Ego-Motion Compensation: External Motion Detection with a Moving Camera

Kian Kenyon-Dean
Monica Patel

April 28, 2017

COMP 558 Final Project

Prof. Kaleem Siddiqi

McGill University

1 Introduction

The ability to track moving objects in a video made with a moving camera is a task that humans perform with remarkable ease and triviality (and they actually do so with two moving cameras, the eyes), yet as a problem for a computer vision system it turns out not to be so easy. The ability to solve the problem, however, would have deep ramifications for the development of artificial intelligence technology across a variety of spectrums, a few of which are listed below.

- Navigation and obstacle avoidance: making robots or self-driving cars that can navigate a complex moving environment (with pedestrians, for example) successfully without crashing into obstacles, perhaps for performing automated street cleaning or distributing horderves at a dinner party.
- Surveillance: improving the tracking algorithms used by the surveillance state's many cameras in order to, eventually, abolish privacy as we know it.
- Automated target tracking and navigation for military weaponry (such as tanks, drones, missiles, etc.).

This problem is characterized by a dual character of two independent motions. On the one hand, we have the movement of the moving camera (e.g., on a the robot) in the environment. On the other hand, we have the external motion of objects in the environment (e.g., pedestrians). While these movements are independent of each other, they appear blended in the robot's sensor data, in the camera's video feed. It is thus necessary to first *compensate* for the independent movement of the robot, after which we can attempt to determine and track the moving objects in the world.

Our overall exploration of the problem is based on the seminal work by Jung and Sukhatme in their works [1, 2]. The methods we use for solving the problem are based on sound approaches expressed in detail in [1]; we provide clear explanations and analyses of their approach and algorithms used, additionally exploring alternative ways to solve the problem with significantly less processing time.

The solution to the problem is thus formally defined by the authors in [1] as a two part process: *ego-motion compensation* and *motion detection*.

Ego-motion compensation can be solved with two different approaches, *direct* and *indirect* estimation. The *indirect* method relies on additional sensor data, such as that which is obtained from a GPS or gyroscope, in order to estimate the robot's pose transformation between two frames. This approach is not necessary for performing ego-motion compensation and may be undesirable in small contexts because it requires additional sensor data to be obtained; we instead opt for the *direct* approach, as is pursued in [1]. The *direct* estimation of the pose transformation undergone by the robot (i.e., estimating the robot's movement) is computed using features extracted from the two image frames obtained by the robot at consecutive time steps. The methods and algorithms we use to perform this ego-motion compensation are discussed in Section 3.1.

Motion detection is a very broad problem that has historically been solved in many different ways, most of which are based on Monte Carlo approaches for solving general filtering problems in signal processing. The approach we use is inspired by [1]; we use a general particle filter algorithm to determine the moving objects in the compensated images, where the particles eventually converge around the moving objects after a certain amount of frames. We discuss the details of our approach to motion detection in Section 3.2.

The rest of this paper is organized as follows. In Section 2 we explore related work on this problem, from the first principles that preceded and set the basis for the work in [1], to applications and extensions of the work in [1] to even more complex scenarios. In Section 3 we provide detailed explanations of the algorithms we experiment with, including those found in [1] and novel variants of their algorithms we explore. In Section 4 we discuss the dataset we use. In Section 5 we provide results obtained with the different algorithms we implemented. In Section 6 we provide a critical discussion of the methods we use and an analysis of the problem in general, along with concluding remarks.

2 Related Work

Jung and Sukhatme began their exploration into the problem of external motion tracking on a mobile robot in 2004 [2]. Their work matured and developed over time, culminating into a more thorough, systematic approach to solving the problem in 2010 [1], where they also present a fusion approach to combining the estimated motion tracking and ego-motion compensation with laser-rangefinder data to perform position estimation in 3D space. In this project, we focus on ego-motion compensation and motion detection, leaving 3D position estimation to future work.

Their approach could not have been pursued without the first principles of many computer vision problems already having been systematically expressed and explored. For example, the task of ego-motion compensation is pursued as an extension of the classical *Lucas-Kanade* image registration problem [3], and is acknowledged as such by Jung and Sukhatme [1, 2]. Additionally, the task of motion detection has been pursued in many contexts; seminal work in the task is systematically explored in the 2001 literature survey found in [4], with many advances made just five years later, as expressed in the the 2006 survey conducted by the same authors [5]. One of the earliest papers on motion detection was in 1980 [6], where the authors detect and model human motion from image data in an idealized context.

The principles and approaches found in [1] have been extended to even more complex tasks; for example, in poorly illuminated environments with thermal camera data [7], or in environments where an arbitrary number of pedestrians are tracked by *multiple* moving robots with cameras on them [8]. Jung and Sukhatme's work has even assisted in solving the problem of human-robot interaction (HRI), where their work has specifically played a role in inspiring the implementations of HRI systems that explore the effect of eye contact between robots and humans [9, 10].

Below we review in detail historical approaches to solving the problems that characterize this task. Both the problems of ego-motion compensation and external motion detection (as well as algorithms used to solve them) have rich literatures and histories of study behind them, even when considered in isolation from each other.

