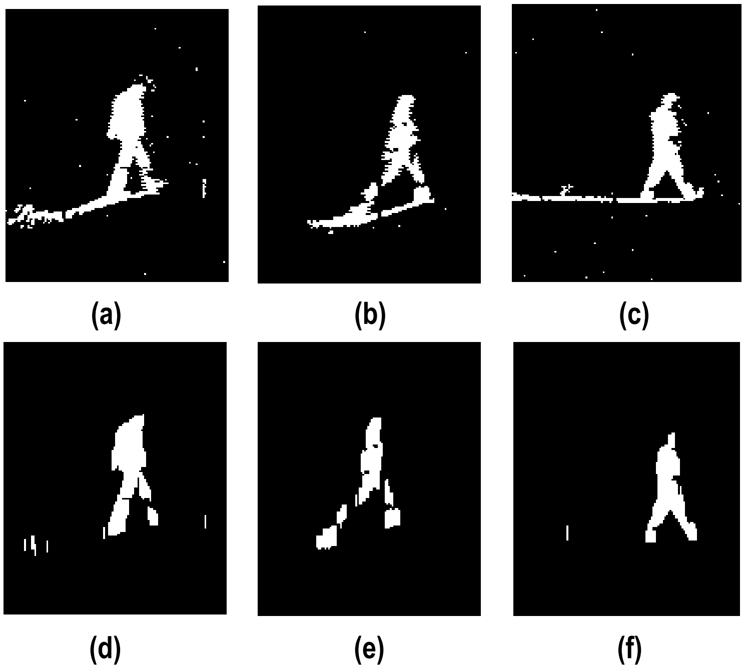


Figure 2.11 Application of default (a to c) and evolutionarily optimized (d to f) background subtraction to single pedestrian tracking. Source : Pérez *et al.* (2006b).

Tracking by detection

Tracking by detection is the approach most closely abiding by the detection-tracking-processing sequence presented earlier, as well as the most exploitable in experimental settings. Such trackers are initially trained on shapes or features of interest ; these may be the uniforms distributed in an experiment (as in Hoogendoorn *et al.*, 2003b), the round shape of the human head as viewed from above (used by Johansson *et al.*, 2007b), face recognition, or the general shape of a moving pedestrian. Once an object has been detected in subsequent frames, tracking can then either be performed by identifying the most similar objects, extrapolating the most likely position from the previous one(s), or a combination of both. One notable advantage of this school of pedestrian tracking is the applicability to mobile cameras, given that background motion does not significantly impact detection.

A large number of pedestrian detection methods rely on more abstract local features, the most common descriptor being the histogram of oriented gradients (or HOG). HOG consists of subdividing the image-space into cells, and evaluating either intensity or color gradients within said cells. The eponymous orientation of the histogram is determined by a weighed "vote" cast by each pixel, the objective being to generate the histogram with the largest variance (and therefore that which the most distinct). The resultant matrix of histograms highlights the outlines of shapes in each frame, facilitating automatic detection humanoid shapes (see figure 2.12). HOG is relatively sensitive to changes in lighting, therefore requiring photometric normalization for increased detection accuracy. When the latter is performed, however, individual body movements of pedestrians have been found to be ignored by the tracker, so long as they maintain a roughly upright position (Dalal and Triggs, 2005).

Two relatively straightforward applications of HOG are those of Jiang *et al.* (2010) and Andriyenko and Schindler (2011), with tracking aided by Kalman filtering and energy minimization, respectively, of low-probability matches. In both cases, the tracker was first trained to detect human shapes (i.e. full human bodies) in a variety of poses. Though neither study offered explicit metrics in the evaluation of their trackers, both appear to function adequately in low density scenes.

A significant problem in human-shape detection is that said shape can easily be occluded in higher density cases. Instead, Ali and Dailey (2009) trained the tracker detector stage using a large sample of human heads, with a variety of positions and postures, and then applied the trained tracker to a sliding window scanning each frame ; as the head is evidently the least likely part of a pedestrian to be occluded, it makes for an opportune target in high density scenes. The resulting detections were classified as either high or low probability matches, with only the former being used as definite targets. The tracking stage then proceeded using a particle filter, aided by the known locations of partial matches when a definite match could not be made. This method detected 76.8% of pedestrians whose heads were visible in a bidirectional, high-density sequence.

Sidla *et al.* (2006) used a similar method, searching for the Ω shape formed by the head and shoulders. This allowed a higher proportion of high-likelihood detections, between which tracking was performed via the KLT feature-based method. The objective being primarily counting, the resulting trajectories were extrapolated in cases where they did not reach the counting thresholds using a simple motion prediction method. Despite training, this tracker had a tendency for overdetection. However, once a correction factor was applied to account for the systematic portion of the error, counting accuracy was nearly 95% in the bidirectional, high-density cases tested.

Of course, constraining the tracker to detecting single shapes ignores the visual cues presented by the other parts of the body. In order to take advantages of these additional cues, Singh *et al.* (2008) applied a previously-developed (Wu and Nevatia, 2005) detector capable of identifying four distinct pedestrian body parts : the head-shoulder shape, the torso, the legs, and the body as a whole. The resulting tracker produced trajectories using the velocity vectors of whichever shape could be detected in a given frame, making hybridization to other methods or use of more elaborate trajectory-optimization techniques unnecessary so long as more than just the upper body was visible. A detection rate of 80 to 90% in complex, low density scenes was obtained.

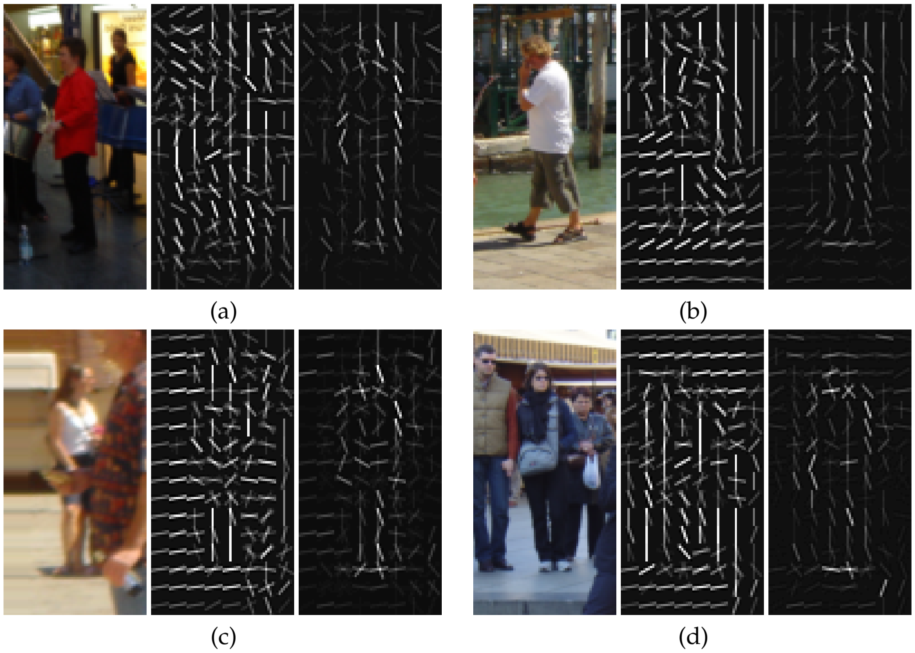


Figure 2.12 Visual cues used in HOG-based detection to generate target outlines. In each triplet is displayed, from left to right : (1) the input image, (2) the corresponding HOG feature vector (the dominant orientation in each cell), (3) the dominant orientations after post-processing by Support Vector Machines. Source : Dalal (2009).

The ability of HOG trackers to detect pedestrians in single frames has lead to their frequent hybridization with the tracking-by-movement methods described above. Khanloo *et al.* (2012) combined HOG with feature-based tracking in order to increase the information available to the tracking phase. Optimization was performed against a manually-generated ground-truth for each scene individually, in order to best combine the generated corner and HOG traces into accurate trajectories (all other portions of the tracker were calibrated manually). Standardized performance metrics were not provided, but it was demonstrated that error rates could be significantly reduced in a variety of scenes in comparison to using either corners or HOG alone. Unfortunately, this method requires manual initilization for each tracked object (making application to real world cases extremely impractical) and has substantial difficulty tracking in cases of occlusion.

In an approach specifically targetted at the occlusion problem, several authors have taken advantage of the fuller target information provided by HOG in order to hybridize the method with the background. Targets are detected via background subtraction, and then associated to their respective intensity or color histograms. When the target is lost (due to the blob being fused with another) the image-space is searched for the best fit to the HOG of the missing object, using an appropriate optimization algorithm. The most commonly used optimization method is the particle filter (for example, Guan *et al.*, 2013) though particle swarm (Zhang *et al.*, 2010) and bacterial foraging optimization (Nguyen and Bhanu, 2012) have demonstrated similar success with shorter run-times. It should be noted that while these methods perform admirably on the complex, high-density PETS datasets (MOTAs of 0.70 and above) their accuracy suffers substantially in lower density cases where the post processing is of little benefit (MOTAs between 0.27 and 0.34).

The methods presented above are summarized in table 2.1, along with their published performance and parameterization.

2.2.4 Trackers Used in this Work

Traffic Intelligence

The open-source Traffic Intelligence (TI) project is an implementation of feature-based tracking, specifically utilizing the KLT corner detection method included in OpenCV (Bradski and Kaehler, 2008). Initially designed for the monitoring of road-traffic, TI is used for the multimodal tracking of the complex movements within intersections. It includes tools for the interpretation of road user trajectories, their behavior and their interaction for safety diagnosis, facilitated by automated classification of the detected objects.

The parameters regulating detection and tracking in TI are presented in table 2.2. At the feature-detection level, they present a straightforward application of the KLT method : features are selectively tracked based on their quality (i.e. their consistency between frames) and filtered by their behavior, including maximum acceleration and minimum duration. They are grouped at the object-level, a process based primarily on ensuring a minimum object size, both in terms of number of constituent features and of maximum distance between said features. It is worth noting that object construction can be (and is always, in the following chapters) performed in the world-space : when the tracker is provided a homography matrix, allowing the conversion of image-space coordinates into real-world positions, inter-feature distance is evaluated in real meters. This allows the grouping of features into objects to form targets of consistent actual size, less distorted by camera perspective.

Table 2.1 Summary of selected pedestrian tracking methods. It should be noted that reported accuracies were evaluated in different scenes and with a variety of metrics, making direct comparison difficult. Pedestrian densities reported in the table correspond to a visual evaluation of density in the scenes : low density denotes that individuals are generally clearly separated, while high density implies substantial occlusion and/or grouping of pedestrians. DR : Detection Rate, calculated differently for each author. Source : Ettehadieh *et al.* (2014).

Tracking method	**Authors**	**Year**	**Novel contributions**	**Test sequence characteristics**	**Accuracy (metric)**	**Parameterization**	**Notes**
Feature-based	Saunier & Sayed	2006	Application to transportation systems	Multimodal ; low pedestrian density	0.23-0.65 (MOTA)	Manually set	Designed for multimodal traffic
	Rabaud & Belongie	2006	Pedestrian counting in dense crowds	High density, complex scenes	0.78-0.93 (DR)	Partial training of detector	Designed specifically for counting
	Ismail et al.	2010	Automated video analysis for before/after safety evaluations	Multimodal ; low pedestrian density	0.70 (DR)	Optimization of some tracker parameters	
	Khanloo et al.	2012	Hybridizes feature-based and Histogram of Oriented Gradients	Low density, multiple directions	*Unreported*	Optimization of cue-selection	PETS dataset
Background subtraction	Pérez et al.	2006	Evolutionary optimization of the segmentation stage	Single pedestrian	*Unreported*	Evolutionary optimization	Optimization of only one tracker stage
	Berclaz et al.	2011	K-shortest paths optimization	Low density, multiple directions	0.58-0.86 (MOTA)	*Undisclosed*	CAVIAR dataset
	Nguyen & Bhanu	2012	Bacterial foraging optimization	Varied	0.35-0.71 (DR)	Manually set	Best performance in high-density scenes
	Jodoin et al.	2013	Use off eature-points for object identification	Multimodal ; low pedestrian density	0.68-0.93 (MOTA)	Manually set	Designed for multimodal traffic
	Guan et al.	2013	Particle filter approach	Low density, uni- and bi-directional	0.63-0.92 (MOTA)	Manually set	
Tracking by detection	Sidla et al.	2006	Searches image for shoulder-and-head shapes	High density, complex scenes	0.89 (DR)	Manually set	Focused on pedestrian counts through tracking
	Singh et al.	2008	Completes tracks using low-confidence traces	Low density, multiple directions	0.73 (DR)	Manually set	CAVIAR dataset
	Ali & Dailey	2009	Confirmation-by-classification	High density, bi-directional	0.77 (DR)	Adaboost	Head detection for dense crowds
	Jiang et al.	2010	Hybridizes Histogram of Oriented Gradients detection with color tracking	Low density, multiple directions	*Unreported*	Manually set	CAVIAR dataset
	Andriyenko & Schindler	2011	Tracking using energy minimization	Low density, multiple directions	0.33-0.85 (MOTA)	Manually set	Tested with multiple sets of parameters

Within the complex multimodal settings for which it was designed, TI has demonstrated good accuracy, with MOTAs between 0.60 and 0.85 (Jodoin *et al.*, 2014). Such cases, however, primarily involve vehicles ; MOTA calculated for pedestrians alone tends to be lower (near 0.50) despite pedestrian density typically being very low. Run-times are dependent on the number of features tracked, and therefore on both the preset minimum feature-quality and the number of targets in a scene ; in the tested cases, it varies between running in real-time and approximately a fifth of real-time.

Urban Tracker

Like TI, Urban Tracker (UT) was designed primarily for road traffic (Jodoin *et al.*, 2014). Built around the ViBe background subtraction algorithm (Barnich and Van Droogenbroeck, 2011), UT's tracking stage is aided by BRISK corner detection (Leutenegger *et al.*, 2011). Though both the background subtraction and feature-tracking methods are individually relatively standard, their combination allows for improved tracking in cases of occlusion, fragmentation and grouping. Additional functions were also implemented in order to identify and ignore shadows and to check whether an entering object is the same as one having previously left the image frame. A list of the tracking parameters is presented in table 2.3.

UT has been directly compared to TI. Although only mild improvements were observed in multimodal tracking, significantly higher accuracies were achieved in pedestrian tracking in the same sequences as examined above (MOTAs ranging from 0.70 to 0.90, in contrast to TI's 0.50) though again pedestrian density was low. These improvements, however, come at a severe performance cost : run-times for UT are generally between 60 and 100 times the length of the evaluated video in our tests.

2.3 Video-tracking metrics

Object tracking, whether it be through video or other means, requires the performance of two fundamental tasks : *detection* and *tracking* (Maggio and Cavallaro, 2011), also referred to as the motion and matching problems, respectively (Trucco and Plakas, 2006). First, the objects of interest must be detected, whether within single frames (by shape or color) or by their motion. Second, the detected objects must be tracked for the duration of their existence in the video space by adequate matching of the individual detections. The capacity of a tracker to perform these tasks is its *accuracy*, as defined in section 1.2.2.

In spite of a growing interest in video-based tracking, there is currently no single, standard measure for accuracy (see table 2.1). It is therefore not uncommon for authors to devise and use their own metrics. The simplest of these is to count the number of tracked objects and to compare it to the number of objects actually in the scene, as done by Sidla *et al.* (2006) (it should be noted that the stated objective of this paper was the counting of pedestrians ; however, it remains the only provided performance metric of the developed tracking method).

Unfortunately, the typical errors committed by automated trackers render counting non-indicative of actual performance. The most evident of these errors take place at the detection phase : *misses*, where an object simply goes undetected, and *false positives*, where a detection occurs in absence of a corresponding object. These errors influence counts in opposite ways ; if both occur a similar number of times, the resulting count may appear adequate despite poor correspondence of tracker trajectories to the real ones. Some authors account for this by evaluating performance as the fraction of misses and false positives relative to the expected number of detections (e.g. Ali and Dailey (2009)).

However, further errors exist at the tracking stage : a *missmatch* occurs when an object is properly detected but is associated to a different trajectory than it was previously, resulting in either two objects exchanging identities or a single object having its track end and be replaced by a new one. The former is undetectable by counts alone ; the latter would increase the count. All the above errors are illustrated in figure 2.13.

Errors in detection and tracking can be accounted for separately : Perera *et al.* (2006) defined a *track completeness factor* (the ratio of detected to present objects) and *track fragmentation* (the average number of tracks per real object). This method has the advantage of providing information on the type of errors committed. However, as stated in section 1.2.2, it would increase the number of variables to optimize for in the context of this work, and the additional information would be unlikely to be of use to a generalized and automated optimization framework.

The metrics used by Sidla, Perera and others also neglect to evaluate the precision of the resulting tracks, or how closely the location an object was detected matches its real position. While of relatively little use in many tracking applications (e.g. counting) in the present context of obtaining microscopic pedestrian flow data, precise locations are vital both for evaluating speed and density as well as calibrating models for behaviors such as collision avoidance. The metrics that follow measure both tracker accuracy and precision, and have both been used in the comparison of multiple video-based trackers.

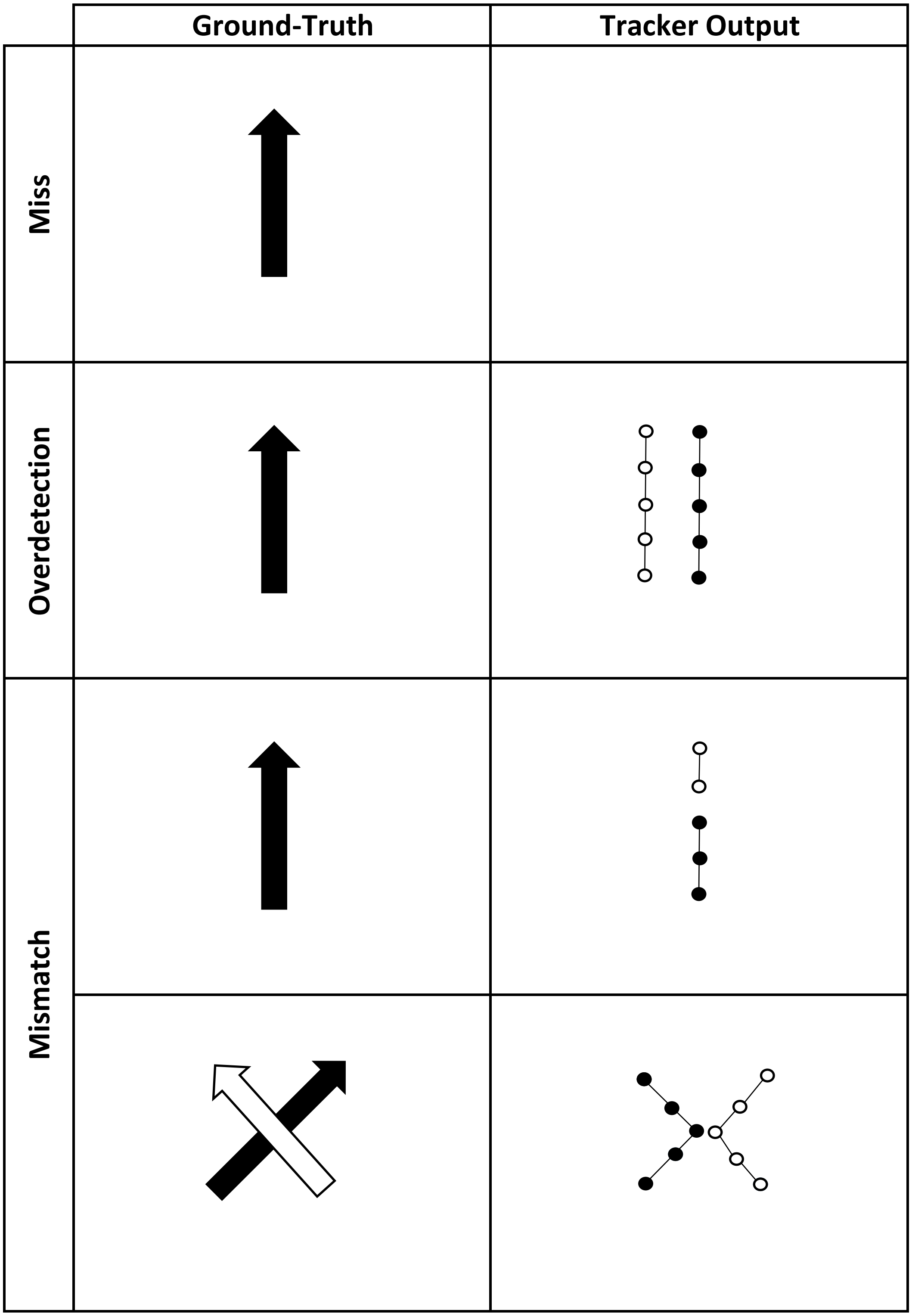


Figure 2.13 Illustration of the typical errors commited by automated trackers.

2.3.1 The VACE metrics

Developed as part of the US Government Video Analysis and Content Extraction program, SFDA (Sequence Frame Detection Accuracy) and ATA (Average Tracking Accuracy) are two metrics which each combine accuracy and precision evaluations in a single measure ; as their names imply, the former evaluates detection alone, whereas the latter measures tracker performance more globally. Given the objectives stated in section 1.3, SFDA could be neglected in favor of ATA ; it is presented here only because the tracking metric is built upon its concept.

SFDA is defined as :

$$SFDA = \frac{\sum_{t=1}^{N_{frames}} FDA(t)}{\sum_{t=1}^{N_{frames}} \exists(N_G^t \vee N_D^t)} \tag{2.2}$$

Where :

$$FDA(t) = \frac{\sum_{i=1}^{N_{mapped}^t} \frac{|G_i^t \cap D_i^t|}{|G_i^t \cup D_i^t|}}{\left[\frac{N_G^t + N_D^t}{2}\right]} \tag{2.3}$$

And :

- G_i denotes the i^{th} ground-truth object
- D_i denotes the i^{th} detected object
- N_G and N_D represent the number of ground-truth objects and the number of detected objects, respectively
- N_{frames} refers to the number of frames where either ground-truth object i or detected object i existed in the sequence
- N_{mapped}^t denotes the number of detected object - ground-truth pairs
- the t index denotes the existence of the object at a given frame t

Essentially, the numerator in equation 2.3 represents the amount of overlap between associated detected and ground-truth bounding boxes. *FDA(t)* is an averaged overlap of bounding boxes within a given frame *t*. SDFA, then, is the average FDA over all frames where either a ground-truth or detected object exists. This equation can be further refined by defining threshold overlaps for FDA, in order to ignore insufficiently accurate detections and/or for-give minor inconsistencies ; this is illustrated in figure 2.14. SFDA ranges from 0 to 1, with 1 representing perfect matches on a frame-by-frame basis.

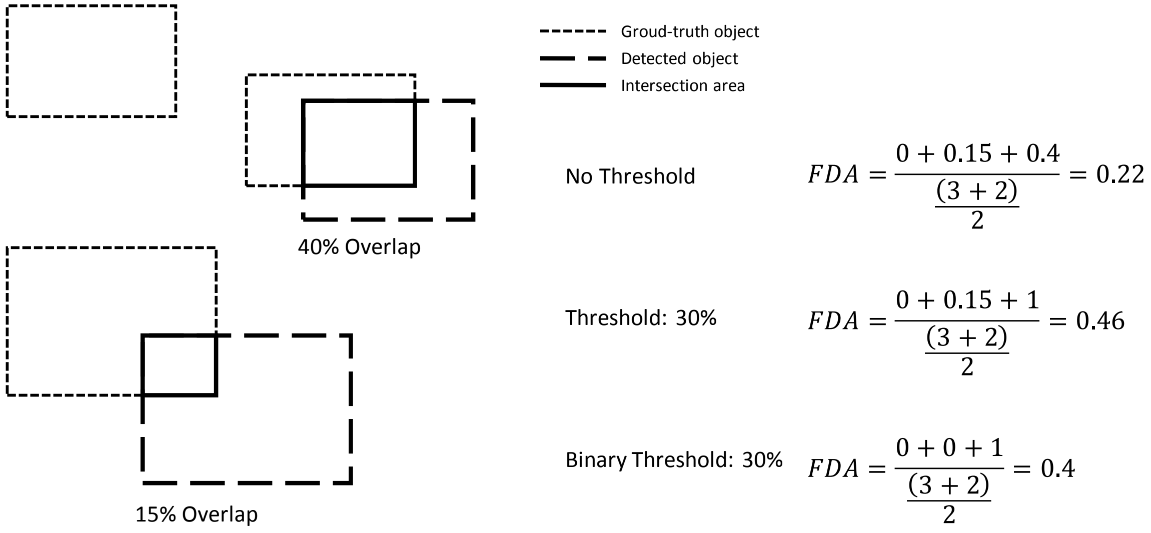

Groud-truth object
Detected object
Intersection area
40% Overlap
15% Overlap
No Threshold
$FDA = \frac{0 + 0.15 + 0.4}{\frac{(3+2)}{2}} = 0.22$
Threshold: 30%
$FDA = \frac{0 + 0.15 + 1}{\frac{(3+2)}{2}} = 0.46$
Binary Threshold: 30%
$FDA = \frac{0 + 0 + 1}{\frac{(3+2)}{2}} = 0.4$

Figure 2.14 Sample example demonstrating the calculation of FDA with various thresholds. Inspired by Kasturi *et al.* (2009)

SFDA is solely a spatial metric operating on single frames : misses and overdetections are penalized, but mismatches are not detected. In contrast, ATA is spatiotemporal, verifying consistency between frames. It relies on the calculation of STDA (Sequence Track Detection Accuracy), calculated as :

$$STDA = \sum_{j=1}^{N_{mapped}} \frac{\sum_{i=1}^{N_{frames}^{t}} \frac{|G_i^t \cap D_i^t|}{|G_i^t \cup D_i^t|}}{N_{(G_i \cup D_i \neq \emptyset)}} \tag{2.4}$$

This equation is very similar to equation 2.3. The difference is that where FDA evaluates overlap within each frame individually, STDA evaluates overlap between a real and detected objects across all frames where the either exists, thereby allowing for detection of mismatch errors. This value is then normalized to the same range as SFDA :

$$ATA = \frac{STDA}{\left[\frac{N_G + N_D}{2}\right]} \tag{2.5}$$

While the above equations would imply that SDFA and ATA are intended solely for trackers that detect objects via bounding boxes, the overlap terms may be replaced with a normalized distance measure :

$$d_{G_iD_i} = 1 - \frac{DistanceBetweenG_iandD_i}{MaximumMatchingDistance} \tag{2.6}$$

This would allow ATA to be generalized to all trackers, particularly given that bounding boxes can be reduced to points by simply calculating their centers. Nevertheless, though it remains in use in some instances of tracker comparison (notably by the CLEAR consortium) ATA scores are rarely published in papers presenting individual trackers, in favor of the following metrics.

2.3.2 The CLEAR MOT metrics

Established in order to provide a harmonized metric for the evaluation and comparison of video-based trackers lead by the CLEAR (CLassification of Events, Activities and Relationships) consortium, the CLEAR MOT metrics (Keni and Rainer, 2008) consist of MOTA (Multiple Object Tracking Accuracy) and MOTP (Multiple Object Tracking Precision). They stemmed from the observation by some researchers that the use of a single metric made the identification of failure components difficult. Like ATA, they also have a detection-level equivalent (MODA and MODP), though these are only atemporal versions of the tracking metrics and will therefore not be discussed here.

MOTA is, simply, the ratio of commited errors to the number detections expected from the ground-truth. It is defined as :

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \tag{2.7}$$

where m_t, fp_t and mme_t are the number of misses, overdetections (false positives), and mismatches, respectively.

MOTP is the average distance between expected and actual detections, determined by :

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{2.8}$$

where d_t^i is the distance between associated ground-truth and detected object pair i at frame t and c_t the number of associated pairs in frame t.

Together, MOTA and MOTP are the most widely utilized metrics in video-based tracking, used by both CLEAR and the PETS (Performance Evaluation of Tracking and Surveillance) workshop (Ellis *et al.*, 2009) as well as a large number of individual papers (see table 2.1).

2.3.3 Associating Detected with Ground-Truth Objects

All the methods described above apply to sets of ground-truth and tracker-detected objects which have already been associated to one another. However, none explicitly prescribes how this association is to be performed, and yet this process is both necessary and highly influent on the subsequent evaluations for any metric.

Object assignment can be very complex (Saunier *et al.*, 2009). While the simplest method is to simply associate each ground-truth object with the nearest detected one at every frame, this may lead to nonsensical matches, such as if the nearest object is particularely far away or if this causes repeated switching between frames as a single ground-truth track is at similar distances from two detected objects. Furthermore, care must be taken in deciding whether to assign objects on a one-to-one basis, or to allow one-to-many or many-to-many matches. While the former option would on the surface appear to be ideal, the latter two have the advantage of facilitating the distinction of different types of error, namely over-segmentation (one ground-truth object to many detected objects), over-grouping (many ground-truth objects to one detected), missed detections and false detections.

2.4 Optimization methodologies

Optimization methodologies are numerous and highly varied in application. This section therefore presents an overview of the selection process, guided by the specific requirements of the current application. The methodologies, their applicable cases and their strengths and weaknesses are all taken from Ross (1997).

In the present case, the optimization problem is defined by the following constraints :

– The desired *generalizability* of the algorithm : Though the parameters presented in tables 2.2 and 2.3 are numerous, an optimization algorithm tailored to them specifically could be simplified and/or accelerated through prior evaluation of the parameters. This would allow greater restriction of the parameter ranges (by eliminating ranges known never to produce optimal solutions, for example) or an iterative approach of optimizing parameters in sequence, each using the most suited methodology. However, in order to be applicable to any current or future tracking method, the optimization process cannot rely on prior knowledge but must instead treat the tracker as a "black box", relying solely on observations made during optimization. In other words, a minimum of assumptions about the problem must be made ; heuristic and metaheuristic optimization methods, being particularly non-reliant on such assumptions, are therefore ideal.

- The *complexity* of the search space : The assumptions mentioned above include those regarding the individual parameters. Although it is likely that some parameters display simple, convex behavior (i.e. a singular global maximum) this is unlikely to universally be the case, particularly given that at low performance the tracker's measured accuracy is liable to be highly sensitive to noise. Furthermore, as multiple parameters tend to regulate any single tracker task (and said task is most probably itself reliant on other tracking phases) parameters certainly demonstrate complex joint distributions, further obfuscating the desired global maximum. Optimization algorithms that are overly heuristic (e.g. the tabu search algorithm, which rapidly eliminates ostensibly low-performance areas from the search space) or reliant on gradient-detection (e.g. sequential quadratic programing) are therefore likely to miss the region containing the optimal solution entirely.
- The potentially *quasi-infinite search-space* : The combination of floating-point and unbounded variables with the hyperdimensionality of the search space makes the number of possible parameter sets infinite. This means that though the complexity of the search-space renders overly-heuristic methods unreliable, the opposite approach of thoroughly examining the granularity of said space (i.e. carefully examining the space as a whole before focusing on optimization in earnest) is equally impractical.
- The *lengthy computation time* for individual observations : As stated in the previous section, tracker run-times vary enormously but are on average substantially longer than real time ; tracking over a single minute of video may take anywhere from a minute to over an hour. Considered alongside the infinite size of the search-space, this makes minimizing the required number of observations a priority if the algorithm is to be practical for real-world application (i.e. if the process is sufficiently costly, it becomes more cost-effective to simply manually extract the desired trajectories). Consequently, methods requiring a large number of observations between positional moves (e.g. genetic algorithms, which evaluate at a minimum ten positions per iteration) or those liable to "get stuck" in some common situations (e.g. adaptive mesh refinement) may be too costly for the present case.

Despite the difficulties highlighted above, the very complexity of the relationship between parameters and tracker performance suggests an avenue for optimization. Indeed, it is safe to assume that no individual parameter influences tracker accuracy completely independent of all others, regardless of the specific tracker in question. The optimization problem can therefore be likened to sampling from an unknown, multi-dimensional joint probability distribution, a class of system often considered best solved by Markov Chain Monte Carlo (MCMC) methods.

Monte Carlo approximation is a method for estimating the probability of certain values of a function when said probability cannot be calculated by exact methods (e.g. integration or summation) but the function *can* be simulated for specific values (Geyer, 2011). In essence, Monte Carlo relies on elementary statistics to evaluate a state-space via pseudo-random sampling, the pseudo- prefix referring to the pseudorandomness of computer simulation. MCMC, then, is the use of Markov Chains to generate the sample population used in the Monte Carlo approximation.

A sequence X_1, X_2,... of random elements of some set is a *Markov Chain* if the conditional distribution of X_{n+1} depends only on X_n (Geyer, 2011). In the case of MCMC, said chain has *stationary transition probabilities*, meaning the conditional distribution of X_{n+1} given X_n is also independent of n (this leads to a complex issue : if we could verify the independence of X_{n+1} to n, we could simply simulate any state X_i independently, rendering the use of Markov Chains pointless ; the continued use of stationary Markov chains depends on a set of theorems that are beyond the scope of this work).

The Metropolis-Hastings algorithm is a specific application of MCMC aimed at functions with multiple variables. After initializing from an arbitrary point X_0, tentative subsequent point X'_{n+1} is selected randomly from a distribution centered at X_n - a procedure known as a *random walk*. Whether X'_{n+1} is kept as X_{n+1} is dependent on the *acceptance ratio* α, which is the ratio of the probabilities of the tentative and prior points.

Simulated annealing is, in turn, an application of the Metropolis-Hastings specifically to optimization. Inspired by the metallurgical process of annealing (a technique involving controlled heating and cooling of a material in order to reduce defects) simulated annealing differs from the previous algorithm in two ways : the use of the simulated values directly (and not their probabilities) for calculation of the acceptance ratio, and the inclusion of a "temperature" variable in the acceptance function in order to decrease the incidence of regression to higher energy states (or less optimal solutions) over time. This heuristic-based search method is presented in the pseudocode below ; a more explicit application of simulated annealing to tracker optimization is presented in section 3.7.

The simulated annealing algorithm exists in a number of implementations, distinguished by their formulations of temperature, the acceptance probability, and next-state generation (though the latter is generally context-specific). Regardless of the equations used, however, all methods share the same inherent advantages : a small number of iterations until convergence and a relative insensitivity to local optima. However, simulated annealing gives no guarantee of providing the best possible solution. This, combined with the fact that changes between iterations are random, means that the final solution may vary both between applications from a same starting point and as a function of the initial parameters ; this concern is addressed in section 3.10.

```
**initialization**;
$i = 0$ #initialization of iteration counter i, used in calculating current temperature;
$Param_{curr}$ = initialized with arbitrary parameters;
$X_{curr} \leftarrow$ performance observed for $Param_{curr}$;
$T = f(i)$ #Definition of initial temperature $T$;
**while** $T$ != 0 **do**
    $Param_{new} \leftarrow$ new parameters generated as a modification of $Param_{curr}$;
    $X_{new} \leftarrow$ value observed for $Param_{new}$;
    $\alpha = min(f(X_{curr}, X_{new}, T), 1)$ #calculate acceptance probability of new parameters, based on the ratio of $X_{curr}$ and $X_{new}$, weighed by $T$;
    **if** $rand(0, 1) < \alpha$ **then**
        if new value is accepted, move to new value;
        $Param_{curr} \leftarrow Param_{new}$;
        $X_{new} \leftarrow X_{curr}$;
    **end**
    $i = i + 1$;
    $T = f(i)$
**end**
```

Algorithm 1: A generic implementation of simulated annealing.

Table 2.2 List of Traffic Intelligence parameters affecting detection and tracking, along with their range and description.

TRAFFIC INTELLIGENCE

	Parameter name	**Type**	**Min.**	**Max.**	**Description**
FEATURES	feature-quality	float	0	1	Minimum quality of corners to track.
	min-feature-distanceklt	float	0	10	Minimum distance between features, in pixels.
	window-size	int	3	10	Distance within which to search for feature in next frame, in pixels.
	pyramid-level	int	1	5	Maximum pyramid level for feature tracking.
	ndisplacement	int	2	4	Number of displacements to test minimum feature motion.
	min-feature-displacement	float	0	0.1	Minimum displacement of features between frames (pixels).
	acceleration-bound	float	1	3	Maximum ratio of speeds between frames.
	deviation-bound	float	0	1	Maximum cosine of feature velocities between frames.
	smoothing-halfwidth	int	0	11	Number of frames to smooth positions.
	min-tracking-error	float	0.01	0.3	Minimum error to reach to stop optical flow.
	min-feature-time	int	5	25	Min. time (in frames) a feature must exist to be saved.
OBJECTS	mm-connection-distance	float	0.5	2	Distance to connect features into objects, in meters.
	mm-segmentation-distance	float	0.1	1.9	Segmentation distance, in meters. Must be less than connection distance.
	min-features-group	float	1	4	Minimum average number of features per frame.

Table 2.3 List of Urban Tracker parameters affected detection, tracking and post-processing of trajectories, along with their range and description.

	URBAN TRACKER **Parameter name**	**Type**	**Min.**	**Max.**	**Description**
BACKGROUND SUBTRACTION	bgs-minimum-blob-size	int	10	-	Min. size of blobs, in pixels.
	max-lost-frame	int	1	-	Max. number of frames to continue searching for a lost object.
	max-seg-dist	float	0	1	Max. distance between two blobs to be considered an object, as a ratio of blob diameter.
	max-hypothesis	int	1	-	Max. frames to consider an object hypothesis.
	minimum-match-between-blobs	int	1	-	Min. number of matching features to establish two blobs as the same object.
FEATURE DETECTION	brisk-threshold	int	1	20	Threshold determining minimum quality of features to detect.
	brisk-octave	int	1	5	Number of layers to use in feature detection for each frame.
	match-ratio	float	0	1	Min. matching ratio between second-best and best match for a given object.
FUNCTIONS	urban-isolated-shadow-removal	boolean			Automated shadow removal.
	verify-reentering-object	boolean			Verifies whether entering objects correspond to preexisting ones.
	bgs-remove-ghost	boolean			Retroactively removes blobs if they are not associated to an object.

CHAPTER 3 METHODOLOGY

To reiterate the objectives stated in section 1.3, the goal of this work is the development of a generalizable optimization framework for the improvement of video-based pedestrian tracker performance in specific scenes, utilizing a short, representative sequence for which the ground-truth has been manually established. Said framework should converge to an adequate, improved solution in a reasonable time-frame (i.e. substantially faster than conducting manual trajectory extraction for the entire scene) regardless of the optimized tracker's parameters.

In order to facilitate application to other trackers not mentionned here, the presented framework, named Tracker Optimization for Pedestrians (TrOPed) is open-source and available online (Ettehadieh, Dariush, 2014). This chapter is dedicated to elucidating its construction.

3.1 Optimization Schema

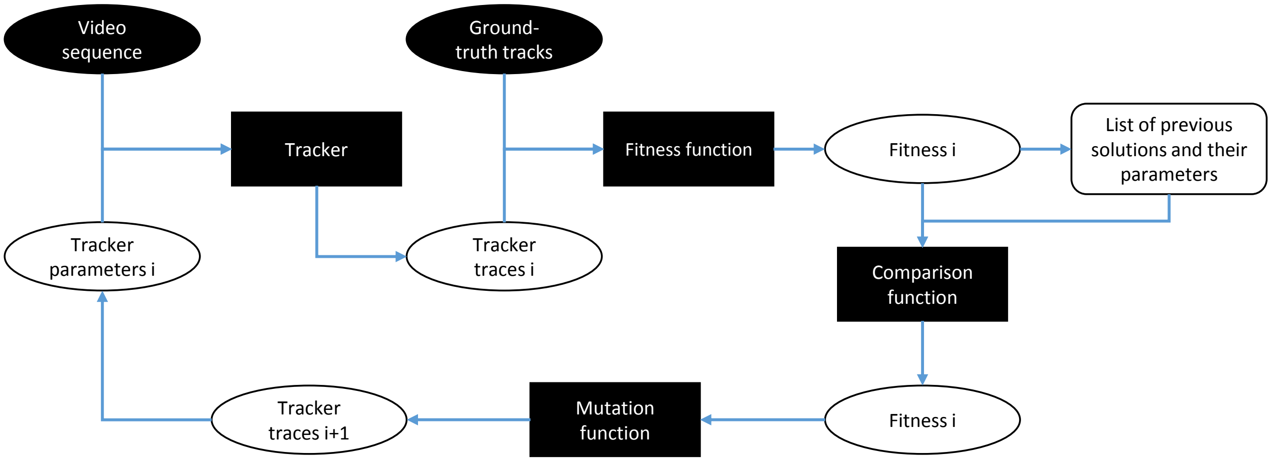


Figure 3.1 Flow diagram of the TrOPed algorithm. Source : Ettehadieh *et al.* (2014).

Figure 3.1 schematically presents the overall structure of the TrOPed algorithm itself. However, both its development and use require additional preliminary steps, notably the collection of the initial video data and the extraction of ground-truth trajectories. The summary below represents a fuller picture of these required steps, and is used to structure the remainder of the present chapter.

- **Data-collection** (section 3.2) : Collection of pedestrian video data in a variety of scenes of sufficient complexity to pose a challenge to the trackers.
- **Establishing the Ground Truth** (section 3.3) : Extraction of ground-truth trajectories from a short, representative sequence of the video data.
- **Algorithm Inputs** (section 3.4) : Algorithm setup and definition of the tracker parameters in a manner suitable for optimization.
- **Homography Parameters** (section 3.5) : Transformation of image-space trajectories to real-world coordinates, in the specific case of pedestrian tracking.
- **Evaluating Performance** (section 3.6) : Evaluation of the tracker's performance with a given set of parameters.
- **Optimization Algorithm** (section 3.7) : Comparison of the current to the previous iteration of the parameters, and selection from the pair to best move towards an optimal solution.
- **State-Generation Function** (section 3.8) : Modification of the current parameters in order to generate the next iteration.
- **Finalization of the Algorithm** (section 3.9) : Encasing the previous steps into a functional optimization loop.
- **Calibration** (section 3.10) : Calibration of the parameters of the algorithm itself.

3.2 Data-Collection

Section 2.2 presented the primary difficulties in tracking pedestrians through video, namely the complexity of their movement and visual effects such as occlusion and grouping. In addition, it was observed that a majority of video-based trackers in the literature were calibrated and tested on selections of scenes that were homogeneous in terms of pedestrian density and behavior.

In order to best confront these issues, real-world video data was sought that periodically presented difficult tracking scenarios, interspersed with ostensibly simpler, lower density periods. The studied locations should also be representative of typical scenarios, and thus not contain overly unique characteristics such as highly reflective surfaces or unusual geometry. Two locations were eventually selected : a central corridor in Polytechnique Montreal, and the outside of a Montreal subway station and bus terminal. Additionally, video was obtained which was recorded from above at a downtown New York City crosswalk located in front of the Pennsylvania train station.

The means by which this data was collected is presented in the subsection below, followed by an examination of the three chosen test cases.

3.2.1 Data-Collection Method

Video was recorded through the use of wide-angle personal cameras, affixed to the walls of the studied locations using adhesive tape. They were positioned so as to best approximate the camera perspectives typical of surveillance cameras, capturing pedestrian movement at a slight downward angle (approximately twenty to thirty degrees from horizontal). Video was recorded over several hours in all cases - limited only by the battery life of the cameras - at a resolution of 1280x720 pixels. The cameras were capable of resolutions of up to 1920x1080 pixels, and higher resolutions increase the information available to trackers and thus their expected performance. However, this comes at the cost of both computation times and the required storage space. As the latter was of significant concern, a compromise was made between resolution and the storage capacity of the devices in an attempt to maximize the length of the obtained video data.

The New York sequence, recorded by a third party, utilized identical cameras, albeit at a different angle (as mentioned above) and at the cameras' maximum resolution. Higher resolutions provide additional information to the trackers, theoretically allowing higher performance, but were judged unlikely (at present) to be utilized in a majority of permanently-installed cameras not specifically intended for research use.

The primary disadvantage of the selected recording equipment is the built-in wide-angle lens. Said lens allows a viewing angle of up to 150 degrees, granting a broader view of the recorded scene and therefore allowing coverage of a same area from closer than would otherwise be possible. However, this comes at the cost of substantial distortion near the edges of the frame, even when the field of view is reduced to the camera's minimum of 74 degrees as was done in the two Montreal sequences.

In the videos recorded for the purposes of this work, the restricted field of view relegates the distortion primarily to the edges. Tools exist to correct this distortion by transformation of each video frame (calibrated by recording a regular checkerboard of known dimensions) and one is included in the Traffic Intelligence project, yet these both remove certain border areas from the scene and increase the resolution of the resulting corrected video (see figure 3.2) increasing the run-time of the trackers. Distortion correction, in the context of optimization-by-calibration to ground-truth, also has to be performed twice : once in order to extract the ground-truth, and again when applying the calibrated parameters to the full video sequence. Given these issues, and seeing as the frame areas displaying marked distortion were outside the zones of interest in both optimized scenes recorded in Montreal (representing walls or pedestrians only partially in the video frame) no correction was performed in these cases.

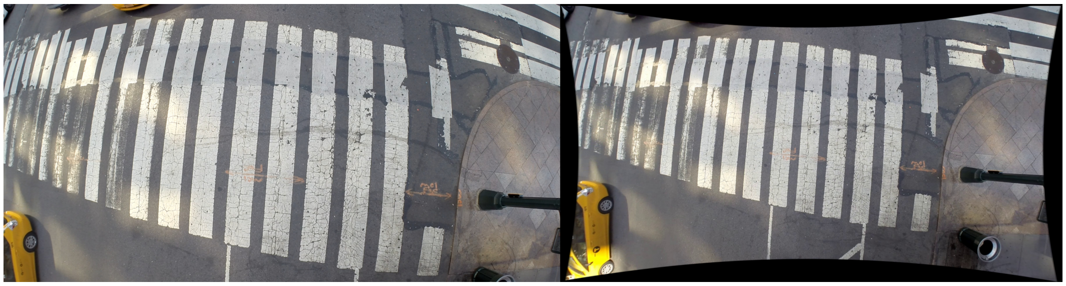


Figure 3.2 Lens distortion in the New York video sequence (left) and the same frame after correction by the tool included with Traffic Intelligence (right). Note that though these frames are presented at the same size for clarity, the left one is 1920x1080 pixels, whereas that after correction is 2515x1415 pixels.

In contrast, the New York sequence was filmed using the camera's maximum field of view, causing the marked curvature of otherwise straight lines that can be observed in the left image of figure 3.2. Although a ground-truth could be extracted directly from the raw video so as to not penalize performance scoring, this would lead to curved trajectories when projected to the real-world as well as being needlessly difficult for the two optimized trackers, both of which assume targets will majoritarily move with constant speed and direction. Therefore, distortion in the New York sequence was corrected for, using parameters taken from Saunier (2011).

3.2.2 Test Cases

All three test cases display the desired variety of tracking complexity, with multiple intersecting directions of movement and densities ranging from zero (one or no pedestrians in frame) to nearly one pedestrian per square meter.

Cameras were plainly visible (and emitted a red light while recording) and, in the case of the Polytechnique Montreal recording sessions, the data-collection team was required to place posters directly outside the observed area informing passers-by that they would be filmed. Because of this, it was feared that some pedestrians might either avoid the cameras or otherwise change their behavior upon noticing the equipment. Fortunately, this did not appear to be the case : only a handful of individuals (less than one percent, as estimated by on-scene observation) appeared to notice the cameras, even when the latter's presence was explicitly announced with posters. When the recording equipment was noticed, it usually elicited only a singular upward glance, though on five occasions (over sixteen hours of filming) individuals approached the camera for closer inspection and were promptly explained the study by the research team.

Polytechnique Montreal was the first recording location. Cameras were installed on opposite ends of a main corridor (in the North-South axis, see figure 3.3) leading to a tunnel connecting the university's two buildings, which serves as the primary inlet to the adjoined building from the public transit network. Said corridor also grants access to classrooms on either side, and is partially obstructed by a stairway leading both to an outside exit and to the university library. Equally of note is the presence of large windows looking into the attached classroom ; it is not uncommon for students to stop walking in front of these windows in order to greet friends within.

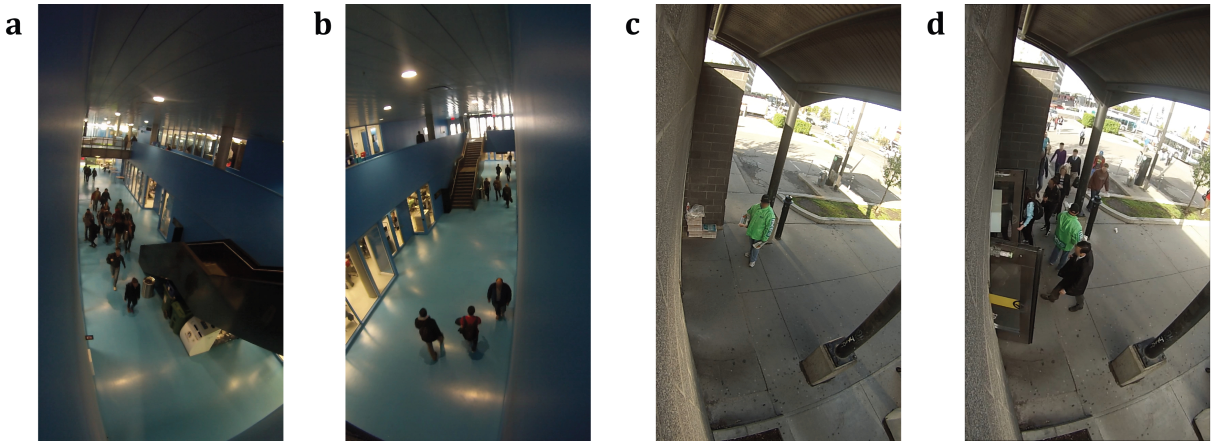


Figure 3.3 Example frames taken from the two recording locations. (a) and (b) represent the two camera angles used in the Polytechnique corridor, (c) and (d) from the sole camera used for optimization at the subway station, with the former showing minimum pedestrian density and the later taken during the arrival of a bus.

Data was collected on two weekday mornings during the end of the winter semester of 2014, for between three and four hours beginning at 8 AM in each instance. This timeframe allowed the capture of the steady influx of employees, punctuated hourly by students' movements between periods, either heading to courses or to the nearby coffee shops during breaks. Movement was therefore predominantly along the corridor's axis and towards the tunnel, but movement between all five accesses was regularly recorded. Given the corridor's width (more than six meters at its widest) average density was never observed to be particularly high, though students had a strong tendency to move in large, dense groups, providing ample challenge to the trackers.

The second data-collection location was the Montreal **subway station**. Due to an agreement with its operator - the Société de Transport de Montréal - the specific station studied and several other identifying details cannot be disclosed in this document. Nevertheless, the station is a local transit hub, providing access to the municipal subway as well as multiple bus lines both via stops on the street an within its bus terminal. It is situated in a high density residential and commercial sector, thereby having substantial flow both into and from the subway, and provides a large number of pedestrian accesses.

Like in the previous location, recording was performed on two summer weekdays beginning in the early morning (7 AM in this case) in order to capture the morning rush-hour. Five cameras were installed in an attempt to record pedestrian entry from all sources (including walkways, individual bus lines and the subway access) and allow eventual construction of an internal origin-destination matrix for the station. For optimization, however, data from a single camera was used, covering what was observed to be the most used doorway. Said doorway serves as primary pedestrian access to the station and has walkways in all four cardinal directions. Again like the prior location, pedestrian flow was periodic, though here it was dependent on transit arrivals (buses and subway) in addition to pedestrian signals at nearby intersections. One final, unanticipated feature of the location provided an additional tracking challenge : individuals were present throughout both recording sessions, distributing newspapers to passers-by and therefore moving irregularly and serving as both obstacles and attractors for pedestrians.

Finally, a third set of pedestrian video was obtained overlooking the crosswalk of the three-way intersection in front of the intercity **Pennsylvania Train Station** in downtown New York. Filmed from above, the single camera was aimed at an individual crosswalk, though it also captured parts of another crosswalk as well as the sidewalk on one corner (see figure 3.2. Unlike the previous cases, where there is some continuous pedestrian flow with punctual increases, this intersection is signalized and therefore has regular periods with no pedestrian movement (save the gradual accumulation on the sidewalk) followed by the crossing of large, dense groups. That this phase is reserved exclusively for pedestrians in all directions allows some pedestrians to cross diagonally, merging or diverging from the primarily bidirectional flow on the crosswalk itself.

Of course, being an intersection, the recorded movement is not exclusively pedestrian : each pedestrian phase has an associated motor-vehicle phase, and cyclists comfortably use the intersection in both. This multimodality impacts both the tracking and optimization processes in various ways ; these are examined in the sections where they occur.

3.3 Establishing the Ground-Truth

3.3.1 Sequence selection

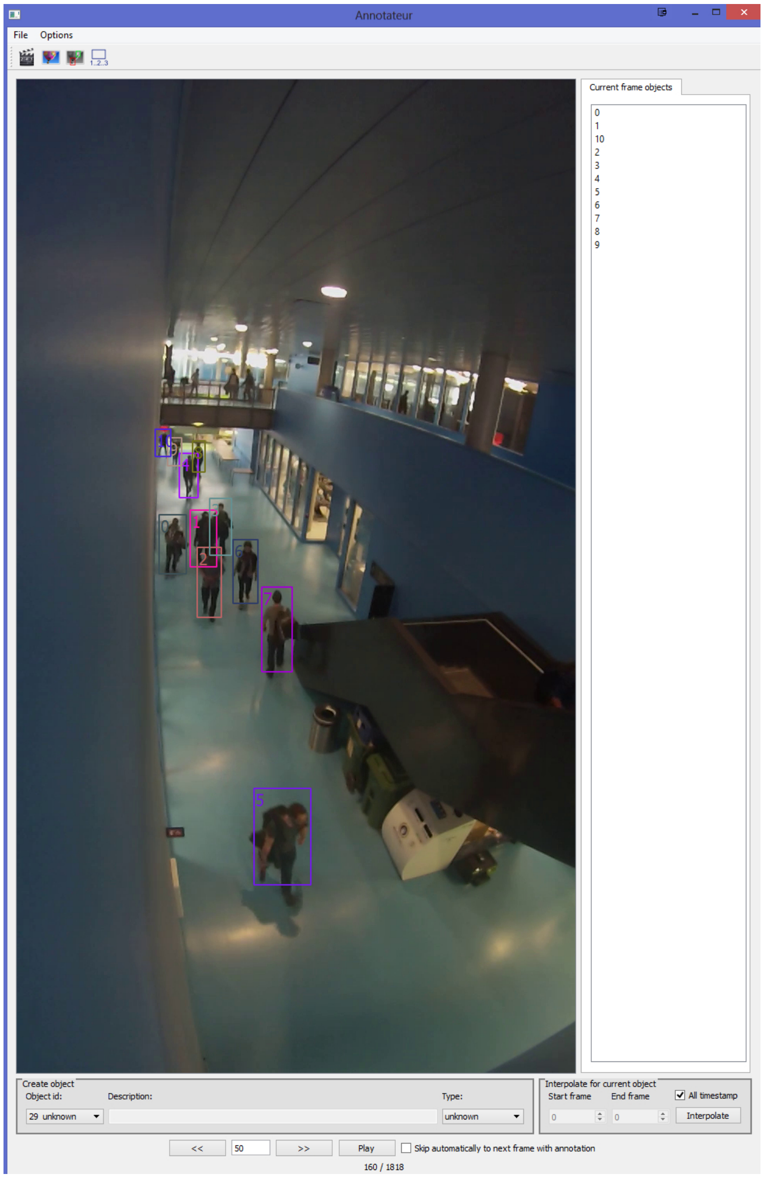
Short sequences were extracted from the full videos of each location in order to manually establish their ground-truth trajectories. It was observed in section 2.2.3 that trackers best calibrated for scenes of high density (and therefore higher tracking difficulty) were not necessarily well suited to simpler videos. The extracted sequences, therefore, were selected so as to contain the full variance of the represented cases, corresponding to the periods immediately before and after the increases in pedestrian flow noted in the previous section.

It was assumed that the number of iterations required for convergence of the optimization algorithm would be independent of scene length, and that consequently the total optimization time would be primarily a linear function of the duration of the calibration sequences. A sequence length of one minute for each scene was judged to be a reasonable compromise between optimization speed and representativity, and so was used in all scenes.

However, another concern with optimizing over relatively short videos is the risk of overfitting the tracker to said videos alone, thus obtaining parameters which are not particularly optimal for the scenes as a whole. Beyond ensuring temporal heterogeneity within the sequences (or simply extending sequence length to some unknown threshold) no clear solution to this problem presented itself. Instead, a second sequence was extracted from each scene in order to serve as a test case and so quantify any overfitting that may occur. In the New York and subway scenes, test sequences were simply a second minute of video taken from the same camera. In the Polytechnique scene, a second camera filmed the same area from a similar angle (albeit a different position) and the test sequence was therefore extracted from said camera in order to test whether overfitting might occur solely due to a specific perspective.

3.3.2 Ground-Truth Annotation

The Urban Tracker package includes a tool for the annotation of videos, specifically designed for establishing ground-truth for tracking applications. This tool was used for the entirety of sequences studied herein, and its interface is illustrated in figure 3.4. A notable advantage of this tool in comparison to other potential annotation methods is its automatic interpolation of positions if none was explicitly entered, thus allowing manual detection only every N frames and greatly shortening annotation time. The number of frames between detections varied by tracked object and observed movement complexity, but generally ranged between five and twenty.



0
1
10
2
3
4
5
6
7
8
9

Object id: 29 unknown

Description:

Type: unknown

Start frame 0

End frame 0

☑ All timestamp

50

□ Skip automatically to next frame with annotation

160 / 1818

Figure 3.4 Presentation of the annotation tool used to manually extract ground-truth trajectories : pedestrians are located by their bounding boxes, and identified by number according to the order in which they were tracked.

In the resulting ground-truth databases, pedestrians are represented by their bounding boxes (i.e. the smallest rectangle encompassing the entire pedestrian), stored as the video-space coordinates (in pixels) of two opposite corners in each frame, along with the unique identifier of the object being "tracked". Further information could be added at this state, such as homography transformations to real-world coordinates, using the center of bounding boxes for comparison with trackers that do no utilize bounding boxes, or classification of objects (e.g. vehicles or pedestrians). The utility of these is dependent on the tracker, however, so the ground-truth was maintained in its original form until deemed necessary by TrOPed.

A macroscopic overview of the calibration and test sequences used is presented in table 3.1.

Table 3.1 Macroscopic characterization of the test and calibration sequences. "Atypical" objects refer to those not moving in the bidirectional axis of highest flow in the sequence.

	PolyMtl		Subway		New York	
	Cal.	Test	Cal.	Test	Cal.	Test
Total ground-truth objects	32	31	53	33	87	64
Atypical objects (%)	41	35	40	55	43	50
Min. objects on screen at once	0	4	1	1	1	2
Max. objects on screen	13	10	11	7	35	17

3.4 Algorithm Inputs

The first iterations of TrOPed had tracker parameters hard-coded into the algorithm, with the user required only to select which of the two test trackers (TI or UT) to use and the test sequence. However, such an algorithm could hardly be considered generalizable. Therefore, current versions of the optimization framework achieve the same functionality while accepting tracker parameters and other requirements as inputs, entered into three setup files : *variable parameters*, *static parameters*, and a general *setup file*. Examples of these three files are presented in appendix A.

The latter *setup file* encodes the higher-level information required for the tracker in question to be run at all. This consists of the command(s) required to run the tracker (stored as strings as they would be entered on the command line) and the number and filenames of the configuration files used by the tracker. This file also holds the parameters and options of TrOPed, which are explored individually in the relevant sections.

Tracker parameters which are not to be optimized (as they do not impact tracker performance) but are nonetheless required for the tracker to run are considered *static parameters*, and are stored in their own input file. These include the input and output filenames, distortion coefficients of the video, options to display trajectories as they are tracked, and the use of analytical tools included with the tracker. They are stored simply as strings (e.g. "video-filename = test.mp4") preceded by the index number of the file they are to be written to as defined by the setup file.

Variable parameters are defined by their name (as understood by the tracker) and the index of the configuration file containing them, much like the static parameters, as well as the range and data type information presented in tables 2.2 and 2.3. In addition, this file contains the initial parameter values and instructions on how each is to be permuted by the algorithm.

Parameter updates are characterized by both the *method* by and the *extent* to which a parameter is to be changed at every iteration, both dependent on the expected sensitivity of the tracker to the variable in question. The first of these, the update method, defines the operation applied to the variable. For the majority of parameters which can be assumed to have a linear (or nearly linear) effect on the portion of the tracker they regulate (e.g. adding Δx to parameter X_i has a similar effect regardless of the value of X_i) simple addition/subtraction is used. Boolean parameters are, functionally, treated as the trivial case of the prior operation where the parameter is an integer with range [0,1], but are assigned their own class given that they may be encoded non-numerically (e.g. [true,false]). A final operation was specifically added for TI's feature-quality parameter, the effect of which appears to be logarithmic, and for which parameter updates are therefore performed via multiplication and division. Though these three operations were judged adequate for the parameters of TI and UT, other trackers may have parameters with sensitivity distributions outside those presently covered (e.g. non-linear distributions not centered at zero, or a categorial variable with more than two possible states). Operations which satisfactorily represent such distributions are, however, simple to append to the existing list.

The *extent* of parameter changes defines the maximum change to bring to a parameter, relative to others (obviously, it has no effect on Boolean parameters). This signifies that if, for example, parameter A is expected to have three times the effect of B after an identical change, the *extent* of A should be a third of that of B. This is illustrated in table 3.2 below :

Table 3.2 Parameter update equations for variable X between iterations i and i+1 in the three operations currently included in TrOPed and defined in the *variable parameters* file. The above equations are for floating-point variables ; other data types use slightly modified versions.

ID	Description	Operation
add	addition/subtraction	$X_{i+1} = X_i + (r * S * rand(0, extent))$
ratio	multiplication/division	$X_{i+1} = X_i * (r * rand(0, extent))^S$
bool	boolean parameter	if $X_i = 0, X_{i+1} = 1$, else $X_{i+1} = 0$

In these equations, r is the relative change, a multiplicative factor determined either manually or by the algorithm, formally presented in section 3.10. S is randomly selected from either 1 or -1 in order to determine whether the parameter value is increased or decreased ; if the parameter is already at its minimum or maximum value, S is fixed accordingly.

The resulting configuration file for parameters to be optimized is summarized in table 3.3.

Table 3.3 Structure of the input file for tracker parameters to be optimized.

configID	param. name	type	operation	init. value	min.	max.	*extent*
0	example-parameter-1	float	ratio	0.2	0	1	2
1	example-parameter-2	int	add	3	1	20	1

3.5 Homography Parameters

As briefly mentioned in section 2.2.4, tracks in the image-space of a video are converted into trajectories in real-world coordinates via their joint homography. This is performed through transformation of coordinates in the former space by multiplication with a *homography matrix* (functionally a specific application of a rotation matrix) generated by association of at least four non-collinear points which are manually located in both spaces. Use of homography in TrOPed is optional. When it is used, the matrix is calculated using the OpenCV toolset (OpenCV, 2008) and applied using the standard equation below (Kriegman, 2007) :

$$p_{image} = \begin{bmatrix} x_{image} \\ y_{image} \\ 1 \end{bmatrix} \tag{3.1}$$

$$p'_{world} = H_{image-world} p_{image} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} w' x_{world} \\ w' y_{world} \\ w' \end{bmatrix} \tag{3.2}$$

$$p_{world} = \frac{p'_{world}}{w'} \tag{3.3}$$

where p_{image} and p_{world} are point coordinates in the image and world spaces, respectively, $H_{image-world}$ the homography matrix, and *w'* a multiplicative factor derived from said matrix.

In the vast majority of applications, including multimodal tracking in TI, the world-space corresponds to the horizontal plane of the ground. Though this leads to some positional error (due to the underlying assumption that the tracked object exists only within the two dimensional surface plane) the effect is often negligible due to a combination of large studied spaces, the width of vehicles, and elevated camera angles. Unfortunately, these mitigating factors do not necessarily apply to pedestrian studies. Cameras aimed at pedestrian-dominant areas are often only three or four meters above the ground and far closer to the targets than their vehicle-tracking counterparts. In addition, pedestrians are substantially taller than they are wide, a fact which is exacerbated by the portions of the body with the least relative motion (and thereby the easiest to track using feature- or detection-based methods) are also the most elevated. Taken together, these issues may lead to far greater positional tracking errors, both relative to the horizontal space occupied by the targets and in absolute terms (see figure 3.5).

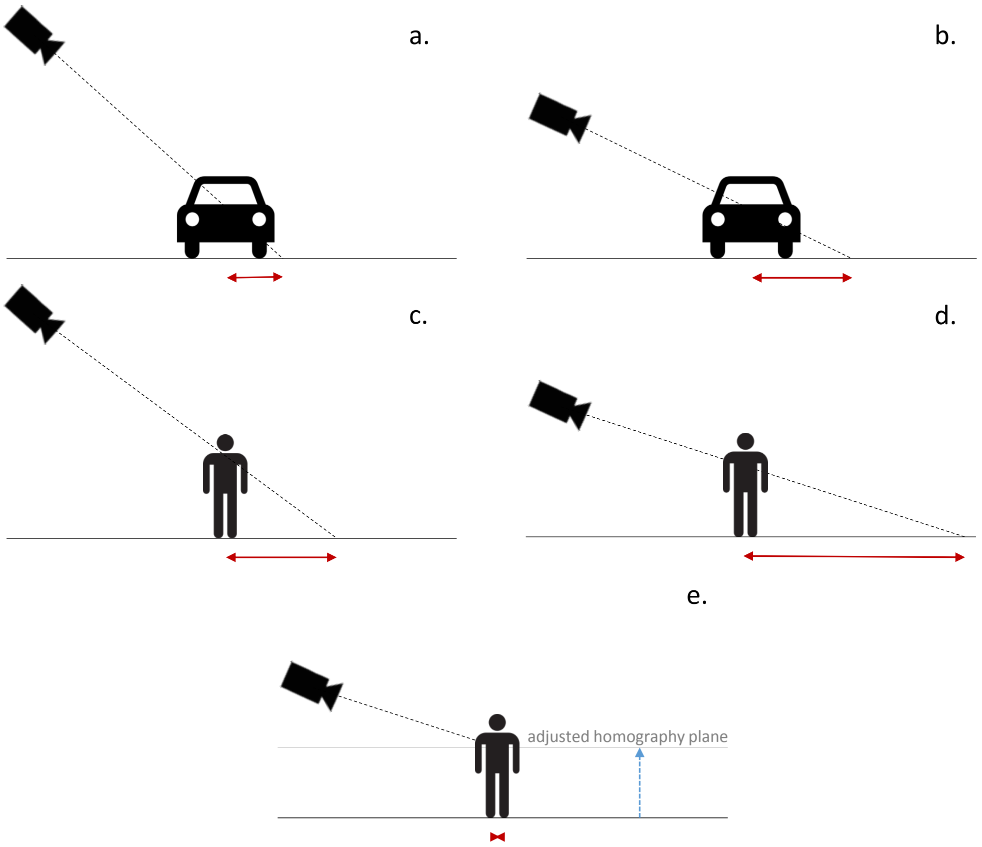


Figure 3.5 Illustration of the effects of low recording angles (b) the height of pedestrians (c) and both (d) on positional tracking error, relative to the common case of vehicle tracking in (a). (e) presents the proposed solution : optimization of the elevation of the homography plane to correspond with the elevation at which pedestrians are most commonly detected.

This problem would be difficult to resolve in a multimodal setting without classification of each tracked object. However, there is no reason (other than convenience) that the homography plane should be that of the ground - simply elevating the plane to the height at which detections are expected to occur would reduce the error with no ill effect on the coordinates of fixed objects. Therefore, in order to both quantify and attempt to resolve this issue, the point correspondence tool from TI was taken and expanded to allow the input of an additional set of points in the image space corresponding to points directly vertical to those four placed on the ground plane (in all the cases examined here, the elevation of these points was set at approximately 1.5 meters). The point eventually used to calculate homography is interpolated between each of the four pairs ; the elevation of each is included as an additional, internal parameter of TrOPed (see figure 3.6). Although optimization would likely be accelerated by the use of a single elevation parameter, individual ones were kept in place in order to compensate for any punctual errors caused by the manual input of points ; fortunately, testing with and without these parameters revealed no notable difference in convergence time.

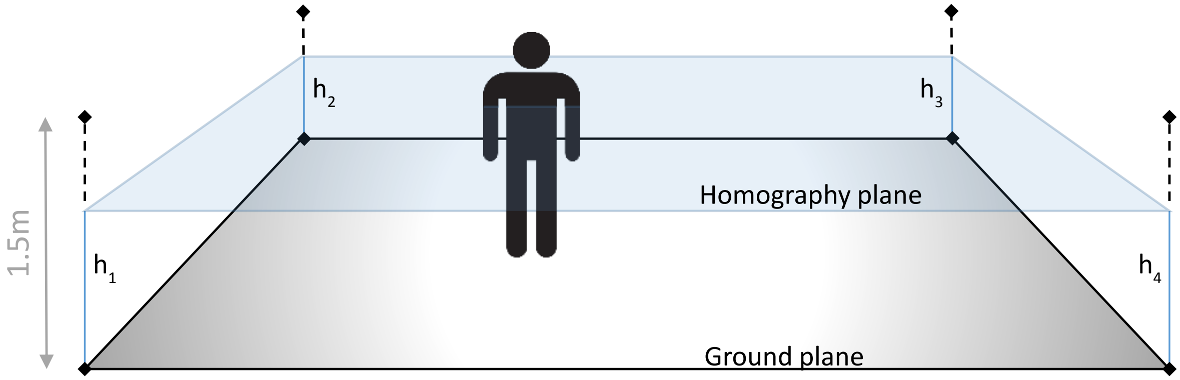


Figure 3.6 Schematic representation of the added homography-elevation parameters and their effect. Note that the elevations are not necessarily equal for each h_i.

Of course, pedestrians are neither of uniform height, nor are they likely to be detected at the same position on their body ; "optimal" homography elevation, then, is only the best average elevation, though variance was observed to be limited. Differing detection heights become more problematic, however, when they occur between the ground-truth and the tracker output : the annotation tool used represents pedestrians by their bounding boxes, to which the only meaningful application of homography is to the center, usually located near the waist. Again in the interest of both quantifying and resolving the issue, this elevation difference was added as a final additional (and optional) optimization parameter, applied simply by addition to the previous four elevation parameters. In all, this resulted in five homography parameters to be calibrated by TrOPed and added to those already present in the trackers ; these are summarized in table 3.4.

Table 3.4 Variable parameters selected for optimization by TI and UT, as input into TrOPed. *prev* indicates a parameter dependent on the previous one to define either its maximum or minimum value.

	parameter	type	operation	init. value	min.	max.	*extent*
TRAFFIC INTELLIGENCE							
FEATURES	feature-quality	float	ratio	0.1	10^{-6}	1	2
	min-feature-distanceklt	float	add	5	0	10	1
	window-size	int	add	5	3	10	1
	pyramid-level	int	add	3	1	-	1
	ndisplacement	int	add	2	2	4	1
	min-feature-displacement	float	add	0.05	0	0.1	0.02
	acceleration-bound	float	add	1.5	1	3	0.5
	deviation-bound	float	add	0.3	0	1	0.2
	smoothing-halfwidth	int	add	6	0	11	2
	min-tracking-error	float	add	0.1	0.01	0.3	0.08
	min-feature-time	int	add	15	5	25	5
OBJECTS	mm-connection-distance	float	add	1	0.5	2	0.4
	mm-segmentation-distance	float	0.75	0.1	prev	0.4	
	min-features-group	float	add	3	1	4	1
HOMOGRAPHY	elevation-1	float	add	1	0	1.5	0.2
	elevation-2	float	add	1	0	1.5	0.2
	elevation-3	float	add	1	0	1.5	0.2
	elevation-4	float	add	1	0	1.5	0.2
	ΔGTelev	float	add	0	0	0.5	0.1
URBAN TRACKER							
BG SUBTRACTION	bgs-minimum-blob-size	int	add	20	10	-	5
	max-lost-frame	int	add	5	1	-	2
	max-seg-dist	float	add	0.3	0	1	0.2
	max-hypothesis	int	add	4	1	-	2
	minimum-match-between-blobs	int	add	3	1	-	1
FEATURES	brisk-threshold	int	add	8	1	20	3
	brisk-octave	int	add	2	1	5	1
	match-ratio	float	ratio	0.2	0	1	0.1
OTHER	urban-islated-shadow-removal	bool	bool				
	verify-reentering-object	bool	bool				
	bgs-remove-ghost	bool	bool				
HOMOGRAPHY	elevation-1	float	add	1	0	1.5	0.2
	elevation-2	float	add	1	0	1.5	0.2
	elevation-3	float	add	1	0	1.5	0.2
	elevation-4	float	add	1	0	1.5	0.2

The above sections present the preliminary work required of the user in order to begin utilizing TrOPed ; at this point, barring potential calibration, the optimization is fully automated. The following sections (3.6 through 3.8) detail a single iteration of the optimization framework, which begins with running the tracker on the calibration video.

3.6 Metric Selection and Performance Evaluation

3.6.1 Metric Selection

Current video tracking metrics were presented in section 2.3. Of these, two (ATA and the CLEAR MOT metrics MOTA and MOTP) were found to adequately evaluate both the accuracy and precision of tracker outputs, as well as being in sufficiently widespread use to provide a generalizable basis for performance comparison. Of these, ATA would on the surface appear to be the best choice for optimization, due in no small part to it having been specifically designed with such use in mind : it conveniently and intrinsically combines both precision and accuracy metrics, and evaluates them at the track level. In contrast, MOTA and MOTP match ground-truth and tracker output at each frame individually, and their use in optimization requires modification in order to produce a single, combined metric.

Despite these issues, however, the CLEAR MOT metrics were selected for use in TrOPed for the three following reasons. First, and most importantly, the combination of two seperate metrics gives one the flexibility to adjust their relative importance in performance evaluation for optimization. Not only does this allow adjustable prioritization of one over the other for the algorithm in general, but also adjustment mid-process. A detailed analysis of the effects of the relative weights of MOTA and MOTP are presented in the following chapter. However, one notable advantage of the ability to adjust them was discovered during the very first tests of the algorithm : the avoidance of local maxima by greater algorithm movement through the search space.

Second, their explicit distinction of precision and accuracy has led to them being utilized in a larger number of tracker evaluations, which in turn facilitates comparison of post-optimization performance with that in the literature.

Third matching at a trajectory level, while more rigorous than the alternative, is most commonly (if not solely - no counter-examples could be found) performed on a one-to-one basis. Though such matching is most likely adequate in a majority of cases, in those where a pedestrian is tracked by multiple partial trajectories ATA scores would be far more heavily penalized than MOTA. Such behavior is acceptable at higher accuracies, but given both the known difficulty of pedestrian tracking and the low expected performance during a majority of the optimization procedure, overpenalization was judged likely to effectively close off potential avenues for performance enhancement.

Indeed, while high accuracy measures by their very nature require that the generated trajectories closely resemble the real trajectories of the tracked objects, precision measures (regardless of metric) measure only the proximity of individual ground-truth positions to the nearest detection. This leads to the possibility of perfect measured precision in a trivial case : detections at every pixel in every frame. At sufficiently low accuracies - such as those likely to be encountered if the initial parameter values are not calibrated in advance - affording a non-zero weight to precision can therefore lead to the optimization algorithm taking the "easier" energy-minimization path of simply generating as much noise as possible, despite this also generating the worst possible accuracy scores.

The only method found to recover from these local maxima was also found to be the most effective way of avoiding them : optimization based solely on accuracy at early iterations of the algorithm. As this would be impossible using ATA, MOTA and MOTP remained the sole candidate metrics and were implemented in TrOPed.

Performance Evaluation

Tracker performance evaluation is performed in two phases : *matchmaking* of trajectories to the ground-truth, and the actual *evaluation* of performance scores.

The second of these is relatively simple. MOTA is evaluated by direct application of equation 2.7. MOTP required some modification in order to be measured on the same scale and direction as MOTA and thus allow combination into a single energy score ; this was achieved simply by normalizing precision to between 0 and 1, and then subtracting the resulting value from unity. This resulted in equation 3.4.

$$MOTP' = 1 - \frac{\frac{\sum_{i,t} d_t^i}{\sum_t c_t}}{matchDistance} \tag{3.4}$$

where *matchDistance* is the maximum distance, in either meters or pixels (depending on whether homography is used) between which a ground-truth - detection pair can be considered a match. As no matches can be made beyond this distance, it also represents the maximum possible positional error for "good" trajectories. Maximum match distance was set at 0.8 meters in all cases, based on the average stride length, human height, and the approximate size of the "comfort zone" maintained by pedestrians in cases of "normal flow". Such flow has no common formal definition, and so is defined here as flow within the speed and density ranges typically observed in pedestrian areas, or outside of panic or other extraordinary conditions.

An outline for the CLEAR MOT matchmaking algorithm is provided in Keni and Rainer (2008). When interpreted literally, it consists summarily of the association at each frame of each ground-truth track to the nearest tracker detection, so long as the latter lies within the maximum matching distance. Such a direct interpretation (found, by comparison of results, to be utilized in evaluations of both UT and TI in Jodoin *et al.* (2014)) is simple to implement. Unfortunately, however, it can lead to underrepresentation of overgrouping errors in the case where two pedestrians move very closely together, as both are considered correctly matched despite sharing a single track. A more rigorous matchmaking algorithm, inspired by Milan *et al.* (2013), was later attempted and is presented in section 5.2. The initial matching algorithm is presented below.

```
matcheTable = empty table of matches;
for Each frame of the video sequence do
    frameMatches = list of unassociated ground-truth objects in the frame;
    for Each ground-truth object in the current frame do
        for Each hypothesis object in the current frame do
            dist = distance between the ground-truth and hypothesis tracks;
            if dist < maxMatchingDistance and dist < previousBestMatch for the
            current ground-truth object then
                associate the two objects in matchTable;
            end
        end
    end
end
```

Algorithm 2: Initial matchmaking algorithm for performance evaluation.

From the resulting table and the number of tracker detections, the components of equation 2.7 are determined as follow :

- m_t (misses) is the number of unassociated ground-truth tracks.
- f_{pt} (false positives) is the number of matched tracks subtracted from the total number of tracker detections.
- mm_e (mismatches) is the number of instances where a ground-truth object is associated to different tracker detections in subsequent frames.
- g_t (ground-truth tracks) is simply the total number of ground-truth points.

This performance evaluation method was validated against that used in Jodoin *et al.* (2014), which conveniently compared TI and UT performances in an expertimental pedestrian setting, and for which the utilized ground-truth tracks were provided alongside the utilized (manually calibrated) parameters and the matching distance used. The results are shown in table 3.5. Produced MOTAs were nearly identical, albeit slightly lower for both trackers when evaluated by TrOPed ; this difference is likely attributable to the use of points rather than bounding boxes for matching.

Table 3.5 MOTA as evaluated by TrOPed and Jodoin *et al.* (2014) for both studied trackers. Scene, tracker parameters, matching distance and ground-truth were provided from the latter authors for testing.

	MOTA	
Tracker	TrOPed	Jodoin *et al.* (2014)
TI	0.67	0.69
UT	0.92	0.94

3.7 Optimization Algorithm

For the reasons presented in section 2.4, a simulated annealing approach was adopted for the optimization of tracker parameters. Implementation of the method - i.e. the specific equations used - vary ; those utilized in TrOPed were taken and adapted from Ross (1997). The optimization procedure consists of three distinct steps at every iteration : energy evaluation, temperature calculation, and the decision of whether or not to move to the latest energy state.

From a practical standpoint, *energy evaluation* requires the entire procedure presented thus far, i.e. the running of the tracker for a given set of parameters and the calculation of MOTA and MOTP for the resulting tracks. The following steps, however, require the concatenation of the two scores into the objective function V, calculated simply as :

$$V = w_{MOTA} * MOTA + w_{MOTP} * MOTP \tag{3.5}$$

where w_{MOTA} and w_{MOTP} are the weights accorded to MOTA and MOTP, respectively. Given that the combined weight of both metrics was set to unity, equation 3.5 was simplified for convenience to :

$$V = w_{MOTA} * MOTA + (1 - w_{MOTA}) * MOTP \tag{3.6}$$

As the metallurgical analogy suggests, the probability that the algorithm will move to a higher energy state (in this case, a lower combined MOTA/MOTP) is dependent on the current *temperature* ($T°$) of the system. In simulated annealing, said temperature is dependent on the current iteration, and is calculated as :

$$T = T_{init} - \lambda * ln(i + 1) \tag{3.7}$$

where T°_{init} is the initial temperature, and λ is a scale factor controlling the rate of temperature decrease and, ultimately, the maximum number of iterations.

Finally, the *probability to move* to a new energy *V'* is defined in Ross (1997) by :

$$P_{move} = min\left(\frac{e^{-TV'}}{e^{-TV}}, 1\right) \tag{3.8}$$

This equation, however, presents two problems in the current context. First, and most obviously, it assumes that one seeks to find a minimum energy, whereas TrOPed is attempting to maximize MOTA and MOTP. While this could be corrected by simply inverting the metrics, eliminating the minus term from the equation achieves the same result whilst making it easier to interpret the resulting scores as the optimization progresses. Second, as the probability to move to a new, lower performance parameter set is directly dependent on T, both T_{init} and λ must be calibrated to ensure that T allows for reasonable regressions throughout the optimization process - particularely given that what constitutes a "reasonable" regression is difficult to define beyond that which leads to consistent and efficient optimization.

Once the above three steps have been completed, the algorithm selects either *V* or *V'* for use as the basis for comparison and parameter update *V* in the following iteration.

3.8 State-Generation Function

The fundamental state-generation functions in use by TrOPed were presented in section 3.4. At every iteration, a random number of parameters (between one and $max_{changes}$, a parameter set in the setup file) are modified. With the exception of Boolean parameters, each individual parameter update is dependent on the term :

$$random(0, r * extent) \tag{3.9}$$

where *extent* is the predefined maximum parameter modification size and r a limiting factor between 0 and 1 of *extent*. r was implemented following an observation during initial tests of TrOPed : if *extent* is too small, early optimization is exceedingly slow ; if it is too large, the algorithm will eventually plateau, circling but not locating the precise, optimal values. Instead of attempting to identify central, ideal values of *extent* for every parameter (which would have to be performed anew when a new tracker is studied) the range of possible parameter changes is reduced whenever the algorithm stagnates for a number of iterations N, dependent on the current temperature of the system :

```
lastChange = iteration at which either a new best solution was found, or $r$ was last
modified;
i = the current iteration;
threshold = T*$N_i$ # Number of iterations during which the algorithm continues to
run without adjustment;
if (i - lastImprovement) >= threshold then
    r = r/2;
    lastChange = i;
end
```

Algorithm 3: Adjustment of parameter change size during optimization.

The above algorithm obviates the need to manually adjust r if TrOPed is found to stagnate near a certain value, as occurred regularly in early tests. It also removes the requirement for careful selection of the *extent* value for each parameter, as these can hence be heuristically assigned values corresponding to the relatively large parameter changes one would make during manual calibration of the tracker.

Globally, the resultant behavior was judged to adequately complement the basic optimization method : simulated annealing itself minimizes the chances of the algorithm getting "caught" near a local maximum, while the gradual reduction of parameter change size ensures both a broad initial sweep and a more refined search during later stages.

3.9 Final Algorithm

Once the next iteration's parameters have been generated, new tracker configuration files are written and the tracker is run anew on the test sequence, thus restarting the cycle. The algorithm continues until either the temperature reaches zero or it is stopped manually. The latter case is more likely : as convergence times can vary greatly (see chapter 4) it is prudent to allow a greater number of iterations than strictly necessary.

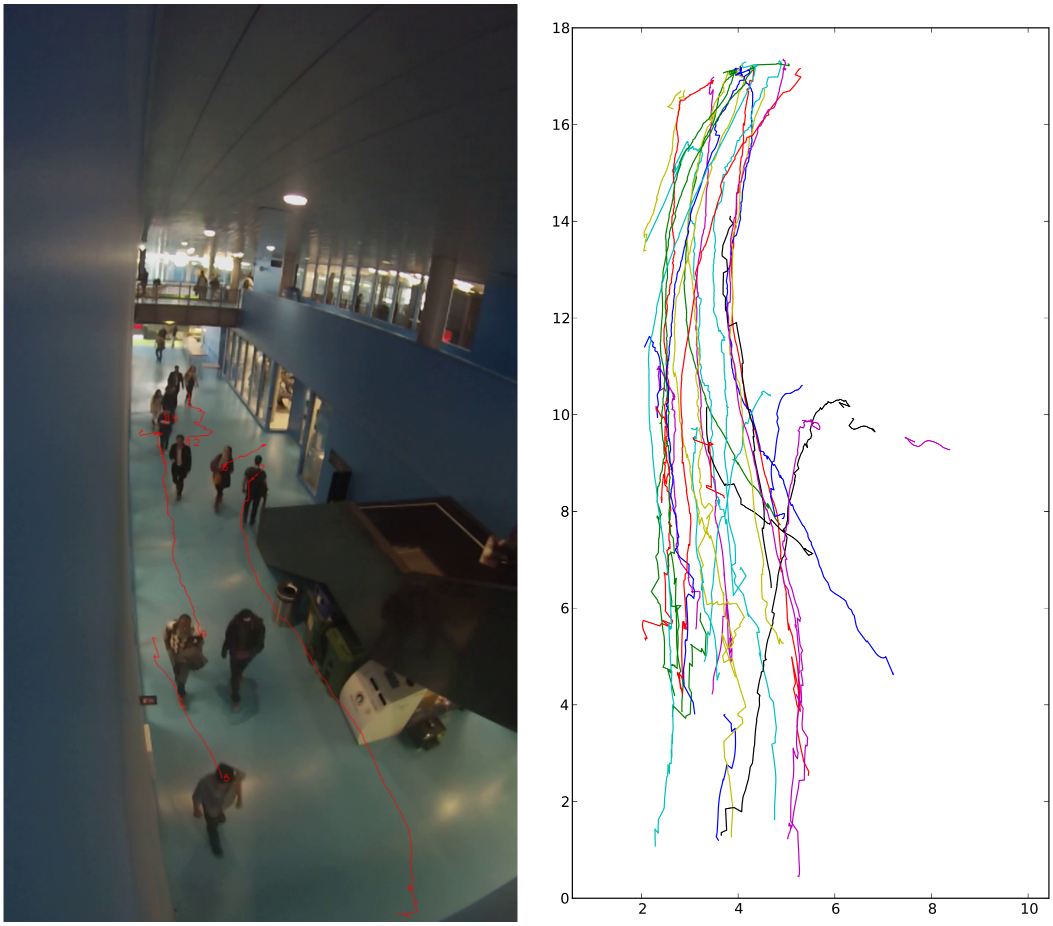


Figure 3.7 Example visualization of mid-optimization tracks in the Polytechnique corridor sequence.

Without the use of temperature, however, there is no clear, rigorous way to determine if optimization is complete. One alternative would be to use the parameter update factor r, stipulating that once it reaches a sufficiently small order of magnitude further parameter updates would be meaningless. Unfortunately, what that size would be is completely dependent upon the nature of the affected parameter and the predefined *extent* value. Instead, it was decided that the easiest method was simply to make a qualitative judgment of how likely better-performing parameters are to be found based on the prior behavior of the algorithm. It should be noted that such judgment is precluded from the algorithm itself ; as mentioned in section 2.4, subsequent iterations form a Markov chain, and are thus independent of all but the single preceding iteration.

In the interest of facilitating said judgment, complete information regarding all prior iterations is permanently saved before another is started. This output file contains parameter values, performance metrics (including the MOTA sub-metrics, e.g. the numbers of misses and false positives) and the iteration at which performance was last improved ; if restarted, TrOPed will use the parameters from this iteration, thus allowing manual correction if the algorithm is thought to diverge too far from the expected maximum performance. Furthermore, upon identification of a new maximum the source parameters are saved in separate configuration files for later use, as is the associated tracker output for visual validation (see figure 3.7).

3.10 Calibration, or : the tragic irony of building a parameter optimization algorithm that itself has twelve parameters

A central problem of calibration tools in general - and the modified simulated annealing algorithm used in TrOPed in particular - is the fact that these tools themselves require calibration. This problem is wrought with the same difficulties as listed in section 1.2, further compounded by the substantial time it would take to test individual calibration settings. Fortunately, these concerns are offset by inability of simulated annealing to produce a complete, optimal solution ; one can expect only an approximation of the ideal parameters, achieved in a reasonable amount of time. Good, reasonable algorithm parameters, therefore, could be constrained only to meet the following four requirements :

- A reasonable *convergence time*, measured in tens or hundreds (at most) iterations : While admittedly vague, the stochasticity of the simulated annealing algorithm, paired with the wide range of cases and trackers it can be applied to, makes defining a more precise threshold impossible.
- The ability to *consistently find similar maxima* : The inherent stochasticity of simulated annealing, found both in the "random walk" selection of parameter sets to move to and the new iterations of previous solutions, makes the path of every optimization run through the search space unique. Despite this, the algorithm must consistently find similar solutions for the same input data and tracker if there is to be any confidence that the ultimate solution

is an adequate approximation of the true optimum.

- *Similar results regardless of starting point* : This is not only convenient from the user's perspective, negating the need for careful parameter selection in advance, but helps ensure the algorithm is not easily sidetracked by local maxima.
- The ability to *meet the above requirements for a broad range of trackers and cases* : Applying TrOPed to particularly complex trackers or difficult video data is liable to take longer no matter the parameters selected. However, recalibration should not be required for the previous three conditions to be respected.

Calibration was performed, in part, in parallel with the development and testing of TrOPed, using the three cases described in section 3.2.2. Indeed, it is these tests that led to the consolidation of previously-existing parameters into more parsimonious implementations, including most notably the r parameter in the state-generation functions which replaced the predefinition of specific optimization phases. The primary means of calibration was, however, a more systematic testing and adjustment of the algorithm's parameters, utilizing the Polytechnique atrium sequence.

Briefly presented in section 3.6.1, the atrium sequence is a recording of several volunteers asked to perform a variety of tasks specifically meant to showcase the features of Urban Tracker. Said tasks range from typical pedestrian behaviors (moving in groups, crossing paths) to more exceptional but still plausible actions such as dropping (and later picking up) items or ceasing movement and beginning to dance. It is worth noting that while the pedestrians' movements are complex from a tracking standpoint, the scene itself is relatively favorable for the task : the uniform, bright orange background provides excellent contrast, and densities remain very low throughout the sequence (see figure 3.8).

The atrium video was used for calibration for three reasons. First, though as stated above the sequence is relatively easy from a tracker standpoint, it contains sufficient complexity that optimization is worthwhile (i.e. performance remains sensitive to tracker parameters even at high MOTA, instead of plateauing). Second, its low resolution (800 by 600 pixels) translates to substantially faster computation times for both UT and TI, greatly facilitating repeated experimentation. Third, high performance parameters (MOTA of 0.93) were known from the outset, providing an initial benchmark for the evaluation of tracker outputs as well as the capacity to begin the algorithm at varying distances from the solution parameters.

Calibration was performed largely by trial and error, with adjustments made in accordance with the overall progress of the algorithm. The sensitivity of TrOPed to the individual parameters is examined in detail in the following chapter. Overall, initial tests simply failed

Figure 3.8 Example frame from the Polytechnique atrium sequence, used for calibration of TrOPed.

to converge towards a meaningful solution : the algorithm would chose to move towards lower-performance parameters too readily. With Traffic Intelligence, the first attempts with functioning parameters converged to a MOTA of 0.83 after approximately 1300 iterations ; after calibration, TrOPed produced parameters attaining a MOTA of 0.85 in only 550 iterations (for comparison, the manually selected values in Jodoin *et al.* (2014) attained MOTA of only 0.67). These results are presented in figure 3.9. Applied to Urban Tracker, these parameters produced a maximum MOTA of 0.96 in only 92 iterations. Final TrOPed parameter values are presented in table 3.6.

Table 3.6 Algorithm parameter values after manual calibration.

Parameter	Defined in :	Optimized value
λ	Eq. 3.7	0.4
T_{init}	Eq. 3.7	16
N_i	Sect. 3.7	75
$max_{changes}$	Sect. 3.8	4

If taken alone, the results in figure 3.9 could be attributed to the stochasticity of the state-generation function ; indeed, the rapid early performance increase in the final attempt (using manually optimized parameters) in figure 3.9 is most likely a result of luck rather than of the parameters themselves. However, it should be noted that these represent the two extremes of a series of attempts ; the intermediate steps are not presented only because parameters were adjusted mid-optimization, rendering their reporting meaningless. Additionally, algorithm parameters were furthervalidated during their application to the other test cases, the results of which are presented in the following chapter.

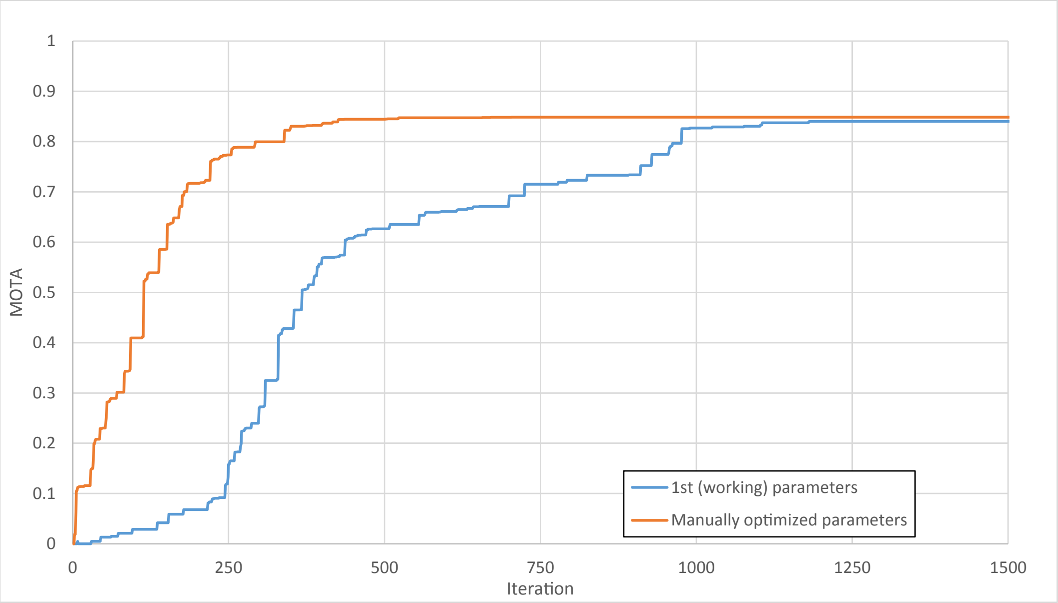


Figure 3.9 MOTA improvements during optimization with TrOPed on the atrium sequence, for both the first set of functioning parameters (i.e. those converging to a solution) and after manual calibration. In both cases, initial tracker parameters were set as far from the known optimum as possible (i.e. either their maximum or minimum values) so as to maximize convergence time.

3.10.1 Complete Algorithm Diagram

Figure 3.10 presents the complete flow diagram of TrOPed. Whereas figure 3.1 represented the initial outline of the optimization framework, this figure explicitely includes the steps detailed in this chapter.

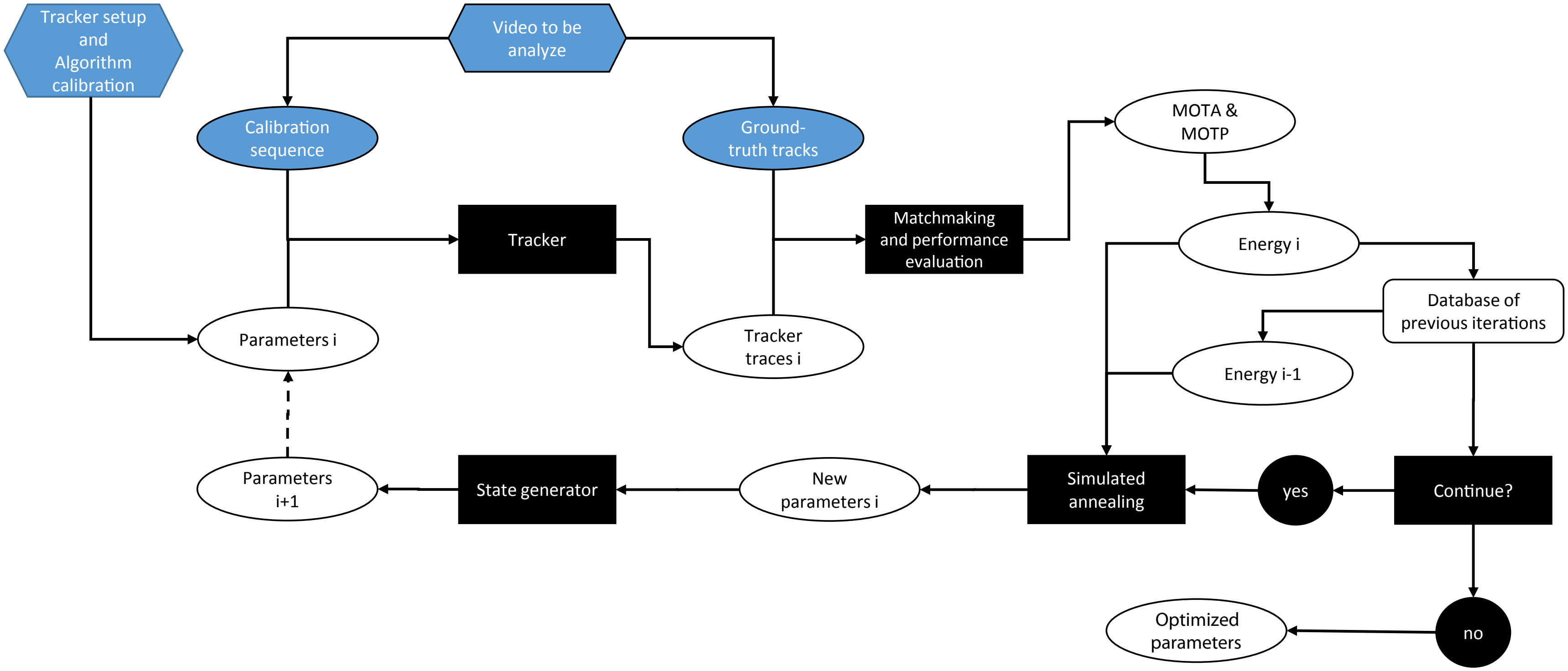


Figure 3.10 Complete flow diagram of TrOPed.

CHAPTER 4 OPTIMIZATION RESULTS

4.1 Overall Results

For both trackers and all three test cases, TrOPed was run on the calibration sequences until convergence of the algorithm, at which point the optimized parameters were applied to the test sequences. In each case, the algorithm was initialized at randomly selected parameters. As the MOTA and MOTP evaluation function included with UT was found in section 3.6.1 to produce different results than that developed for TrOPed, and TI lacks any such function, performance on the latter sequences was evaluated by simply running TrOPed for a single iteration on the new videos.

In order to provide a reasonable basis for comparison, performance was similarly evaluated for manually calibrated parameters. In the case of both the corridor and subway station sequences, said parameters were provided by the authors of each tracker, based on visual validation of the resultant tracks. For the later-obtained New York crosswalk video, this calibration was performed by the present author using the same methodology. Homography in all cases was defined at ground level when applying manually-calibrated parameters, as was performed by the authors. Performance for the randomly selected starting parameters, manually calibrated parameters and those produced by TrOPed are presented in tables 4.1 and 4.2.

Optimization proceeded in a manner similar to that observed during the calibration of TrOPed in section 3.10, with two notable exceptions : Urban Tracker consistently crashed when applied to the New York sequences, regardless of the parameters used, and an identical error occurred in the subway station test scene, though in this case only when using parameters manually selected for the calibration scene. Likely causes of these errors are discussed in section 4.3.3, but in short they appear to be a failing of the tracker itself rather than of the optimization algorithm. These exceptions aside, tracking accuracy was found to be markedly higher when utilizing TrOPed-optimized parameters than manually calibrated ones throughout all test cases and with both trackers. In addition, this occurred despite the random input parameters producing substantially weaker performance than those obtained via manual calibration.

The improvement proffered by TrOPed is more clearly visualized in figures 4.2 and 4.1. Though direct comparisons of MOTA values reveal little pattern, the number of committed errors (which, as described in equation 2.7, can be deduced simply by subtracting MOTA

Table 4.1 MOTA and MOTP performance of Traffic Intelligence before and after optimization, as well as for manually calibrated parameters. MOTP is presented according to the standard definition (average error in meters) rather than the normalized version utilized by TrOPed.

TRAFFIC INTELLIGENCE

	CALIBRATION SCENES					
Parameter selection	Random		Manual		Optimized	
	MOTA	**MOTP**	**MOTA**	**MOTP**	**MOTA**	**MOTP**
Poly. Corridor	-0.13	0.44	0.28	0.38	0.69	0.39
Subway Station	-1.51	0.49	-0.01	0.41	0.39	0.34
NY Crosswalk	0.01	0.46	0.24	0.49	0.68	0.30
	TEST SCENES					
	MOTA	**MOTP**	**MOTA**	**MOTP**	**MOTA**	**MOTP**
Poly. Corridor	-0.11	0.43	0.48	0.41	0.62	0.33
Subway Station	-1.63	0.47	-0.22	0.37	0.28	0.35
NY Crosswalk	0.02	0.37	0.26	0.46	0.63	0.27

Table 4.2 MOTA and MOTP performance of Urban Tracker before and after optimization, as well as for manually calibrated parameters. The absence of MOTP results in two subway station tests indicates that the tracker failed to produce sufficient meaningful tracks. As above, MOTP is presented according to the standard definition (average error in meters).

URBAN TRACKER

	CALIBRATION SCENE					
Parameter selection	Random		Manual		Optimized	
	MOTA	**MOTP**	**MOTA**	**MOTP**	**MOTA**	**MOTP**
Poly. Corridor	0.27	0.37	0.70	0.23	0.89	0.46
Subway Station	-1.22	-	-1.62	-	0.54	0.26
NY Crosswalk	*tracker failure*					
	TEST SCENES					
	MOTA	**MOTP**	**MOTA**	**MOTP**	**MOTA**	**MOTP**
Poly. Corridor	0.23	0.43	0.10	0.23	0.42	0.22
Subway Station	-1.43	-	N/A		0.38	0.27
NY Crosswalk	*tracker failure*					

scores from unity) was reduced in a relatively consistent manner. Indeed, the mean error reduction after optimization was 53%, and only a single outlier (TI applied to the Polytechnique corridor sequence, where error was reduced by only 27 %) was outside a +/- 15 % range.

Between the two trackers, error rate was reduced more strongly in TI (average of 63%) than

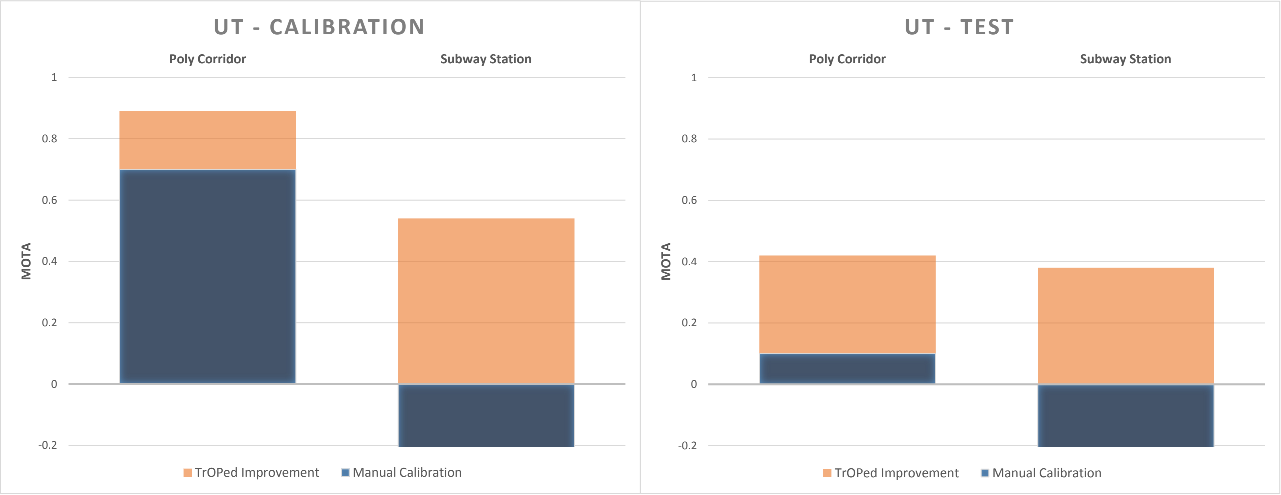


Figure 4.1 Comparison of manual and optimized parameter performance using Urban Tracker. Negative values are not fully displayed, and crashes on the New York sequences precluded performance evaluation.

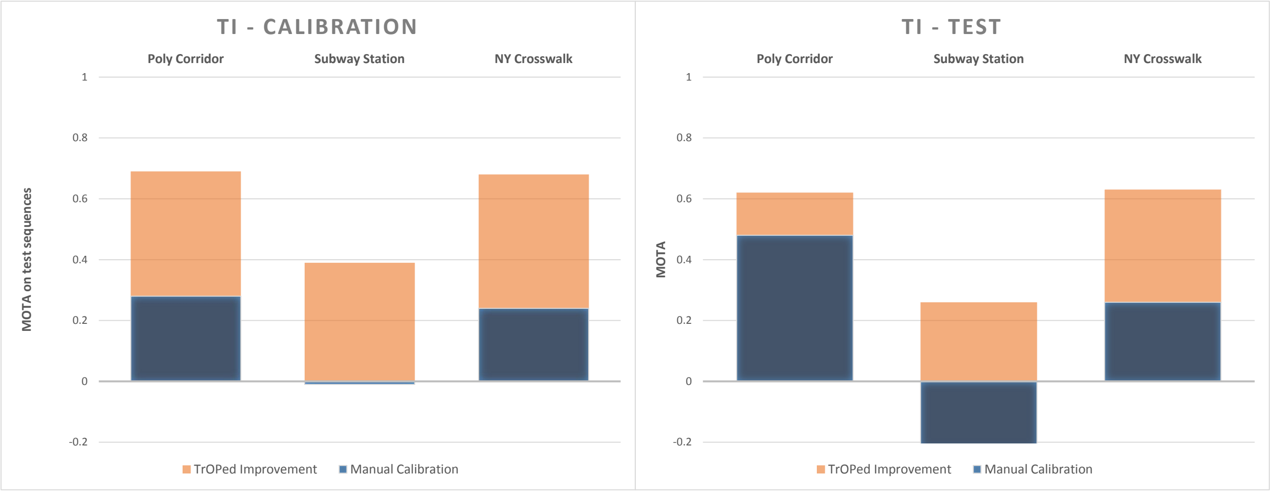


Figure 4.2 Comparison of manual and optimized parameter performance using Traffic Intelligence. Note that large negative values are not fully displayed.

with UT (48%). This difference is likely attributable to the larger number of parameters in TI : combined with the fact that a larger proportion of UT's parameters are either integer or Boolean values, this leads to a far smaller search-space for the latter, facilitating manual calibration (a more thorough examination of each tracker is presented in section 4.3).

A more important distinction should be made between performance on the calibration and test sequences. As mentioned in section 3.3, it is highly likely that an optimization algorithm such as that used in TrOPed would lead to overfitting of tracker parameters to the sequence used for calibration, leading to inferior performance on the video data as a whole, represented here by the test sequences.

To a certain extent, this appears to be the case : though the optimized parameters maintain superior MOTA scores, they are notably weaker in the test sequences for both trackers and in all scenes. However - and interestingly - the same effect can be observed to a lesser extent for the manually selected parameters : though there is larger variance in the changes to accuracy scores when applied to the test sequences (see figure 4.3) error increases an average of 33%, versus 67% for optimized parameters. Furthermore, these numbers exclude those cases where the manually chosen parameters failed to produce any meaningful result during testing. It appears that manual calibration may in certain cases also lead to overfitting.

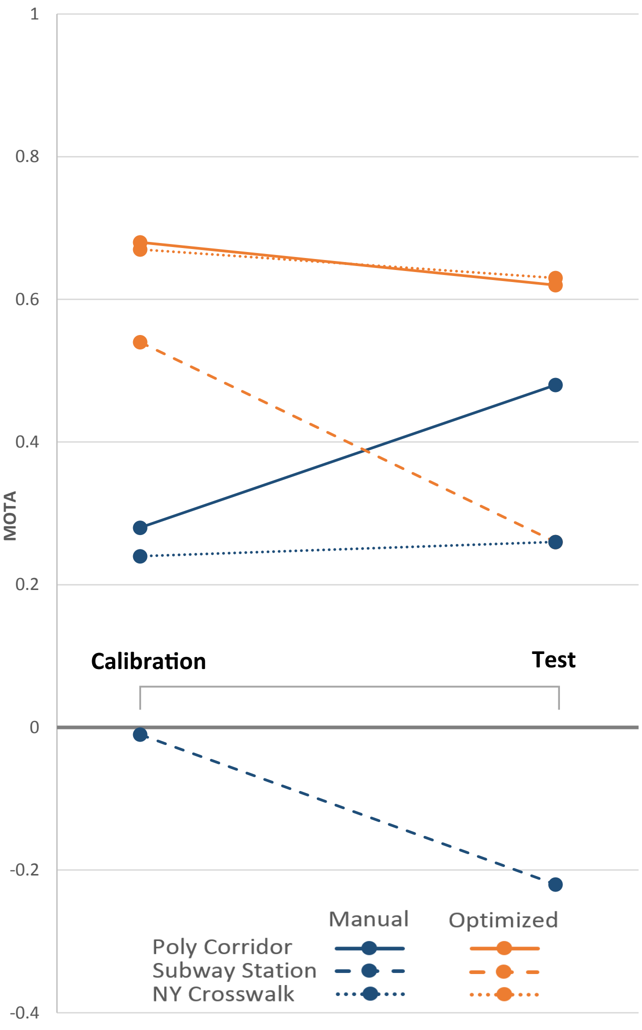


Figure 4.3 MOTA scores for Traffic Intelligence parameters on the calibration and test sequences.

Tracker precision, on the other hand, is less clearly advantaged by TrOPed optimization. While in the majority of cases (six of eight, again excluding those where manual calibration failed to produce meaningful results) MOTP was improved, on average the effect was only 3%, or 2.5 centimeters. This is far less than the difference in positional accuracy between the two trackers. UT's average error was 32% (or 25 cm) less than that of TI, likely attributable to the difference in tracking methodologies : UT detects entire moving objects, in contrast to the amalgamations of corners produced by TI, intuitively producing more precise tracks.

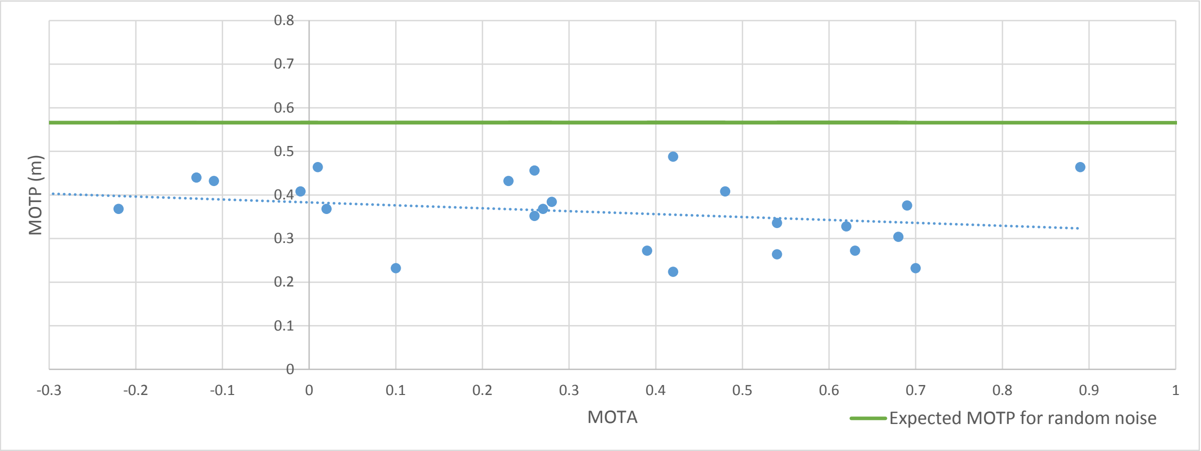


Figure 4.4 Relationship between MOTP and MOTA scores across all cases and both trackers.

Figure 4.4 further illustrates the relatively unpatterned behavior of MOTP. All tracks produced, including those from trials with random parameters, which are not included in the figure, demonstrated better precision than the 0.56 meter average positional error which would be expected by random chance (this number is not half of the matching distance - 0.4 meters - as might be expected, but the radius at which the interior and exterior areas are identical). While the figure makes no distinction between trackers or scenes, the trendline implies some relationship between precision and accuracy : for a given increase of MOTA, MOTP on average increases by 7% as much. Further data would be required in order to isolate this effect from other factors, but it corresponds to what could be expected, i.e. increased accuracy implies tracker trajectories better follow those of the ground-truth, leading to a smaller number of faulty associations which are less liable to be precise.

That MOTP is principally a function of the tracker and MOTA is further corroborated by the observed progression of the algorithm. As noted in section 3.6.1, TrOPed begins optimizing for MOTA alone, with MOTP only given non-zero weighting in later stages (practically, when convergence for MOTA has already been observed). The inclusion of precision had little effect, however, when the relative weight was maintained between 0 and 0.5 ; beyond this threshold, improvements to MOTP came only at the expense of accuracy, appearing to converge (albeit slowly) towards the "precise noise" scenario which the initial weighting sought to avoid.

Using weights of 0.6 and 0.4 for MOTA and MOTP, respectively, UT managed some slight overall improvement (on average, an increase of five centimeters of precision in exchange for a decrease of MOTA of 0.02), a tradeoff unlikely to be worthwhile. TI, in contrast, failed to measurably improve at all. Possible causes of these behaviors are examined in further detail in the following sections.

4.2 Observed Algorithm Behavior

One foreword for this section : at the time of the presented applications of the framework, TrOPed did not record times, either for individual iterations or for the optimization process as a whole. While this functionality has since been added, all times given in this section are therefore approximations based on observation.

The progress of the optimization algorithm for the calibration scenes, starting from random parameters, is shown in figures 4.5 and 4.6. The algorithm displayed notably different behavior when applied to each of the two trackers. Convergence was markedly slower (in terms of the number of iterations) with TI, taking between 500 iterations for the New York sequence and nearly 2500 iterations for the subway station scene. In contrast, UT's convergence occurred in less than 80 iterations for both scenes in which it functioned. Of course, though these values represent the iterations at which the final solution was achieved, in practice identifying these solutions as such required a substantially greater time : at multiple points in figure 4.5, TrOPed can be observed to have stagnated on a given point for upwards of 400 iterations, at MOTAs only a fraction of that eventually obtained.

The relative efficiency of the algorithm when optimizing UT is likely attributable to the reduced size of the search-space, as mentioned earlier. However, the low number of iterations belies the actual computing time required : as noted in section 2.2.4, single runs of UT can take upwards of sixty minutes, in contrast to one or two for TI. In addition, this time was observed to be highly dependent on the parameters used, varying between 20 (when few or no objects were detected) and 90 minutes (significant overdetection). Consequently, optimization of UT took a similar amount of time as TI ; these results are summarized in table 4.3.

Table 4.3 Convergence times of TrOPed for the test cases, both to best solution and to cessation of the algorithm. Times in hours are approximate.

Tracker	Scene	Best solution		Confirmation	
		iterations	hours	iterations	hours
TI	Poly. Corridor	1553	23	1800	26
	Subway Station	2379	34	2900	39
	NY Crosswalk	703	12	1100	16
UT	Poly. Corridor	51	26	70	28
	Subway Station	67	39	90	38

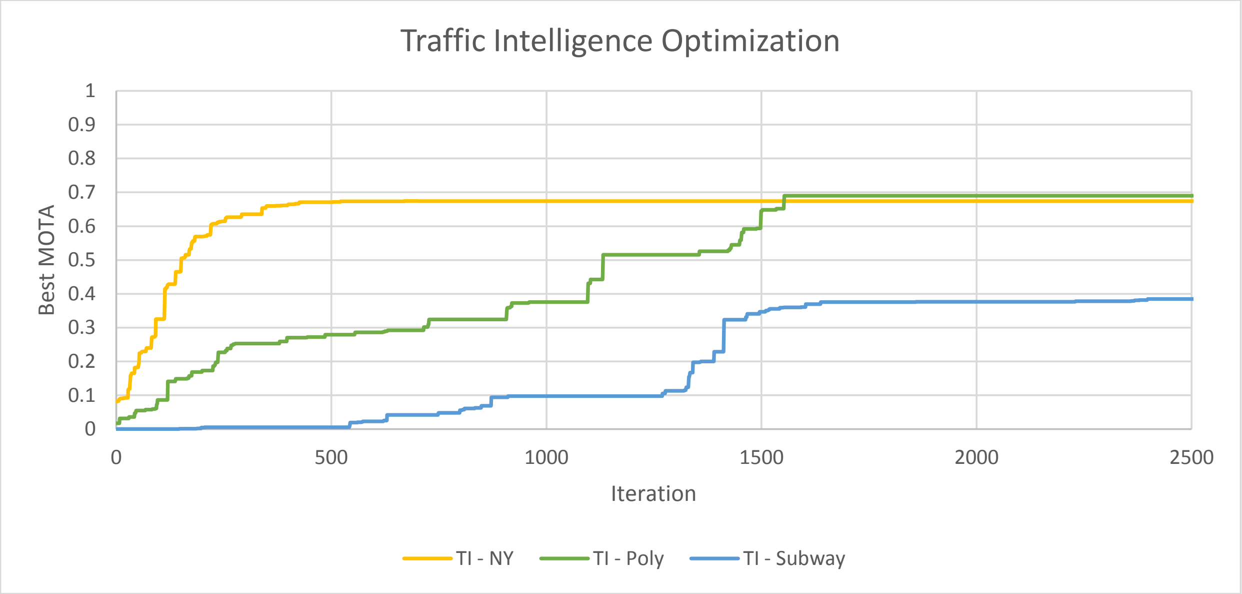


Figure 4.5 Evolution of last-best MOTA scores during optimization for the three calibration sequences with Traffic Intelligence.

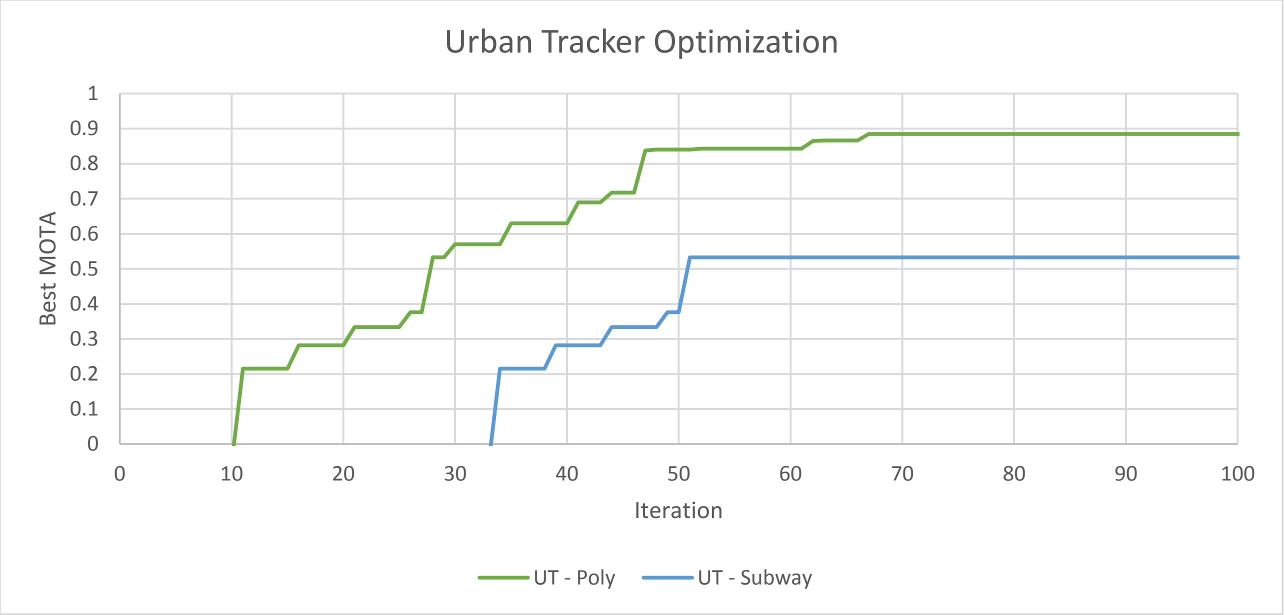


Figure 4.6 Evolution of last-best MOTA scores during optimization for the three calibration sequences with Urban Tracker. Values below zero are not shown.

In comparison to the behavior observed during the algorithm calibration on the atrium sequence, convergence time for the three case studies was notably longer. Though this, alone, may be attributable to the stochasticity of the state-generation function, it can also be observed that the number of iterations between subsequent performance improvements tended to be greater for the calibration scenes. One conjecture (albeit a difficult one to verify) is that this effect is in part a consequence of the relative difficulty of pedestrian tracking in the test cases in comparison with the atrium scene : more complex scenes may require a more refined range of parameters in order for tracking to produce meaningful - as well as optimal - results. Each new iteration of the parameters, therefore, would have a lower probability of producing improved performance, thus leading to the observed behavior.

4.2.1 Sensitivity to Starting Parameters

Alternatively, a more easily tested hypothesis is that convergence time depends, in part, on the performance of the starting parameters. Indeed, in figure 4.5 scenes with greater initial MOTA appear to converge faster (that the inverse is observed in figure 4.6 may then be a consequence of the lower number of iterations increasing the observable effect of state-generation stochasticity).

In practice, of course, there is little advantage to initializing the algorithm with random parameters : even the most lax approach would be to simply utilize those provided as defaults, which are at the very least likely to produce visually valid tracks. Another, more involved alternative is to first proceed to perform manual calibration, and then to use the obtained parameter values to initialize optimization. As such parameters had already been produced so as to provide a basis for comparison, this last approach was used in a reinitialization of TrOPed for the same scenes. The results of this second optimization are presented in figure 4.7.

With parameters already judged to produce suitable tracks, convergence time was more than halved for both scenes which had previously taken more than 1500 iterations. The New York crosswalk sequence was the one exception, though in its case the random and manually selected parameters were relatively similar. The subway station sequence, which previously took nearly 36 hours to converge, here took only 12, while the other two converged in approximately 8 hours.

Once again, convergence time appears to be positively correlated with the performance of the initial parameters. The duration of "stagnant" periods is also similar to that observed before, lending credence to the conjecture that scenes which took longer to optimize require a more restrained range of parameters than that used for calibration.

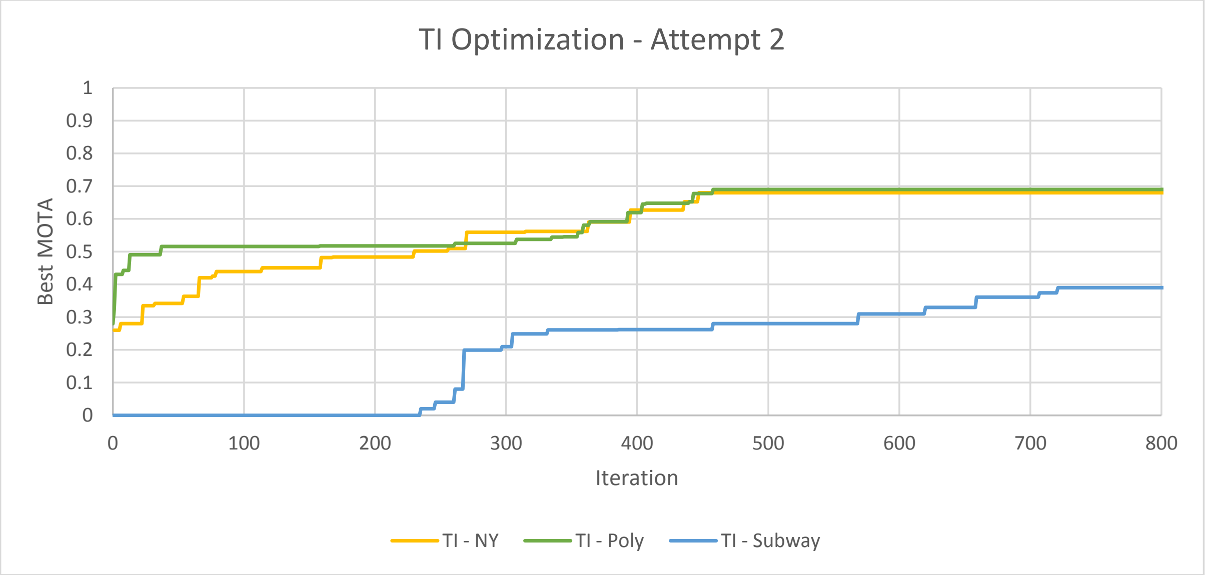


Figure 4.7 Evolution of MOTA scores during optimization of TI, initialized with manually-calibrated parameters.

It is important to note that the results obtained in this second optimization were nearly identical to those achieved previously : both MOTA and MOTP were within 0.01 of those of the initial tests. This would imply that TrOPed is capable of consistently attaining the global maxima - or, more conservatively, the same local maxima - regardless of starting parameters, a notion further corroborated by attempts to reinitialize the algorithm from intermediate points. Furthermore, that these maxima are more quickly obtained when using parameters which have been manually calibrated to produce tracks consistent with visual observation promises that the final parameters are not only measurably accurate but also meaningful in the context of pedestrian tracking.

The remainder of this chapter examines more specific aspects of the algorithm and tracker performances, namely the effect of the homography parameters and a more detailed overview of the results for each tracker and test case.

4.3 Tracker Performance

4.3.1 Homography Parameters

The results presented above all made use of the homography parameters described in 3.5. This includes those obtained from manually calibrated parameters, which utilized point-correspondences placed approximately 1.2 meters above ground level (or rather, at elevations set at exactly 1.2 meters within TrOPed, but interpolated between manually input points which could only be visually approximated to be at 0 and 1.5 meters). The full effect of these parameters, therefore, cannot be evaluated from the optimization data alone.

Overall, the prediction that 1.2 meters would be an appropriate height for pedestrian homography was reasonably close to the results : average elevation across the samples was 1.15 meters and the averages in specific sequences varied little, ranging from 1.04 meters in the Polytechnique corridor video to 1.26 for the New York crosswalk.

Within scenes, however, variance between individual point elevations was more marked (see figure 4.8). The subway station sequence, in particular, displayed a difference of more than two meters between its highest and lowest points - far greater than the difference that could be attributed to input error. Surprisingly, the most consistent elevations were found in the New York sequence in which, having been recorded from directly above, it was assumed that these parameters would be the most liable to shift unpredictably.

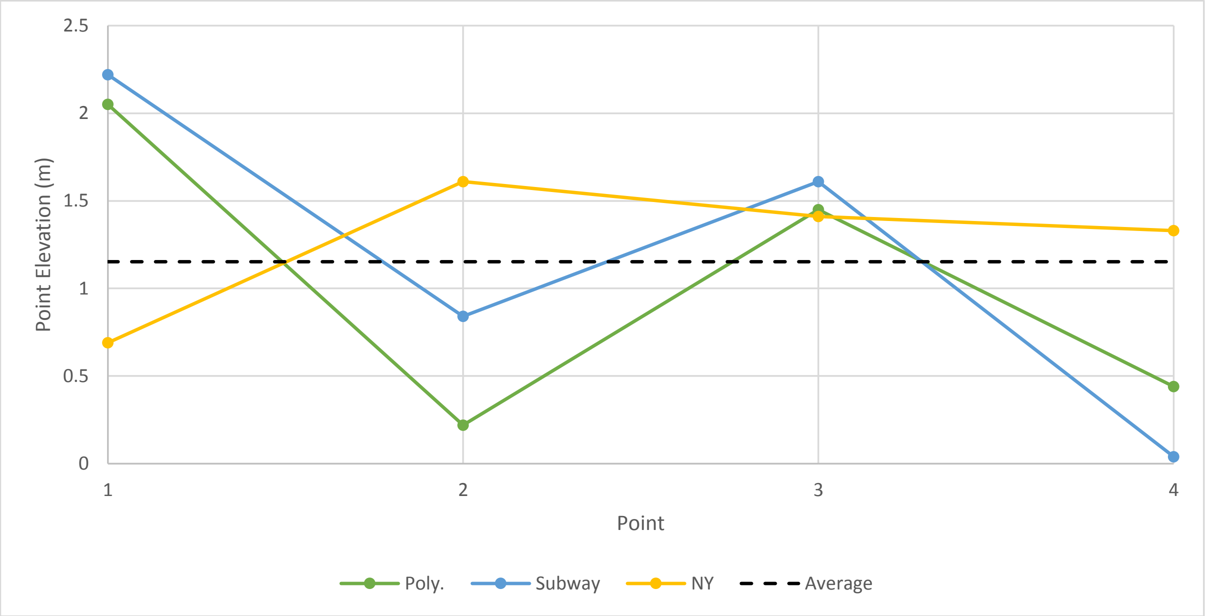
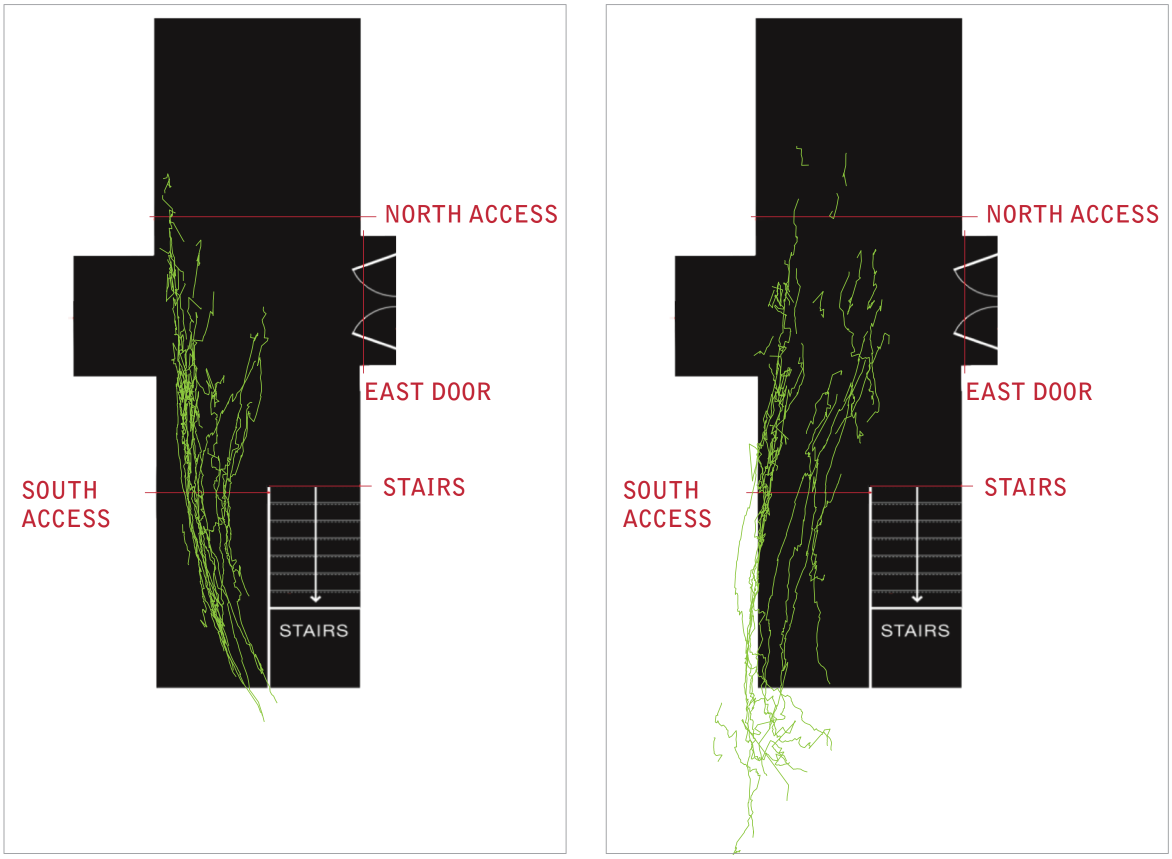


Figure 4.8 Optimized point-correspondence elevations. Point numbering in all cases begins in the right-most foreground, and proceeds counter-clockwise.

Also interesting is that the Polytechnique and subway station videos, which share similar camera angles, also share similar patterns in point-correspondence elevations, with points 1 and 3 (in both cases, those nearest and farthest from the camera, respectively) being nearly a meter higher the other two corner points. In both cases, the axis defined by the two highest points is equally that of the principal movements observed in the sequences, as well as the longest diagonal in the polygons formed by the four points.

The effect of this behavior, intuitively, is the stretching in the principal movement axis of the coordinate system to which pedestrians are projected. This may improve the ability of the trackers to distinguish between individuals walking one behind the other, particularly during the feature-grouping stage of TI - though inversely potentially making distinction of those walking side-by-side more difficult. Furthermore, the advantage in tracking accuracy that results may come at the expense of the meaningfulness of the track positions produced.



With homography parameters

Standard homography

Figure 4.9 Trajectories produced by optimized Traffic Intelligence in the Polytechnique corridor calibration sequence, with and without optimized homography elevation. Lens distortion was not corrected for in this trial, leading to the exaggerated curvature of the tracks on the lower portion of the corridor.

In order to better isolate the effect of homography parameters, TI was run with optimized tracker parameters and all homography elevations set to zero. TI was selected for this test, as UT's tracking occurs entirely in the image-space and is only afterwards projected to the world-space, minimizing the potential impact. The tracks produced by runs with and without homography parameters are presented in figure 4.9.

The most evident impact of modified homography elevation is that it produces tracks within the observed area ; in contrast, using standard ground-level homography, several trajectories are outside the corridor, effectively within the walls. Moreover, tracks are more consistent in the former case, demonstrating less visible noise or partial tracks. This is reflected by the accuracy scores of the trials : while elevated point-correspondence achieves the previously presented MOTA of 0.69, standard homography leads to a MOTA of only 0.39. Of course, parameters had been optimized specifically for the modified homography, most likely exaggerating the performance impact.

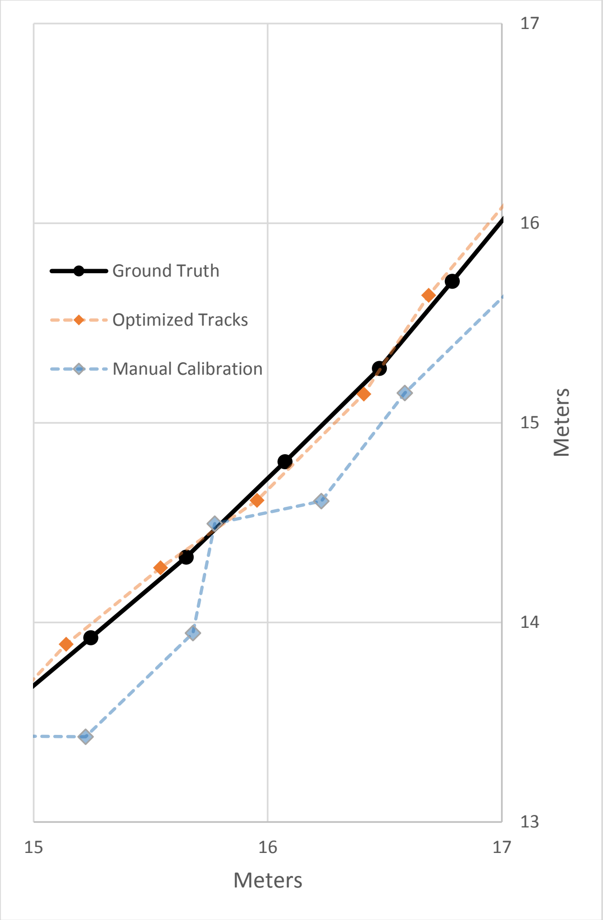


Figure 4.10 Comparison of ground-truth and example tracks plotted every 10 frames for a single pedestrian in the New York sequence, using default (labelled manually calibrated) and optimized homography elevations.

The effect of TrOPed's homography parameters on precision were in line with expectations. Figure 4.10 represents tracks of a single pedestrian with and without optimized homography. This example is particularly egregious and therefore not perfectly representative of common cases, but demonstrates how the additional positional error of unoptimized homography is in large part a result of systematic errors, whereas after optimization positional error is randomly distributed around the ground-truth tracks.

Given that the errors are systematic, one might expect that a simpler solution than optimizing homography would be to simply estimate the size of size and direction of positional errors and correct the tracks accordingly. This would have the added benefit of reducing the number of parameters to be optimized, thereby accelerating the algorithm's convergence.

Unfortunately, positional errors are not uniform in the image-space, but a function of the detected object's position relative to a central point directly vertical to the camera. Specifically, the size of the effect depends on the distance of the object from said point, and its direction on the angle between the camera-center and object-center axes. This fact helps to explain why globally the systematic error was on the order of five centimeters, whereas it was observed to attain upwards of thirty centimeters when evaluated within single square meters.

The fifth homography parameter was intended to compensate for the expected difference between the elevations of tracker (specifically, TI) and ground-truth detections. For both the Polytechnique and subway station sequences, this parameter, once optimized, was similar to expectations : the centers of bounding boxes used for ground-truth annotation were best projected to a homography plane 12 (corridor sequence) and 16 (subway) centimeters below that of the tracker. For the New York sequence, the ground-truth was actually placed 3 centimeters above the tracker plane, functionally difficult to distinguish from zero given the size of both the tracked area and of parameter changes between iterations. It is possible that the vertical camera angle used for this last video, and the resultant reduction in difference in the positions of ground-truth and tracker objects, greatly reduced the utility of its optimization - a possibility made more likely given that the NY video produced the greatest precision of the three test cases.

One final test was performed on the Polytechnique corridor sequence, optimizing TI parameters while maintaining the homography at ground level (this trial was initialized at pre-optimized parameters so as to reduce computation time). The results were a MOTA of 0.57 and MOTP of 0.44, both weaker than performance with modified homography of 0.69 and 0.39 respectively. It is apparent that ground-level homography is problematic for pedestrian tracking, at least when recording is done at short range.

4.3.2 Traffic Intelligence

An example of TI's tracking is shown in figure 4.11. Though error-prone, the produced tracks were majoritarily realistic and corresponded overall to the observable pedestrian movement.

A detailed overview of the types of errors committed by optimized TI is presented in table 4.4. The relative prominence of each error type was consistent throughout the three sequences, being composed majoritarily of missed pedestrians, with mismatches and false positives representing only a minority of committed errors.

Table 4.4 Detailed performance of Traffic Intelligence on the three test cases. Percentages represent the ratio of the related error to the number of ground-truth tracks, which is equivalent to the absolute MOTA reduction caused.

Sequence	Corridor	Subway	Crosswalk
MOTA	0.69	0.39	0.68
MOTP	0.39	0.34	0.30
Ground-Truth Positions	9550	7876	24389
Misses	2241	4588	5779
	23%	52%	24%
Missmatches	75	52	389
	1%	1%	2%
False Positives	128	217	621
	6%	8%	6%

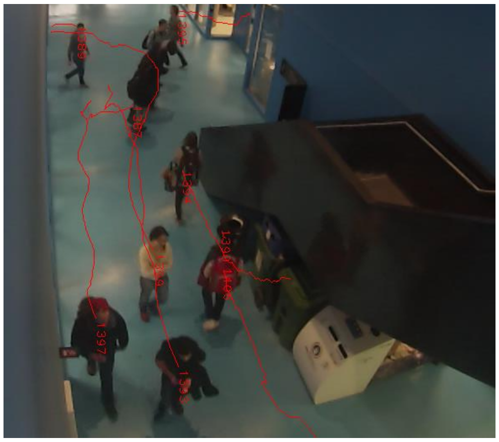


Figure 4.11 Example frame of Traffic Intelligence tracking on the Polytechnique calibration sequence.

The low number of false positives is promising, indicating both a low level of noise and sufficient precision to associate tracks with their related pedestrians. Visualizing the sequence with overlaid tracks confirms that false positives are both relatively rare and most often constrained to very short tracks, easily distinguished (both visually and numerically) from correct trajectories.

The above submetrics underline the primary problem observed in the optimized TI tracks, namely overgrouping of features leading to closely grouped pedestrians being tracked as single objects. Overgrouping also occasionally occurred on single pedestrians, detected twice and therefore producing pairs of parallel trajectories ; together, these errors far outnumbered complete misses, where pedestrians were not detected at all. These behavior translates into both misses and an increase in mismatches as an overgrouped trajectory is successively matched to alternating pedestrians in a group.

One error entirely unique to feature-based tracking was occasionally observed : as such methods account only for detectable corners with no regard for what lies between them, clearly distinct targets (i.e. targets in no way occluding one another) could be grouped into a single object if the underlying features were sufficiently close together. An example of this can be seen in figure 4.12. This may partially explain TI's weaker performance in comparison to UT, as it renders the tracker far more liable to commit overgrouping errors than background-subtraction methods.

Such errors are clearly largely punitive in terms of model calibration. However, higher-level flow characterization remains possible, and is presented in the following chapter.

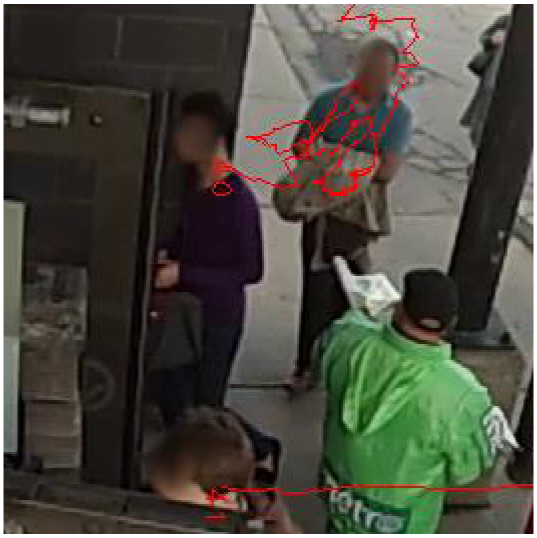


Figure 4.12 Example of Traffic Intelligence failing to dissociate visually distinct targets.

4.3.3 Urban Tracker

A relatively representative example of Urban Tracker's detection is presented in figure 4.13. As befitting the greater MOTA and MOTP scores produced by this tracker relative to Traffic Intelligence, detected pedestrians were generally accurately represented by their associated bounding boxes - in fact, in many cases they compared favorably to the ground-truth itself, the latter having been defined only every 5 to 20 frames and linearly interpolated in between. It is therefore very likely that a portion of the positional error of this tracker (and therefore, by extension, a smaller portion of those in TI) were caused by errors in the ground-truth rather than in the tracker itself.

Detailed error counts are presented in table 4.5. On the subway station sequence, the relative proportion of the different error types is comparable to that observed using TI, and appears to be attributable to a similar range of observable errors.

The Polytechnique corridor sequence, however, is more interesting. The numbers of misses and of false positives are nearly identical, behavior one might intuitively expect from a tracker approaching the limits of its ability, when detection sensitivity is finely balanced with detection accuracy. Indeed, MOTA and MOTP for the calibration sequence were similar to those published in Jodoin *et al.* (2014) though MOTA decreased by more than half on the test video.

Table 4.5 Detailed performance of Urban Tracker on the two functioning test cases. Percentages represent the ratio of the related error to the number of ground-truth tracks, which is equivalent to the absolute MOTA reduction caused.

Sequence	Corridor	Subway
MOTA	0.89	0.54
MOTP	0.46	0.26
Ground-Truth Tracks	9550	7876
Misses	521 5%	3169 40%
Mismatches	38 0%	69 1%
False Positives	507 5%	357 5%

Visualization of the produced tracks reveals error types majoritarily on par with those seen in TI, with one notable exception. As is visible near the background of figure 4.13, on some occasions pedestrians who were adequately tracked individually in addition grouped as a single, larger object, leading to an interesting case of overdetection. Indeed, such errors were common in the subway station sequence (where individuals were more likely to be grouped closely together) with co-moving pedestrians being identified as a single large blob, regardless of the trackers' apparent ability to distinguish all or some of the individuals within.

Though UT produced performance was in most cases far greater than that of TI, it was also more sensitive to overfitting to specific scenes, demonstrating far larger reductions (both in absolute and proportional terms) in tracking accuracy when applied to the test sequences. This effect was even more pronounced with manually calibrated parameters, inversely to TI where visual validation for a given sequence tended to produce similar accuracies during testing. No specific cause could be found for this effect, either by visualization of the tracks or analysis of the error rates, though it was found that the ratio of misses to false positives for the Polytechnique test sequence regressed to that observed in the other examined cases for TI.

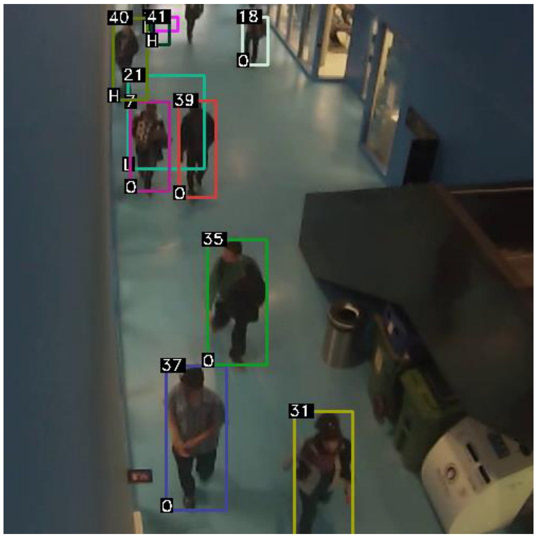


Figure 4.13 Representative frame of Urban Tracker on the Polytechnique calibration sequence.

Of course, the most notable result of the tests of UT were its inability to generate tracks on either of the New York sequences, regardless of input parameters. The apparent cause of this was, in all attempts, a problem in associating specific, individual blobs to either objects or noise. This error recurred at the same frame (and ostensibly the same blob) with subsequent runs of the tracker using the same video file and parameters, though said frame varied when tracker parameters were modified. It is therefore possible that a certain set of parameters would allow tracking on the New York sequence. Unfortunately, in the absence of any generated tracks, MOTA and MOTP metrics cannot be computed, and TrOPed would thus be relegated to functioning as a poor approximation of a brute-force approach. The possibility exists that with small modifications - namely, optimization by frame- or time-of-failure if no tracking metrics are available - TrOPed could be modified to function even in cases of tracker failure, but such an approach was not attempted given the availability of functional data on the other sequences as well as the substantial computation time of the tracker.

4.4 Sensitivity to Test Cases

While the most notable differences between the various tests of TrOPed-optimization occurred between the two trackers themselves, the individual scenes each contained unique elements affecting performance across both UT and TI. These elements are presented below.

4.4.1 Polytechnique Corridor

The corridor sequences produced relatively adequate tracks with both trackers. Though pedestrians climbing or descending the staircase were expected to produce erroneous trajectories given their movement outside the homography plane, the presence of the handrail appears to have sufficiently obscured them so that they were most commonly not detected when more than a handful of steps away from ground level.

This scene also displayed the most complex movement, with several individuals stopping temporarily to converse as well as a number of pedestrians reversing direction entirely. Such actions were generally well handled by both trackers ; in fact, such individuals were almost universally moving alone, and were therefore observed to be tracked more accurately than their more closely packed but simpler-moving compatriots. Of course, as both trackers rely on movement for detection, unmoving individuals effectively become invisible and were therefore lost only to be detected anew once they continued along their paths, but such errors are impossible to avoid given the tracking methodologies.

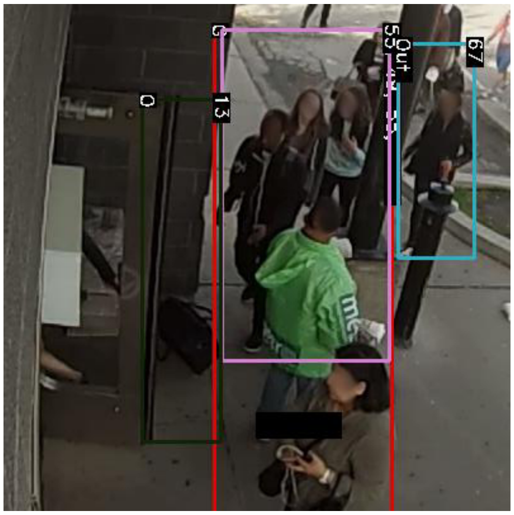
Interestingly, the reversal of camera angles between the calibration and test sequences appear to have had no effect whatsoever ; the performance decrease during testing was in fact the lowest of the three test cases, for both trackers as well as for both manually calibrated and TrOPed optimized parameters. While this represents only a single trial, it is promising in that tracker optimization may only be required for a single viewpoint within an installation if camera angles and scene complexity are sufficiently similar.

However, this scene did present one problem common to both trackers, and certainly true of any scene of sufficient size : tracker accuracy was negatively correlated with the distance of the detected pedestrian from the camera, due to the reduced size in pixels of more distant targets. In the case of TI, this leads to sufficient visible corners for the tracker to detect and thereby group. For UT, blob sizes simply become too small to be distinguished from noise, particularly as both manual and TrOPed-assisted calibration are most likely to be performed for the more numerous and easier proximal targets. Moreover, unlike the inability of these trackers to detect unmoving targets, this problem is certain to extend to any tracker (including tracking performed by humans) as eventually targets are simply too small to present any meaningful detail whatsoever. The only solution to this would be to restrict tracking to areas falling within a certain maximum range, dependent on both the tracker's capability and the recording resolution.

4.4.2 Subway Station

The subway station scene was found to be the most difficult for both trackers, which was initially thought to be due to it also demonstrating the highest pedestrian density. While this factor certainly contributed to a substantial number of misses due to overgrouping (see figure 4.14) two other factors were found to be particularly difficult for either tracker to handle.

First, the presence of a man distributing newspapers to passers-by was highly confusing to both trackers. He was largely immobile, but the movement of the papers themselves was often tracked. In addition, when he did move (as he did regularly to pick up additional newspapers) he produced short, odd tracks. While this may not have been particularly problematic during more generalized use of the data, given that such short and therefore meaningless tracks could easily be discounted from any typical analysis, this presented a significant problem in establishing the ground-truth. It was decided that this individual should not be manually accounted for at all, yet this led to him being, to the optimization algorithm, a significant source of noise. This was particularly problematic given the small space examined, as he occupied a substantial portion of the image-space.



[redacted]

Figure 4.14 Typical tracker failure in the subway station scene : pedestrians approaching from opposite the camera were often highly grouped together both in reality and by the trackers.

The second problem was the presence of the station's doors themselves, one of which was within less than two meters of the camera when opened. Much like the newspaper distributor, this movement was not accounted for in the ground-truth (though in this case it would have been unambiguously egregious to do so) but was too large to be ignored by any combination of tracker parameters, again leading to a large amount of noise. Both trackers seem to have responded to this problem by reducing sensitivity ; it was apparently more beneficial to MOTA to ignore the regular movement of the doors entirely than to attempt to better distinguish individual grouped pedestrians.

Together, these sources of error occupied too large a space for their areas of influence to be eliminated from the tracking area. Both these problems could be at least partially avoided by the use of human-detecting trackers, rather than one based primarily on movement. With the tested trackers, it would be wise to install the camera so as to avoid such areas as much as possible.

4.4.3 New York Crosswalk

As noted earlier, the New York crosswalk sequence contained by a large margin the highest number of pedestrians of the three test cases. Despite this, given the near-elimination of occlusions due to the vertical camera angle and higher resolution, accuracy and precision remained comparable to those of the Polytechnique corridor across both the calibration and test videos.

A representative sample of the New York sequence is presented in figure 4.15. Those pedes-

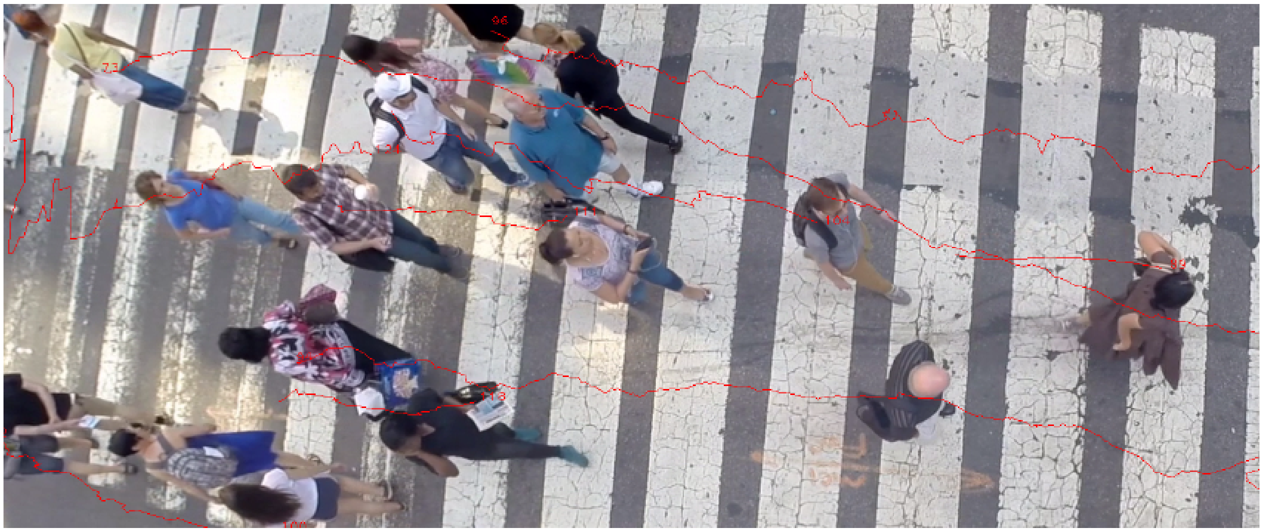


Figure 4.15 Example frame of TI tracking on the New York crosswalk sequence.

trians who were accurately tracked produced long trajectories, representing the near-entirety of their path through the image frame. However, where overgrouping was a primary source of error in the previous scenes, in the crosswalk videos the primary observable error appeared to be failure to detect certain pedestrians entirely - and this, despite them often being two or three meters from others, or their wearing distinct, bright clothing.

A likely cause of this is evident : the fact that movement in these videos consisted not only of pedestrians but also of motor vehicles and cyclists. The largest of these could occupy nearly a quarter of the frame during their passage. Had TI's parameters been more sensitive, therefore, such vehicles would have caused a large number of false positives as small portions were tracked individually. Instead, it would appear that an optimal balance between vehicular and pedestrian accuracy was sought by TrOPed.

As vehicular traffic was constrained almost completely to the video's Y axis, MOTA was recalculated for groups of tracks filtered by their primary direction. Where overall MOTA was 0.68 on the test sequence after optimization, it was only 0.57 for objects moving within 30 degrees of the vertical, and 0.70 for those that were not. Since vehicles would be expected to be far easier to track, these results would imply that the overwhelmingly higher number of pedestrians lead to TrOPed slightly favoring their tracking.

Of course, the stated objective of this work is pedestrian tracking. Consequently, the capability to filter tracks by the desired direction was integrated into TrOPed directly and optimization resumed from the previous solution. After 200 additional iterations, MOTA and MOTP were increased to 0.73 and 0.3 meters, respectively. This filtering removed tracking of targets crossing on the adjacent crosswalk (in the extreme upper right corner of the frame) but given that these pedestrians are at best fully in frame for two seconds, it can be assumed their tracking was not a primary objective of those who gathered the data.

CHAPTER 5 APPLICATION AND DISCUSSION

5.1 Applications of the Optimized Trackers

Once optimization is complete, TrOPed outputs configuration files for the calibrated tracker requiring only user-specification of the video files to be analyzed. Given that the total data represents nearly ten hours of video, and UT not only runs in one sixtieth of real time but failed entirely on the New York crosswalk sequence, the analyses that follow were performed using optimized TI alone.

5.1.1 Tracks

The figures below present the unfiltered tracker outputs using the TrOPed-optimized parameters of the respective scenes. Only single, 30 minute videos were used so as to maintain some clarity in the resultant data.

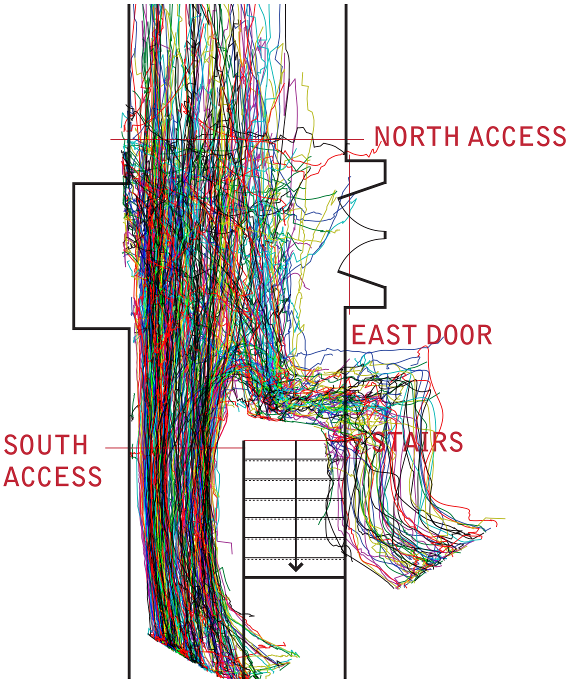


Figure 5.1 Trajectories produced by optimized Traffic Intelligence for a 30 minute sequence in the Polytechnique corridor scene.

These tracks are placed on to-scale maps of the locations in question. Though stylized for presentation herein, the locations of obstacles and the represented areas are identical to the maps used in calculating the homography matrices and therefore to which the tracks were projected by TI.

Figure 5.1 presents the tracks produced for 30 minutes of the Polytechnique corridor videos. Odd behavior can be observed near the staircase, as pedestrians appear to abruptly turn towards (and through) the impassable wall, but this is almost certainly a result of individuals beginning to climb the stairs, thereby exiting the optimized homography plane and being projected farther from the camera. A clear cut-off exists at the west exit of the hallway, a consequence of said area being obstructed to the camera by the wall. Otherwise, the tracks fit the observed space relatively well, though in the absence of directional or density data it is difficult to observe behaviors other than obstacle avoidance.

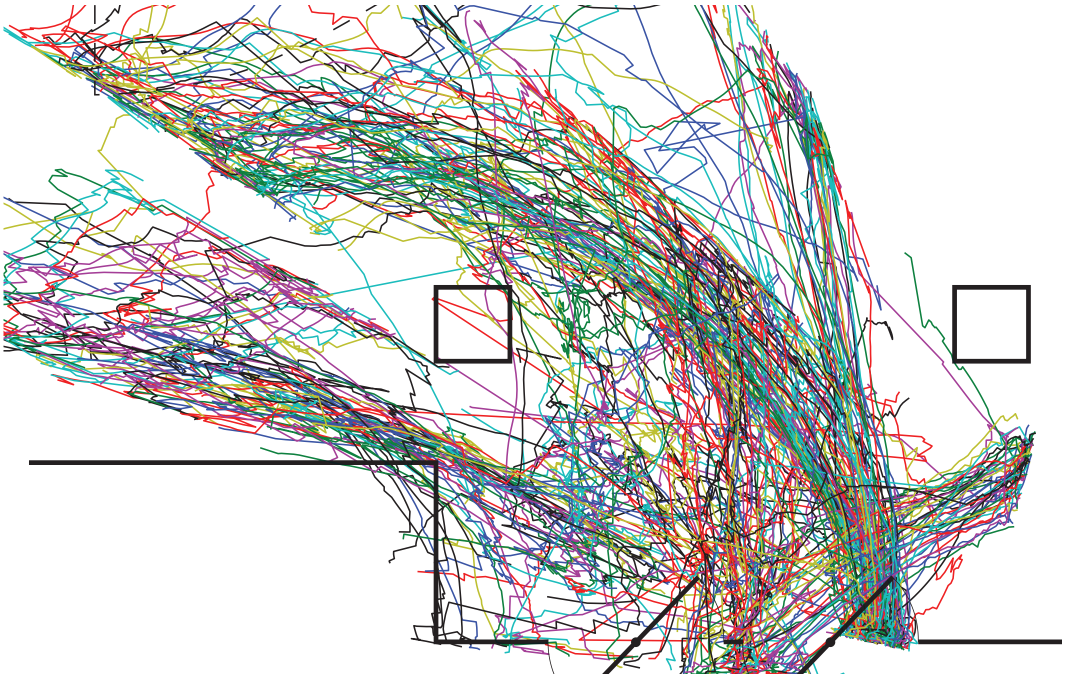


Figure 5.2 Trajectories produced by optimized Traffic Intelligence for a 30 minute sequence in the subway station scene.

Tracks for the subway station entrance are presented in figure 5.2. Again, the produced trajectories fit the studied area well, with clear avoidance of the station's columns and a low density of erratic tracks in the area most commonly occupied by the man handing out newspapers (just right of the leftmost column). This scene, having the poorest CLEAR MOT performance of the three, is also host to the most obviously erroneous traces, with tracks traversing walls and the columns.

Figure 5.3 presents the full set of tracks produced for the New York sequence, unfiltered by

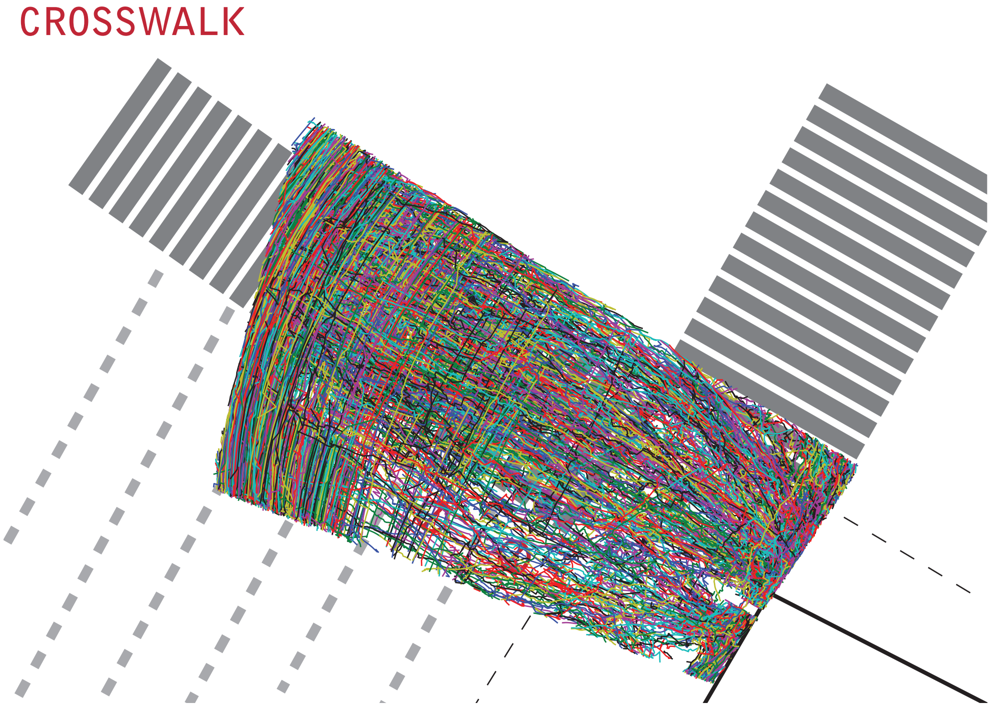


Figure 5.3 Trajectories produced by optimized Traffic Intelligence for a 30 minute sequence in the New York crosswalk scene.

direction. The high traffic volume in this sequence makes behaviors difficult to distinguish ; these tracks are essentially an outline of the area covered by the video frame, particularly given that pedestrians tended to be unconstrained by the crosswalk's markings. Filtered tracks are displayed in figures 5.4 and 5.5.

These tracks appear to be less well aligned than those of the previous scenes. The vehicular tracks in particular seem to imply by their curvature that lens distortion was not entirely corrected for. In addition, the filtering method (performed by evaluating angle relative to the map's X or east-west axis) is not particularly well adapted to the fact that the crosswalk was not built along the cardinal directions. These issues aside, these tracks are again majoritarily consistent with the observed behavior, with clear obstacle avoidance on the sidewalk, smoother vehicular than pedestrian motion, and a general New York disregard for lane discipline.

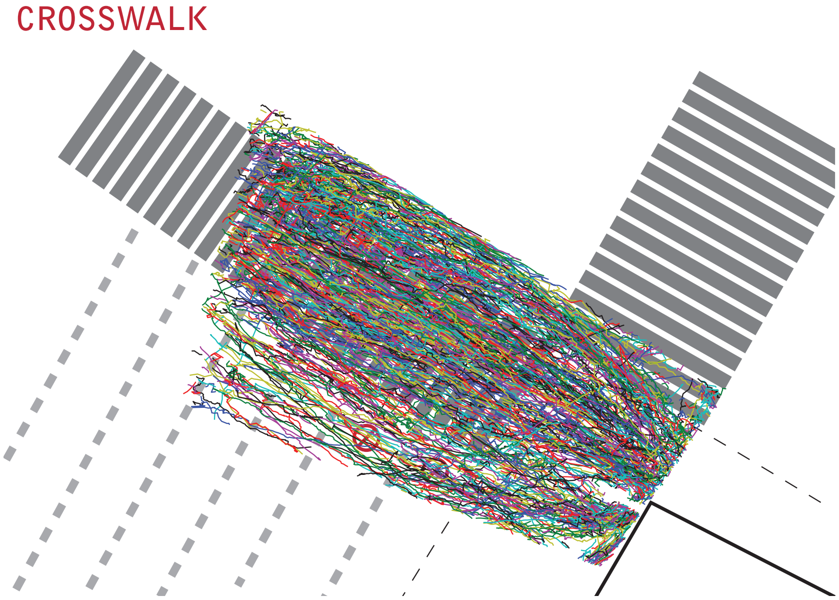


Figure 5.4 New York crosswalk trajectories, filtered to only represent near-horizontal (crossing) tracks.

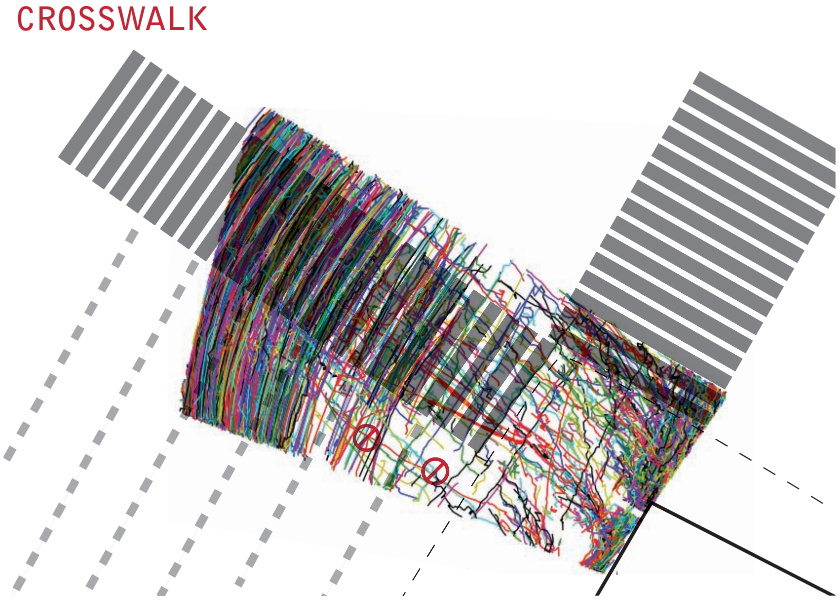


Figure 5.5 New York crosswalk trajectories, filtered to only represent near-vertical (vehicular and sidewalk) tracks.

5.1.2 Density Maps

Detection densities were plotted using Python's matplotlib plotting library, specifically using the hexagonal density function hexbin to discretize the area into a regular array of hexagons. For clarity, only data within the area of interest was plotted.

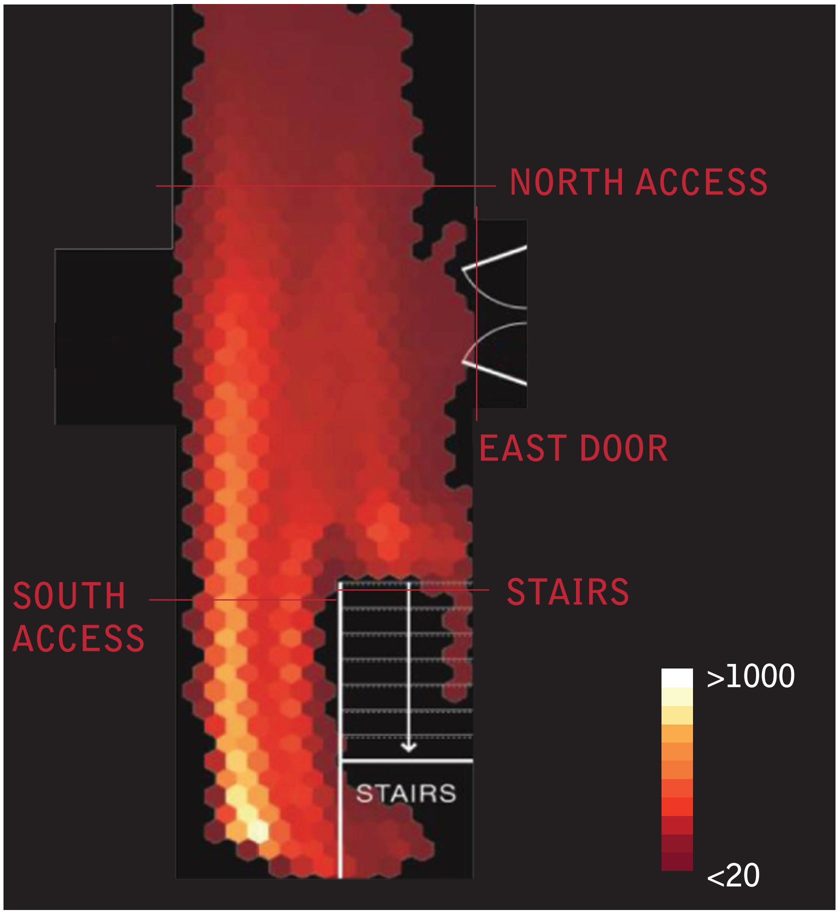


Figure 5.6 Heatmap of pedestrian detection in the Polytechnique corridor sequence. Hexes are approximately 0.4 meters wide. Note that colors represent individual tracker detections, not pedestrians.

Figure 5.6 is the heatmap for the Polytechnique corridor video. The primary observation which can be made from this visualization is clear lane formation, accentuated in the portion of hallway that is narrowed by the presence of the staircase. The larger number of pedestrians accessing the tunnel to the south can be seen taking the right-hand side (from their descending perspective) in both the corridor as a whole and when approaching the stairway ; in the former case, they can also be observed to converge towards one of the two doors giving access to the tunnel.

The tracker's detection rate can also be seen to drop off and distance from the camera (located in the south-western corner) increase, though this effect is compounded by the convergence of pedestrians into lanes causing them to effectively spread out when between the two side accesses.

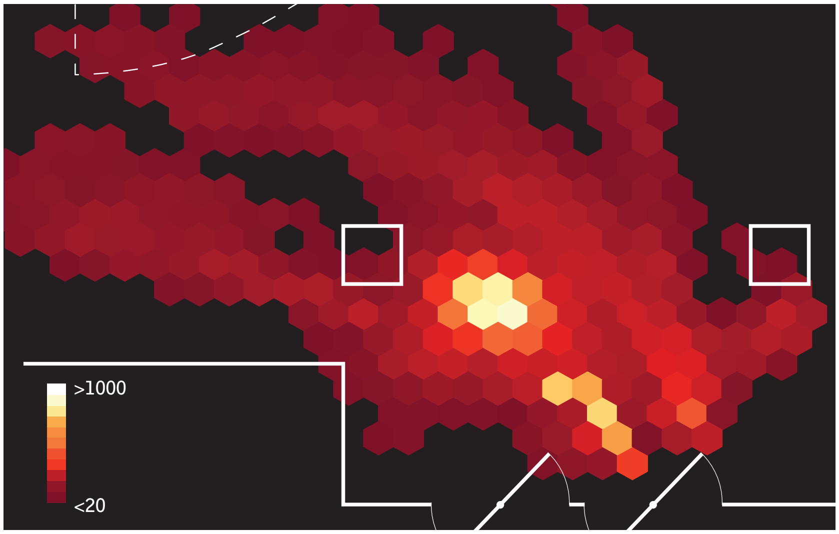


Figure 5.7 Heatmap of pedestrian detection in the subway station sequence. Hexes are approximately 0.4 meters wide.

The same visualization is performed for the subway station video in figure 5.7. Here again, primary movement lanes are visible : one approaching the station to the south of the column, the other exiting through the rightmost door and turning left north of the column. The alternate access on the east side of the picture is also visible, but it is apparent that movement towards the larger, north passage was not well tracked. Moreover, the detection shadows cast by the two columns due to positioning of the camera in the lower-right corner of the image is evident, particularly given that most individuals entering the station from the west side of the figure in reality entered from the north-west corner.

Density maps of NYC tracks filtered by primary direction are presented in figures 5.8 (horizontal) and 5.9 (vertical). The latter figure offers little more information than was visible in figure 5.5, other than highlighting the noise caused by partial tracks in the area between the primary vehicle lanes and the sidewalk. The former, on the other hand, demonstrates that a majority of the tracked pedestrians does indeed stay within the indicated crosswalk, and makes clear how they converge when arriving at the south-eastern corner of the intersection.

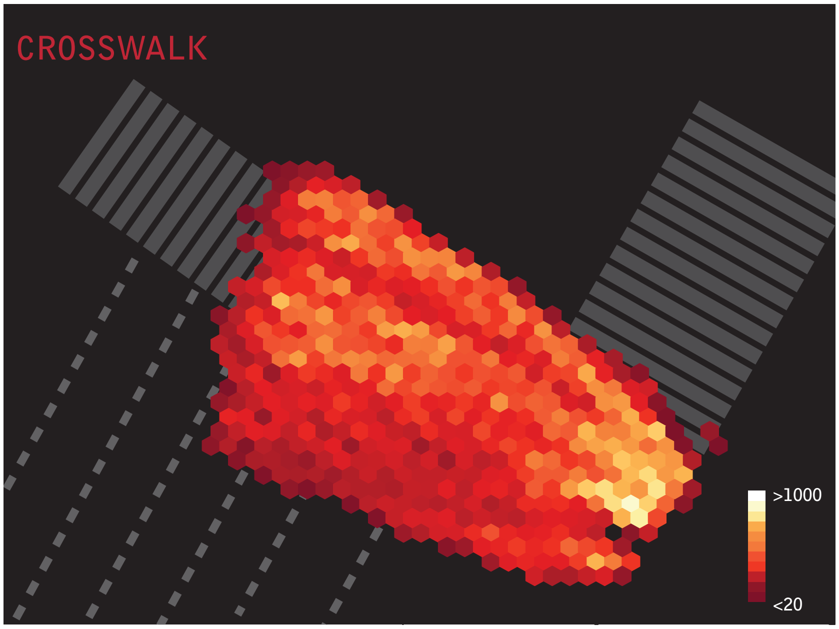


Figure 5.8 Heatmap of pedestrian detection in the New York crosswalk scene, filtered to display only near-horizontal tracks. Hexes are approximately 0.4 meters wide.

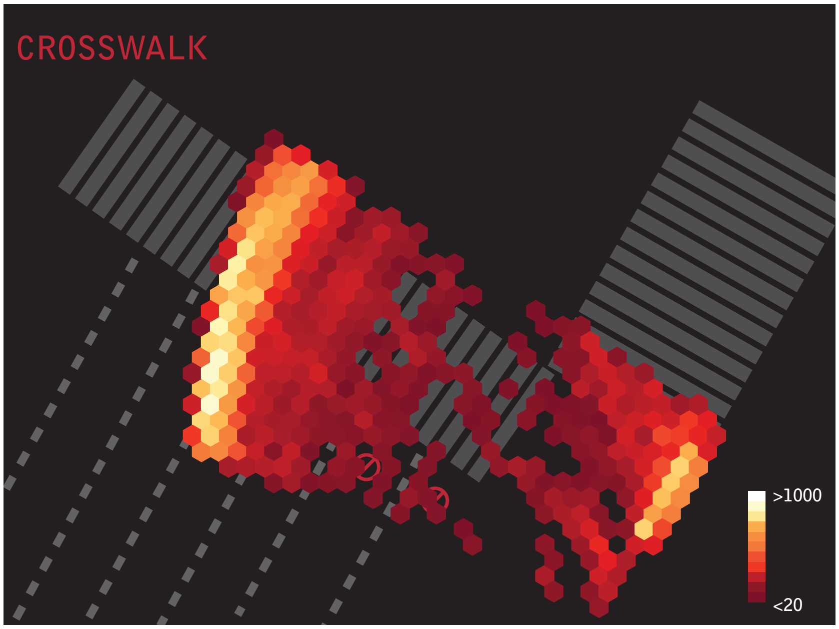


Figure 5.9 Heatmap of pedestrian detection in the New York crosswalk scene, filtered to display only near-vertical tracks. Hexes are approximately 0.4 meters wide.

5.1.3 Speed Profiles

Speeds of detected objects were calculated for all tracks with durations of 100 frames (approximately 3.3 seconds) or more. They were evaluated every second - or 30 frames - and averaged for each object. Figures 5.10, 5.11 and 5.12 present the resultant speed distributions for the Polytechnique corridor, subway station entrance and New York crosswalk, respectively.

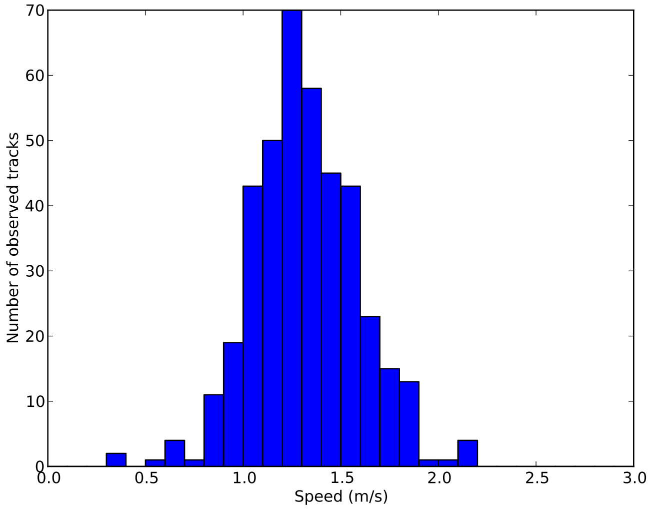


Figure 5.10 Speed distribution for the Polytechnique corridor video.

Pedestrians in the corridor video had a mean measured speed of 1.32 m/s, and a standard deviation of 0.27 m/s. Of the three scenes, it compares best in both shape and average speed to the literature (e.g. Daamen and Hoogendoorn (2003) experimentally measured an average of 1.34 m/s) for unobstructed pedestrian movement. The peak is however distinctly narrower than that obtained by Daamen & Hoogendoorn, possibly reflecting the more homogeneous demographics (a majority of male students) of the scene.

The subway station scene, in contrast, displays a markedly lower average speed of 0.74 m/s, as well as a distribution heavily weighed towards zero. The former observation may be attributable to grouped pedestrians being constrained in the narrow passage between the wall and column, as well as to slowing down at the station's doors themselves. These bottlenecks, and the adjacent open area where flow is uninhibited, help explain the higher standard deviation of 0.32 m/s. The high occurrence of very slow tracks may be a result of detection of both the doors and the man distributing newspapers, whose tracks might have been sufficiently long to bypass the aforementioned filtering of trajectories lasting less than 100 frames.

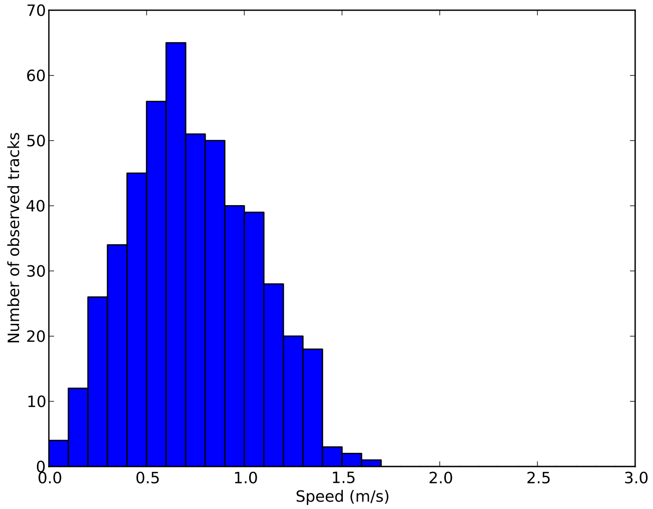


Figure 5.11 Speed distribution for the subway station video.

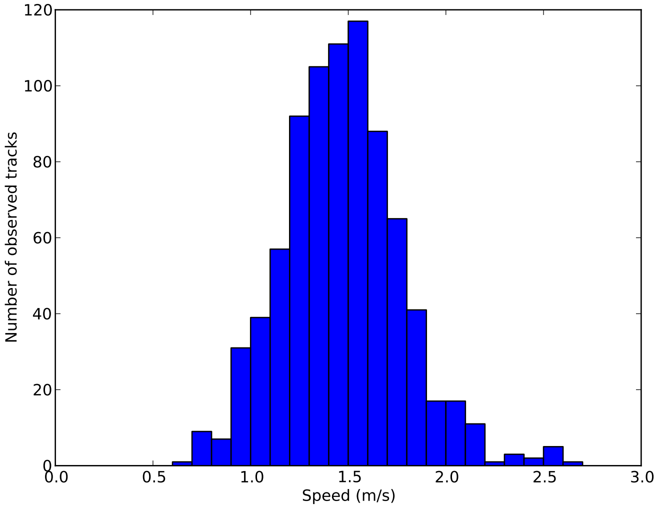


Figure 5.12 Speed distribution for the New York crosswalk video.

An original analysis of the New York videos produced speed distributions with two peaks : one near 1.5 m/s for pedestrians, and another at 3.5 m/s which ostensibly represented vehicles and cyclists. As an alternative to the directional filtering performed above, the data was simply truncated to those tracks with speeds of less than 3 m/s. The result is figure 5.12, with a mean speed of 1.47 m/s and standard deviation of 0.30 m/s. The global distribution is generally similar to that obtained in the Polytechnique scene, with a small number of low-speed observations (likely pedestrians encroaching on the street as they prepare to cross). There is also a small group of tracks of higher speed (near 2.5 m/s) which almost certainly represents cyclists.

That the mean speed is higher in the New York City scene than that observed in the corridor scene despite a more heterogeneous population can be explained by the added urgency in crossing a major avenue. Alternatively, it may simply indicate that pedestrians in New York walk faster than those in Montreal, a hypothesis in line with stereotypes of the "typical New Yorker". Of course, it may simply be the result of poor definition of the dimensions of the crosswalk, though there is little reason to expect this to be the case.

The above analyses contrast with the others performed in this chapter in that they are the least sensitive to the overall performance of the tracker : if an object is tracked for any length of time (as is garanteed by the exclusion of tracks of less than 3.3 seconds) the only tracker error which would translate into significant errors in speed would be repeated mismatches. As this type of error was comparatively rare, one can expect that the resultant speeds are majoritarily accurate.

5.1.4 Directional Counts

Counts were performed by defining screenlines within the world map used for homography calculation ; TrOPed then determines the number of tracks that cross each pair of screenlines. While in theory this allows for local origin-destination analyses in the tracked area, the relatively low tracker accuracy - and consequent large number of partial tracks - of TI made the definition of adequate screenlines difficult. Indeed, counts would vary greatly with only small changes in their location.

More accurate results were obtained by instead tracing parallel thresholds crossing areas of interest. The resulting small-scale origin-destination data then served as linear, directional counts for these areas, which in certain simple cases are functionally identical to the larger scale method attempted earlier. The counts thus produced were compared with manual counts, and the results are presented alongside the utilized thresholds in figures 5.13, 5.14 and 5.15. In an attempt to increase counting accuracy in cases of partial tracks, the latter were extrapolated 2 meters beyond their first and last detections, in a manner inspired by the similar method utilized by Rabaud and Belongie (2006).

Pedestrian counts on the Polytechnique sequence (presented in figure 5.13) were surprisingly accurate : though the average absolute error rate was 34% globally (comparable to the tra-

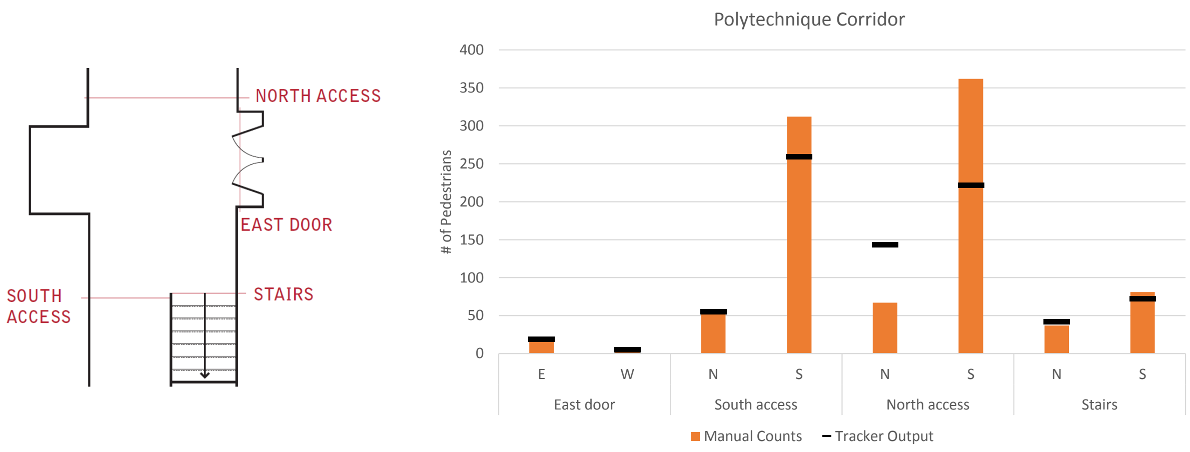


Figure 5.13 Counting thresholds (left) and comparison of automated vs. manual counts (right) for the Polytechnique corridor video.

cking error rate of 32% on the test sequence) this includes the substantially higher errors on the north access, at which tracker accuracy was noted to diminish significantly. If the latter area is ignored, the absolute error decreases to only 20%, or an average overestimation of 11%.

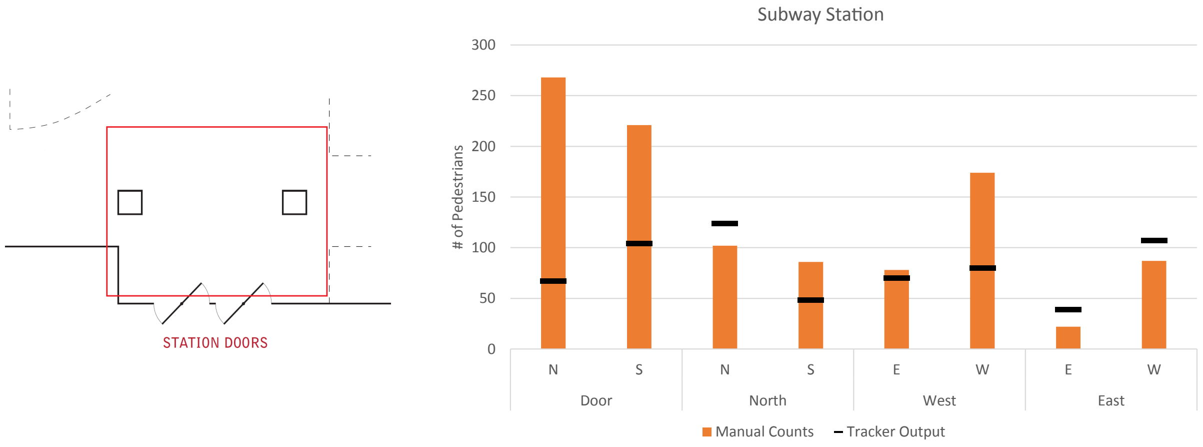


Figure 5.14 Counting thresholds (left) and comparison of automated vs. manual counts (right) for the subway station video.

The two other scenes, however, counting accuracy was substantially lower. The relatively poorly-tracked subway station scene (figure 5.14) had an average absolute error of 45%, with the previously observed underdetection confirmed by a mean error of -14%, particularly pronounced at the door. The New York crosswalk (figure 5.15) in contrast demonstrated overestimation of counts at all three screenlines, averaging a +54% error. While for vehicular counts this can easily be attributed to their large size and the resultant overdetection of individual cars, the pedestrian counts proffer no explanation beyond the presence of noise.

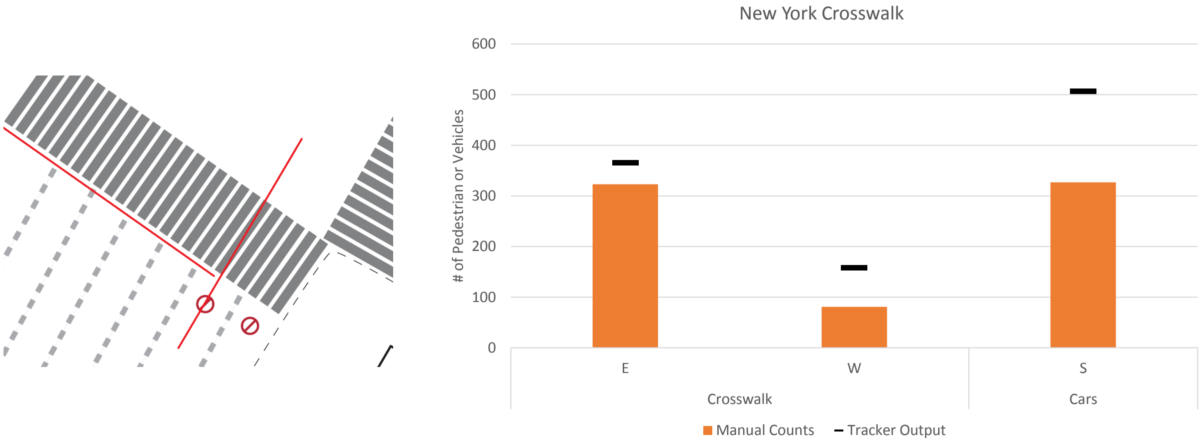


Figure 5.15 Counting thresholds (left) and comparison of automated vs. manual counts (right) for the New York crosswalk video.

5.2 Revised Clear MOT Metrics

As discussed in section 3.6, the performance metrics used thus far were a literal interpretation of Keni and Rainer (2008). The resultant performance measures were directly validated against those obtained in Jodoin *et al.* (2014), and were ostensibly similar to those in the literature as a whole.

However, the matchmaking algorithm underlying the evaluation of MOTA and MOTP demonstrate one significant oversight : although every ground-truth detection can only be associated to a single tracker object, the inverse is not true. In cases where two pedestrians are grouped for any amount of time, therefore, the evaluation function as applied is likely to underestimate the committed error, as both individuals could (if sufficiently close together) be associated to the same tracker track. This behavior would help explain the predominance of overgrouping by both trackers in the majority of examined sequences.

To correct this, the algorithm was expanded to ensure one-to-one matching of both ground-truth and tracker tracks. Such matching is, effectively, a special case of the *stable marriage problem* (Gusfield and Irving, 1989) of finding the best matches for a pair of sets, where the sets are of unequal size and where one - in this case the ground-truth tracks - is predominant over the other. The resulting algorithm is presented below.

By replacing the extant matchmaking algorithm with the revised version above, TrOPed optimization was conducted anew on the test sequences. Again, given the significant performance difference between TI and UT, the former was utilized for these tests. The results are presented in table 5.1, alongside those of the earlier optimizations.

Table 5.1 Comparison of performance and error rates after optimization of TI using both the originally presented and revised matchmaking algorithms.

Sequence	Corridor		Subway		Crosswalk	
Matchmaking	Original	Revised	Original	Revised	Original	Revised
MOTA	0.69	0.32	0.39	0.01	0.68	0.47
MOTP	0.34	0.33	0.34	0.36	0.30	0.26
Ground-Truth Tracks	9550		7876		24389	
Misses	2241 23%	6022 63%	4588 52%	7769 99%	5779 24%	7769 48%
Mismatches	75 1%	164 3%	52 1%	28 2%	389 2%	624 4%
False Positives	128 2%	280 6%	217 4%	0 0%	621 4%	1061 8%

Performance suffered substantially from the more strict matchmaking method. Largely, this is explained by an increase the number of misses, as would be expected from the inability of the evaluation function to match objects to more than a single object. Interestingly, this was accompanied by a parallel increase in the number of false positives, quite possibly representing an attempt by TrOPed to compensate for the now-detectable incidences of overgrouping by reducing the requisite size of objects, thereby increasing the tracker's sensitivity to noise. The primary positive result was the decrease in the ratio of mismatches, implying that those tracks that were not noise were indeed more representative of the tracked pedestrians.

This last observation suggests that such matchmaking is preferable to the less robust one used previously in terms of producing reliable and meaningful microscopic trajectories. Unfortunately - with TI at least - achieving such tracks for all pedestrians in complex scenes appears to be impossible.

```
Matches = empty table of matches;
for Each frame of the video sequence do
    GTpoints, TrackerPoints = ground-truth and tracker points existing in the current frame;
    GTmatches, TrackerMatches = empty lists containing current best matches for every GT and Tracker point in the current frame;
    newMatches = 1 # Allow the following while-loop to begin;
    while better matches have been found do
        newMatches = 0 # Reinitialize counter ;
        for each GTpoint do
            for each Tracker point do
                dist = distance between GT and Tracker points;
                if $dist < maxMatchingDistance$ then
                    if $dist <$ the current best distance for both the GT and Tracker points then
                        register new best match in GTmatches and TrackerMatches;
                        newMatches += 1;
                    end
                end
            end
        end
    end
    add matches for this frame to Matches table ;
end
```

Algorithm 4: Matchmaking algorithm for performance evaluation.

5.3 Discussion

While the analyses performed above provided no surprises in terms of revealing pedestrian behavior that would not be easily obtained by simple observation, such observations are precisely the objective of automated tracking and data collection. Indeed, the very evaluation of the performance of such tools - whether by the Clear MOT metrics or otherwise - is performed through comparison with manually extracted data, with the goal of achieving similar or superior accuracy and precision while substantially reducing their cost.

In terms of the MOTA and MOTP metrics, the obtained tracking performances are at best comparable to the state of the art trackers, and generally significantly worse. However, it is important to recall that the trackers used to test the optimization and homography methods were not conceived for pedestrian tracking in particular, but for vehicular movement and multimodal safety diagnosis. They therefore lack many of the more recent tools used specifically in the detection, tracking and prediction of pedestrian movement, and when applied using only manual calibration (as is done with the more specialized trackers) perform markedly worse than their nominally more appropriate counterparts - yet this difference nearly disappears when TrOPed's optimization is used. In addition, the scenes studied in this work were selected specifically for the complexity, making direct comparison between these and other trackers difficult.

Under the assumption that pedestrian-focused trackers lie somewhere between Traffic Intelligence and Urban Tracker in terms of sensitivity to their input parameters, it is therefore not unreasonable to expect similar improvements than those presented herein. Given that average error reduction after TrOPed optimization was found to consistently be near 50% and that state of the art trackers regularly attain MOTA scores approaching 0.85, application of TrOPed to the latter trackers may - if similar results are obtained - reduce the automated pedestrian tracking error to only 7 or 8%. This is well within the range of human error in manual counts, measured by Diogenes *et al.* (2007) to average 14%. If such accuracies can be achieved, the manual alternative could be relegated solely to giving the tracker sufficient ground-truth data to calibrate itself to the scenes to be studied.

Furthermore, as a consequence of the efforts made to ensure TrOPed is can be applied to a wide range of potential video-based pedestrian trackers, it is in - in theory - sufficiently adaptable to apply to *any* parameterizable tracking tool, or indeed any software which seeks to reproduce spatial trajectories at all. Nor is it limited to two-dimensional space : MOTA and MOTP are both readily expandable to the third (or in fact any number of) dimensions, allowing its use with, for example, trackers using stereo cameras, though TrOPed's homography adjustments would not be of use.

Of course, the above assumptions on the optimization algorithm's utility are subject to the generalizability of the test cases. While these were selected specifically for their tracking difficulty (as well as, admittedly, their utility in parallel research projects) they are neither among the standard pedestrian tracking cases nor necessarily representative of the wide range of potential applications of automated tracking. A similar question arrises in terms of TI and UT, which have not been explicitely compared to other trackers in terms of their potential for optimization.

This may be particularly problematic with regards to the search-spaces which TrOPed would confront when applied to other trackers. Those of both TI and UT demonstrated a majority of low-performance parameter sets, punctuated by one primary and relatively easily detected island of better accuracy. Other trackers - particularly ones that are sufficiently parsimonious to produce adequate MOTA and MOTP scores regardless of their parameters, or whose detection and tracking methods differ significantly from those studied here - may present altogether different search spaces for which TrOPed's simulated annealing algorithm is less appropriate and thus less likely to find the sought-after global maximum.

CHAPTER 6 CONCLUSION

6.1 Summary

The research presented herein aimed to offset the predominance of manual calibration of video-based trackers in the literature, particularly in the difficult case of pedestrian tracking, through the development of a generalizable framework for automated tracker optimization. In essence a simplified learning algorithm, said framework - Tracker Optimizer for Pedestrians, or TrOPed - would utilize a small segment of manually-tracked data in order to calibrate tracker parameters to a given scene so as to maximize performance as evaluated by the CLEAR MOT accuracy and precision metrics.

The primary means of accomplishing this task was the selection and implementation of the simulated annealing metaheuristic so as to solve the underlying global optimization problem. Calibrated alongside the state-generation function for consistent and efficient (albeit inherently stochastic) convergence to the subjected tracker's maximum performance on a test sequence, the framework then outputs the optimal parameters for use on an amount of data for the same or similar scenes.

Tested on two trackers of inherently differing methodologies (Traffic Intelligence's feature-based tracking and Urban Tracker's background subtraction) applied to three cases selected for their tracking difficulty, TrOPed consistently reduced tracking error by between 35 and 65% regardless of the initial parameters and without requiring adjustment of the optimization algorithm for specific tracker-scene combinations.

The above task was facilitated by the addition of two secondary functionalities. The first of these was the addition of parameters regulating the elevation of the homography plane, used in the projection of trajectories from the video-frame to real-world coordinates. Stemming from the observed difficulties in precisely locating detected objects, particularly in video recordings from confined spaces and of vertically elongated pedestrians, these homography elevation parameters are optimized parallel to the tracker parameters. Comparisons with typical ground-level homography revealed not only greater precision in the resultant tracks, but also superior overall accuracy when used in conjunction with trackers making direct use of real-world coordinates.

The second added functionality was a set of integrated tools allowing both the analysis and visualization of trajectories output by the tracker. By aggregating either all or a filtered subset of the output tracks, these tools facilitated holistic validation of the automatically collected data against visual observation, confirming that algorithmically calibrated parameters were capable of adequately reproducing real microscopic behaviors even in cases of low CLEAR MOT measured performance.

6.2 Limitations and Future Work

One aspect of video-based tracking which has not been implemented in TrOPed is the automated classification of detected objects, a highly useful feature in multimodal settings and one which is currently partially implemented in Traffic Intelligence (Zangenehpour *et al.*, 2014). Although in the presented cases this functionality could be heuristically achieved by the directional or speed-based filtering of tracks, larger or more complex cases benefit greatly from the ability to distinguish vehicles, cyclists and pedestrians, and to optimize tracking for one without simply neglecting the others.

In addition, the overfitting of trackers to calibration sequences and the resultant decrease in tracker performance when applied to other sequences from the same scene has thus far only been superficially studied. Of course, attempts were made to select calibration sequences which best represented the scenes as a whole, and some amount of overfitting is inevitable with the proposed method. However, the extent of this effect is likely highly dependent on the selected calibration scene, and what constitutes a representative sequence was selected relatively arbitrarily. It would therefore be interesting to test overfitting on a broader array of calibration sequences for each scene so that optimal selection criteria can be better and more systematically characterized.

Though the studied trackers both converged relatively rapidly, the apparent dependence on parameter number and range may be problematic when TrOPed is applied to hybridized trackers. As these utilize multiple tracking and/or detection methodologies, such trackers are liable to possess not only the full parameters of each method but also higher level parameters regulating their interaction. The required number of iterations for optimization for such trackers is therefore likely to be significantly higher than those observed here - to say nothing of the computation time for each run of these more complex tools.

Of course, convergence time could be reduced if the distributions and intercorrelations of tracker parameters are better understood and integrated in the state-generation. If TrOPed is applied to a sufficient number of scenes with the same tracker, it would be relatively simple to limit parameter ranges to those known to produce optimal tracks, to initialize optimization from a value near previous optima, or even to exclude the parameter from optimization entirely if it is found always converge to a unique value. While these measures can easily (albeit manually) be taken with the current version of TrOPed, such functionality could be further expanded upon to include more complex relationships between parameters.

In a similar vein, the manner in which TrOPed executes tracker runs at each iteration could be further optimized. In the case of Traffic Intelligence, for example, certain parameters affect only the grouping stage, thereby not requiring that the far longer feature-detection stage be run anew. Further characterization of the optimized parameters could allow TrOPed to avoid the latter stage if it would be unaffected by the parameters modifed by the state-generation function. Moreover, these parameters could be specifically targetted, taking advantage of faster run-times to perform more exhaustive optimization.

Even further improvements could be made through improvement of the central optimization algorithm. While some calibration has been performed manually, a more thorough sensitivity analysis of convergence times to both the initial tracker parameters and to those of TrOPed itself could offer substantial increases in performance. Similar analyses could be performed for the data collection methods, including camera position (both height and angle), orientation, lighting conditions and camera type ; while the impact of these factors can to a certain extent be inferred from the literature, the ability to at least partially eliminate the impact of tracker calibration would allow better isolation of their impact.

Finally, the most important limitation of the presented research is that only two trackers were tested, neither of which were developped with a large emphasis on pedestrian tracking. The achieved improvements are therefore not necessarily representative of those that would be observed on state of the art pedestrian trackers, particularely if the latter are less sensitive to their parameters, or simply possess a smaller number and range of said parameters. TrOPed, while a novel method in increasing tracker performance and demonstratably superior to prior methods in calibrating those trackers to which it has been thus far applied, should be tested on more specialized trackers so as to ascertain its true contribution to pedestrian tracking.

In spite of these limitations, the analyses performed on the extracted tracks highlight the importance of real-world data in the calibration of pedestrian models. Where one could expect similar behavior in the corridor and crosswalk scenes (both of which represent unobstructed, mostly bidirectional flow) they are markedly different both in terms of their speed distributions and the qualitative nature of the flow, namely the absence of clear lane formation in the crosswalk scene). These differences are too marked to be fully attributable to tracker error, and are therefore almost certainly a result of the particularities of the two scenes. Even limited to the tracker performances obtained here, analysis of a larger number of such scenes would allow the factors influencing pedestrian movement to be identified and quantified, in a manner difficult to replicate in an experimental setting.

REFERENCES

ALI, I. and DAILEY, M. N. (2009). Multiple human tracking in high-density crowds. *Advanced Concepts for Intelligent Vision Systems*. Springer, 540–549.

ANDRIYENKO, A. and SCHINDLER, K. (2011). Multi-target tracking by continuous energy minimization. *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 1265–1272.

ANTONINI, G., BIERLAIRE, M. and WEBER, M. (2004). Simulation of pedestrian behaviour using a discrete choice model calibrated on actual motion data. *4th STRC Swiss Transport Research Conference*. vol. 7, 249–258.

ASANO, M., IRYO, T. and KUWAHARA, M. (2010). Microscopic pedestrian simulation model combined with a tactical model for route choice behaviour. *Transportation Research Part C : Emerging Technologies*, 18, 842–855.

BARGH, J. A., CHEN, M. and BURROWS, L. (1996). Automaticity of social behavior : Direct effects of trait construct and stereotype activation on action. *Journal of personality and social psychology*, 71, 230.

BARNICH, O. and VAN DROOGENBROECK, M. (2011). Vibe : A universal background subtraction algorithm for video sequences. *Image Processing, IEEE Transactions on*, 20, 1709–1724.

BERCLAZ, J., FLEURET, F., TURETKEN, E. and FUA, P. (2011). Multiple object tracking using k-shortest paths optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33, 1806–1819.

BEUTIN, T. (2012). Ptv viswalk realistic simulation of pedestrian flows. *Improvins education in the field of urban transport*.

BIRCHFIELD, S. (2007). Klt : An implementation of the kanade-lucas-tomasi feature tracker.

BLUE, V. J. and ADLER, J. L. (2001). Cellular automata microsimulation for modeling bi-directional pedestrian walkways. *Transportation Research Part B : Methodological*, 35, 293–312.

BRADSKI, G. and KAEHLER, A. (2008). *Learning OpenCV : Computer vision with the OpenCV library*. " O'Reilly Media, Inc.".

BU, F., GREENE-ROESEL, R., DIOGENES, M. C. and RAGLAND, D. R. (2007). Estimating pedestrian accident exposure : Automated pedestrian counting devices report. *Safe Transportation Research & Education Center*.

BURSTEDDE, C., KLAUCK, K., SCHADSCHNEIDER, A. and ZITTARTZ, J. (2001). Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A : Statistical Mechanics and its Applications*, 295, 507–525.

CHATTARAJ, U., SEYFRIED, A. and CHAKROBORTY, P. (2009). Comparison of pedestrian fundamental diagram across cultures. *Advances in complex systems*, 12, 393–405.

CITY OF MONTREAL (2002). Montreal master plan. `http://ville.montreal.qc.ca/portal/page?_pageid=2762,3101662&_dad=portal&_schema=PORTAL`.

DAAMEN, W. (2002). Simped : a pedestrian simulation tool for large pedestrian areas. *Conference Proceedings EuroSIW*. 24–26.

DAAMEN, W. and HOOGENDOORN, S. P. (2003). Experimental research of pedestrian walking behavior. *Transportation Research Record : Journal of the Transportation Research Board*, 1828, 20–30.

DALAL, N. (2009). Finding people in images and videos, figure 4.11. `https://tel.archives-ouvertes.fr/file/index/docid/390303/filename/NavneetDalalThesis.pdf`.

DALAL, N. and TRIGGS, B. (2005). Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, vol. 1, 886–893.

DANALET, A., FAROOQ, B. and BIERLAIRE, M. (2014). A bayesian approach to detect pedestrian destination-sequences from wifi signatures. *Transportation Research Part C : Emerging Technologies*, 44, 146–170.

DENZLER, J. and NIEMANN, H. (1997). Real-time pedestrian tracking in natural scenes. *Computer analysis of images and patterns*. Springer, 42–49.

DIOGENES, M. C., GREENE-ROESEL, R., ARNOLD, L. S. and RAGLAND, D. R. (2007). Pedestrian counting methods at intersections : a comparative study. *Transportation Research Record : Journal of the Transportation Research Board*, 2002, 26–30.

ELLIS, A., SHAHROKNI, A. and FERRYMAN, J. M. (2009). Pets2009 and winter-pets 2009 results : A combined evaluation. *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*. IEEE, 1–8.

ETTEHADIEH, D., FAROOQ, B. and SAUNIER, N. (2014). Systematic parameter optimization and application of automated tracking in pedestrian dominant situations. *Transportation Research Record : Journal of the Transportation Research Board*.

ETTEHADIEH, DARIUSH (2014). Troped : Tracker optimization for pedestrians. `https://github.com/Drushkey/TrOPed`.

FELIZ ALONSO, R., ZALAMA CASANOVA, E. and GÓMEZ GARCíA-BERMEJO, J. (2009). Pedestrian tracking using inertial sensors.

FORSYTH, D. A. and PONCE, J. (2002). *Computer Vision : a Modern Approach*. Prentice Hall Professional Technical Reference.

FOXLIN, E. (2005). Pedestrian tracking with shoe-mounted inertial sensors. *Computer Graphics and Applications, IEEE*, 25, 38–46.

GEYER, C. (2011). Introduction to markov chain monte carlo. *Handbook of Markov Chain Monte Carlo*, 3–48.

GREENE-ROESEL, R., DIOGENES, M. C., RAGLAND, D. R. and LINDAU, L. A. (2008). Effectiveness of a commercially available automated pedestrian counting device in urban environments : Comparison with manual counts. *Safe Transportation Research & Education Center*.

GUAN, Y., CHEN, X., WU, Y. and YANG, D. (2013). An improved particle filter approach for real-time pedestrian tracking in surveillance video. *2013 International Conference on Information Science and Technology Applications (ICISTA-2013)*. Atlantis Press.

GUO, R.-Y., HUANG, H.-J. and WONG, S. (2012). Route choice in pedestrian evacuation under conditions of good and zero visibility : Experimental and simulation results. *Transportation research part B : methodological*, 46, 669–686.

GUSFIELD, D. and IRVING, R. W. (1989). *The Stable Marriage Problem : Structure and Algorithms*, vol. 54. MIT press Cambridge.

HANISCH, A., TOLUJEW, J., RICHTER, K. and SCHULZE, T. (2003). Online simulation of pedestrian flow in public buildings. *Simulation Conference, 2003. Proceedings of the 2003 Winter*. IEEE, vol. 2, 1635–1641.

HELBING, D. (1998). A fluid dynamic model for the movement of pedestrians. *arXiv preprint cond-mat/9805213*.

HELBING, D. and MOLNAR, P. (1995). Social force model for pedestrian dynamics. *Physical review E*, 51, 4282.

HENDERSON, L. (1974). On the fluid mechanics of human crowd motion. *Transportation research*, 8, 509–515.

HIMANN, J. E., CUNNINGHAM, D. A., RECHNITZER, P. A. and PATERSON, D. H. (1988). Age-related changes in speed of walking. *Medicine and science in sports and exercise*, 20, 161–166.

HOOGENDOORN, S., DAAMEN, W. and BOVY, P. (2003a). Extracting microscopic pedestrian characteristics from video data, figure 1. `http://pages.citg.tudelft.nl/`

`fileadmin/Faculteit/CiTG/Over_de_faculteit/Afdelingen/Afdeling_Transport_en_Planning/Traffic_management_and_traffic_flow_theory/Dynamisch_Verkeers_Management/Special_Projects/Pedestrians/Publications/doc/TRB03h.pdf`.

HOOGENDOORN, S. P. and BOVY, P. H. (2004). Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B : Methodological*, 38, 169–190.

HOOGENDOORN, S. P., BOVY, P. H. and DAAMEN, W. (2002). Microscopic pedestrian wayfinding and dynamics modelling. *Pedestrian and evacuation dynamics*, 123, 154.

HOOGENDOORN, S. P., DAAMEN, W. and BOVY, P. H. (2003b). Extracting microscopic pedestrian characteristics from video data. *Transportation Research Board 2003 Annual Meeting, CD-ROM, Paper*. No. 477.

ISOBE, M., ADACHI, T. and NAGATANI, T. (2004). Experiment and simulation of pedestrian counter flow. *Physica A : Statistical Mechanics and its Applications*, 336, 638–650.

JIANG, Z., HUYNH, D. Q., MORAN, W., CHALLA, S. and SPADACCINI, N. (2010). Multiple pedestrian tracking using colour and motion models. *Digital Image Computing : Techniques and Applications (DICTA), 2010 International Conference on*. IEEE, 328–334.

JODOIN, J.-P., BILODEAU, G.-A. and SAUNIER, N. (2014). Urban tracker : Multiple object tracking in urban mixed traffic.

JOHANSSON, A. and HELBING, D. (2008). From crowd dynamics to crowd safety : A video-based analysis, figure 2. `http://arxiv.org/pdf/0810.4590.pdf`.

JOHANSSON, A., HELBING, D., AL-ABIDEEN, H. Z. and AL-BOSTA, S. (2008). From crowd dynamics to crowd safety : a video-based analysis. *Advances in Complex Systems*, 11, 497–527.

JOHANSSON, A., HELBING, D. and SHUKLA, P. (2007a). Specification of a microscopic pedestrian model by evolutionary adjustment to video tracking data, figure 2. `http://arxiv.org/pdf/0810.4587.pdf`.

JOHANSSON, A., HELBING, D. and SHUKLA, P. K. (2007b). Specification of the social force pedestrian model by evolutionary adjustment to video tracking data. *Advances in complex systems*, 10, 271–288.

KASTURI, R., GOLDGOF, D., SOUNDARARAJAN, P., MANOHAR, V., GAROFOLO, J., BOWERS, R., BOONSTRA, M., KORZHOVA, V. and ZHANG, J. (2009). Framework for performance evaluation of face, text, and vehicle detection and tracking in video : Data, metrics, and protocol. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31, 319–336.

KENI, B. and RAINER, S. (2008). Evaluating multiple object tracking performance : the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008.

KERRIDGE, J., ARMITAGE, A., BINNIE, D., LEI, L. and SUMPTER, N. (2004). Monitoring the movement of pedestrians using low-cost infrared detectors : Initial findings. *Proceedings US Transport Research Board Annual Meeting.*

KHANLOO, B. Y. S., STEFANUS, F., RANJBAR, M., LI, Z.-N., SAUNIER, N., SAYED, T. and MORI, G. (2012). A large margin framework for single camera offline tracking with hybrid cues. *Computer Vision and Image Understanding*, 116, 676–689.

KRETZ, T., GRÜNEBOHM, A., KAUFMAN, M., MAZUR, F. and SCHRECKENBERG, M. (2006). Experimental study of pedestrian counterflow in a corridor. *Journal of Statistical Mechanics : Theory and Experiment*, 2006, P10001.

KRETZ, T., HENGST, S., ROCA, V., PEREZ ARIAS, A., FRIEDBERGER, S. and HANEBECK, U. D. (2011). Calibrating dynamic pedestrian route choice with an extended range telepresence system. *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on.* IEEE, 166–172.

KRETZ, T., HENGST, S. and VORTISCH, P. (2008). Pedestrian flow at bottlenecks-validation and calibration of vissim's social force model of pedestrian traffic and its empirical foundations. *arXiv preprint arXiv :0805.1788.*

KRIEGMAN, D. (2007). Homography estimation. *Lecture Computer Vision I, CSE A*, 252.

LAKOBA, T. I., KAUP, D. J. and FINKELSTEIN, N. M. (2005). Modifications of the helbing-molnar-farkas-vicsek social force model for pedestrian evolution. *Simulation*, 81, 339–352.

LARSON, J. S., BRADLOW, E. T. and FADER, P. S. (2005). An exploratory look at supermarket shopping paths. *International Journal of research in Marketing*, 22, 395–414.

LEUTENEGGER, S., CHLI, M. and SIEGWART, R. Y. (2011). Brisk : Binary robust invariant scalable keypoints. *Computer Vision (ICCV), 2011 IEEE International Conference on.* IEEE, 2548–2555.

MAGGIO, E. and CAVALLARO, A. (2011). *Video Tracking : Theory and Practice.* John Wiley & Sons.

MASOUD, O. and PAPANIKOLOPOULOS, N. P. (1997). Robust pedestrian tracking using a model-based approach. *Intelligent Transportation System, 1997. ITSC'97., IEEE Conference on.* IEEE, 338–343.

MEI, X. and LING, H. (2011). Robust visual tracking and vehicle classification via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33, 2259–2272.

MILAN, A., SCHINDLER, K. and ROTH, S. (2013). Challenges of ground truth evaluation of multi-target tracking. *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*. IEEE, 735–742.

NGUYEN, H. T. and BHANU, B. (2012). Real-time pedestrian tracking with bacterial foraging optimization. *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*. IEEE, 37–42.

NOLDUS, L. P., SPINK, A. J. and TEGELENBOSCH, R. A. (2002). Computerised video tracking, movement analysis and behaviour recognition in insects. *Computers and Electronics in Agriculture*, 35, 201–227.

OPENCV, L. (2008). Computer vision with the opencv library. *GaryBradski & Adrian Kaebler-O'Reilly*.

OXLEY, J., LENNÉ, M. and CORBEN, B. (2006). The effect of alcohol impairment on road-crossing behaviour. *Transportation Research Part F : Traffic Psychology and Behaviour*, 9, 258–268.

PERERA, A. A., SRINIVAS, C., HOOGS, A., BROOKSBY, G. and HU, W. (2006). Multi-object tracking through simultaneous long occlusions and split-merge conditions. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, vol. 1, 666–673.

PÉREZ, Ó., PATRICIO, M., GARCíA, J. and MOLINA, J. M. (2006a). Improving the segmentation stage of a pedestrian tracking video-based system by means of evolution strategies. *Applications of Evolutionary Computing*, Springer. 438–449.

PÉREZ, Ó., PATRICIO, M., GARCíA, J. and MOLINA, J. M. (2006b). Improving the segmentation stage of a pedestrian tracking video-based system by means of evolution strategies, figure 6. `http://e-archivo.uc3m.es/bitstream/handle/10016/9321/improving_molina_LNCS_2006.pdf?sequence=3`.

PICCARDI, M. (2004). Background subtraction techniques : a review. *Systems, man and cybernetics, 2004 IEEE international conference on*. IEEE, vol. 4, 3099–3104.

QIU, F. and HU, X. (2013). Spatial activity-based modeling for pedestrian crowd simulation. *Simulation*, 89, 451–465.

RABAUD, V. and BELONGIE, S. (2006). Counting crowded moving objects. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, vol. 1, 705–711.

REMAGNINO, P., BAUMBERG, A., GROVE, T., HOGG, D., TAN, T., WORRALL, A. D., BAKER, K. D. *ET AL.* (1997). An integrated traffic and pedestrian model-based vision system. *BMVC*.

ROBIN, T., ANTONINI, G., BIERLAIRE, M. and CRUZ, J. (2009). Specification, estimation and validation of a pedestrian walking behavior model. *Transportation Research Part B : Methodological*, 43, 36–56.

ROSS, S. M. (1997). Simulation, statistical modeling and decision science. *Harcourt Academic Press. Observed and Fitted Distribution of Number of Claims : Poisson-Inverse Gaussian, Maximum Likelihood Method k()() n N tk=()() nP N tk= 0*, 96978, 5.

SAHALEH, A. S., BIERLAIRE, M., FAROOQ, B., DANALET, A. and HÄNSELER, F. S. (2012). Scenario analysis of pedestrian flow in public spaces. *Proceeding of the 12th Swiss Transport Research Conference (STRC), Monte Verità, Ascona, Switzerland.* Citeseer.

SAUNIER, N. (2011). wiki.polymtl.ca : Equipment. `http://wiki.polymtl.ca/transport/index.php/Equipment`.

SAUNIER, N. and SAYED, T. (2006). A feature-based tracking algorithm for vehicles in intersections. *Computer and Robot Vision, 2006. The 3rd Canadian Conference on.* IEEE, 59–59.

SAUNIER, N., SAYED, T. and ISMAIL, K. (2009). An object assignment algorithm for tracking performance evaluation. *11th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2009), Miami, Fl, USA, June.* vol. 25, 9–17.

SCHADSCHNEIDER, A., KLINGSCH, W., KLÜPFEL, H., KRETZ, T., ROGSCH, C. and SEYFRIED, A. (2009). Evacuation dynamics : Empirical results, modeling and applications. *Encyclopedia of complexity and systems science*, Springer. 3142–3176.

SCHLICHT, J., CAMAIONE, D. N. and OWEN, S. V. (2001). Effect of intense strength training on standing balance, walking speed, and sit-to-stand performance in older adults. *The Journals of Gerontology Series A : Biological Sciences and Medical Sciences*, 56, M281–M286.

SERGEANT, D., BOYLE, R. and FORBES, M. (1998). Computer visual tracking of poultry. *Computers and Electronics in Agriculture*, 21, 1–18.

SEYFRIED, A., STEFFEN, B., KLINGSCH, W. and BOLTES, M. (2005a). The fundamental diagram of pedestrian movement revisited. *Journal of Statistical Mechanics : Theory and Experiment*, 2005, P10002.

SEYFRIED, A., STEFFEN, B., KLINGSCH, W. and BOLTES, M. (2005b). The fundamental diagram of pedestrian movement revisited, figure 3.

SIDLA, O., LYPETSKYY, Y., BRANDLE, N. and SEER, S. (2006). Pedestrian detection and tracking for counting applications in crowded situations. *Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on.* IEEE, 70–70.

SINGH, V. K., WU, B. and NEVATIA, R. (2008). Pedestrian tracking by associating tracklets using detection residuals. *Motion and video Computing, 2008. WMVC 2008. IEEE Workshop on.* IEEE, 1–8.

TEKNOMO, K. and GERILLA, G. P. (2008). Mesoscopic multi-agent pedestrian simulation. *Transportation Research Trends*, 323–336.

TOMASI, C. and KANADE, T. (1991). *Detection and Tracking of Point Features.* School of Computer Science, Carnegie Mellon Univ. Pittsburgh.

TRUCCO, E. and PLAKAS, K. (2006). Video tracking : a concise survey. *Oceanic Engineering, IEEE Journal of*, 31, 520–529.

WEIFENG, F., LIZHONG, Y. and WEICHENG, F. (2003). Simulation of bi-direction pedestrian movement using a cellular automata model. *Physica A : Statistical Mechanics and its Applications*, 321, 633–640.

WU, B. and NEVATIA, R. (2005). Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on.* IEEE, vol. 1, 90–97.

YAMAMOTO, K., KOKUBO, S. and NISHINARI, K. (2007). Simulation for pedestrian dynamics by real-coded cellular automata (rca). *Physica A : Statistical Mechanics and its Applications*, 379, 654–660.

ZANGENEHPOUR, S., MIRANDA-MORENO, L. and SAUNIER, N. (2014). Automated classification based on video data at intersections with heavy pedestrian and bicycle traffic : Methodology and application. *Canadian Multidisciplinary Road Safety Conference.*

ZANLUNGO, F., IKEDA, T. and KANDA, T. (2011). Social force model with explicit collision prediction. *EPL (Europhysics Letters)*, 93, 68005.

ZHANG, J., KLINGSCH, W., SCHADSCHNEIDER, A. and SEYFRIED, A. (2011). Transitions in pedestrian fundamental diagrams of straight corridors and t-junctions. *Journal of Statistical Mechanics : Theory and Experiment*, 2011, P06004.

ZHANG, X., HU, W., QU, W. and MAYBANK, S. (2010). Multiple object tracking via species-based particle swarm optimization. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20, 1590–1602.

ZHAO, H. and SHIBASAKI, R. (2005). A novel system for tracking pedestrians using multiple single-row laser-range scanners. *Systems, Man and Cybernetics, Part A : Systems and Humans, IEEE Transactions on*, 35, 283–291.

APPENDIX A Example Setup Files for TrOPed

setup.ini - primary setup file

```
[ConfigFiles]
nconfigs : 1
config0 : Testconfig.cfg

[HomographyOptions]
nohomography : 0
includehomoaltitudemod : 1
shiftgthomo : 1
meterspperpixel : 0.016533
homofilename : homography.txt
pointcorrfilename : pct.txt
gthomofilename : gthomo.txt
videoframefile : PolyTestSnapshot.png
worldfile : Floorplantestworld.png

[RunSettings]
nrunlines = 2
runline0 = feature-based-tracking PolyTestconfig.cfg –tf
runline1 = feature-based-tracking PolyTestconfig.cfg –gf

[GeneralSettings]
weightmota : 1
maxiterations : 2058
relativechange : 0
maxnchanges : 1
storagefilename : PolyTeststorage.csv
videofilename : PolyTest.mp4
groundtruthsqlite : groundtruthtest.sqlite
sqlitefilename : PolyTestCal.sqlite
```

```
[OptimizationParameters]
probconstant : 1
tinit : 20
maxmatchdist : 1
lamda : 0.5
emax : -100
```

varparams.txt - variable parameters file

```
0,feature-quality,float,ratio,0.5,0.000001,1,2
0,min-feature-distanceklt,float,add,5,0,10,0.4
0,window-size,int,add,5,3,10,1
0,pyramid-level,int,add,5,3,10,1
0,ndisplacements,int,add,3,2,4,1
0,min-feature-displacement,float,add,0.05,0,1,0.1
0,acceleration-bound,float,add,2,1,3,0.4
0,deviation-bound,float,add,0.5,0,1,0.2
0,smoothing-halfwidth,int,add,6,0,11,1
0,min-tracking-error,float,add,0.15,0.01,0.3,0.04
0,min-feature-time,int,add,15,5,25,1
0,mm-connection-distance,float,add,2,0.5,5,0.8
0,mm-segmentation-distance,float,add,1,0.1,prev,0.4
0,min-nfeatures-group,float,add,2.5,1,4,1.5
```

statparams.txt - static parameters file

```
0,video-filename = test.mp4
0,database-filename = test.sqlite
0,homography-filename = homography.txt
0,intrinsic-camera-filename = none
0,distortion-coefficients = -0.11759321
0,distortion-coefficients = 0.0148536
0,distortion-coefficients = 0.00030756
0,distortion-coefficients = -0.00020578
0,distortion-coefficients = -0.00091816
0,undistorted-size-multiplication = 1.31
0,interpolation-method = 1
```

```
0,load-features = false
0,display = false
0,video-fps = 30
0,measurement-precision = 3
0,frame1 = 0
0,nframes = 0
0,max-nfeatures = 1000
0,use-harris-detector = false
0,k = 0.133561
0,nframes-velocity = 3
0,max-number-iterations = 20
0,min-feature-eig-threshold = 0.0001
0,max-distance = 5
0,min-velocity-cosine = 0.188628
0,max-predicted-speed = 50
0,prediction-time-horizon = 5
0,collision-distance = 1.8
0,crossing-zones = false
0,prediction-method = na
0,npredicted-trajectories = 10
0,min-acceleration = -9.1
0,max-acceleration = 2
0,max-steering = 0.5
0,use-feature-prediction = true
0,max-normal-acceleration = 2
0,max-normal-steering = 2
0,min-extreme-acceleration = 2
0,max-extreme-acceleration = 3
0,max-extreme-steering = 3
0,mask-filename = testmask.png
```