2.1 Ego-Motion Compensation

The general problem of ego-motion compensation is applicable across a broad domain of vision problems. This is to be expected because if one wants to reason about a sequence of images, one is significantly limited if the camera has to remain stationary. Yet, even if the camera remains stationary the problem of ego-motion compensation arises due to infinitesimal movements or unintended shaking of the camera. It is for this reason that the

problem of image stabilization for both stationary and moving cameras requires a model capable of recovering and compensating the image sequence for the camera's ego-motion to stabilize the video, as is explored in [11, 12], and surveyed in [13].

Compensating for the ego-motion of a camera, therefore, is a crucial first step for any computer vision system that seeks to reason about a sequence of images. In the domain of robotics, this problem is even more clearly necessary to solve. For example, in [14] the authors find that ego-motion compensation is very necessary for having successful autonomous navigation and landing of an aircraft on a runway, a runway which has obstacles that must be detected (and can only be detected after ego-motion compensation) in order to avoid high-risk collisions. Their approach is in an idealized scenario in which the obstacles are assumed to be stationary, but it is nonetheless a non-trivial problem. This idealization is done away with in the context of the similar scenario presented in [15], where the authors have to compensate for the ego-motion of an airborne moving platform in order to track moving and stationary targets on infrared image data.

In [16], ego-motion compensation is explored in the context of a 360° omni-directional camera, where the traditional approach to compensation had to be significantly extended because of the specific nature of omni-directional cameras, such as the inevitable distortion found in 360° images (similar to images obtained with a "fish-eye" lens). The authors successfully compensate for ego-motion in this setting and thus are able to detect and track moving vehicles along a road surrounding the camera (mounted on a moving vehicle) with encouraging results up to a speed of 65 miles per hour (105 kilometers per hour).

2.2 Motion Detection

In the idealized setting of a stationary camera in an environment with a static background, the problem of motion detection is simplified, where external object movement can be determined by taking the difference between frames obtained at two different time steps. Even in this scenario, however, to actively track an arbitrary number of moving objects, not just determine if movement is happening, is a problem that requires a certain amount of engineering to do correctly. This requires the use of some sort of tracking algorithm, which is often expressed as a filtering problem, a field so rich that it is necessary to devote a separate discussion for it, found below in Section 2.3.

Movement can be detected by subtracting away the background from the foreground; however, this is not always easy in real-world settings, particularly due to changes in illumination and occlusion. One common approach to segment foregrounds and backgrounds uses Gaussian mixture models [17, 18, 19], which are well-understood in the context of computer vision. These approaches most accurately solve the motion detection problem in settings where the background is as static as possible, in environments with controlled lighting conditions. However, recent work [20] has been able to apply extensions of these models by using advanced Bayesian methods in less ideal scenarios, allowing for successful foreground-background segmentation in the context of varying degrees of illumination.

2.3 Motion Detection as a Filtering Problem

Filtering is a general problem in which a "state" must be estimated in order to determine the true value of a system. This estimation often must be performed in the context of a series of incomplete, noisy signal observations of the system. In the case of a linear system, it turns out that the Kalman Filter (classically introduced in 1960 by Kalman [21]) produces

the optimal estimate of the system [22]. Unfortunately, the vast majority of systems that need to be approximated are nonlinear, which makes the problem considerably harder and eliminates the guarantee of optimality when approximating the system. Nonetheless, many variants of the Kalman Filter have been created to extend it to the case of nonlinear systems, such as the Extended Kalman Filter (EKF) [22], and the Unscented Kalman Filter (UKF) which, in [23, 24], is claimed to objectively improve upon the EKF. In [8], the authors combine a Kalman filter with a global-nearest-neighbor data association algorithm to track detected pedestrians. Using a standard filtering algorithm, their task is made more complex since the authors have to synthesize data obtained from multiple moving cameras.

Our problem of external movement detection and tracking can be viewed as a Bayesian filtering problem where we need to determine the state (e.g., position and velocity) of moving objects by approximating the posterior probability distribution of the state (discussed formally in Section 3.2). While Kalman Filters are well-suited for problems in which there are a predetermined number of states to be approximated (e.g., approximating the global position of a plane using GPS and other sensor data), they do not naturally extend (without significant engineering [8]) to the case of when a varying number of states need to be approximated. Our problem is defined by an arbitrary number of moving objects which may move in and out of vision (e.g., out of our sensor range) — it is thus necessary to have a filtering algorithm capable of handling this scenario.

Fortunately, we have sequential Monte Carlo methods, commonly called particle filters, which are naturally suited to our problem. Indeed, as indicated in [1, 2, 25], particle filters have the inherent advantage of "multi-modality"; a single set of particles can, in theory, track multiple varying moving objects that come in and out of a sequence of images. With the term "particle filter" coined in [26], and their application to visual tracking being formalized in [27] and extended to mobile robotics in [25], particle filters are argued to be appropriate alternatives to Kalman filters which make the limiting assumption of unimodal Gaussian densities. Instead, particle filters actively represent multiple alternative hypotheses about the state of the system [27].

A commonly used improvement to the standard particle filter algorithm is the *adaptive particle filter* [28], which allows for dynamic variable change in the number of particles according to a preset range. This dynamic changing results in a marked improvement in efficiency, and is used by Jung and Sukhatme in [1, 2], the main works our implementation is inspired by. In [7] we find particle filters being implemented for a task similar to ours. The authors use adaptive particle filters with an online-learning random forest classifier to determine the confidence of detecting a human in order to represent the likelihood of a particular state. Their novel approach obtains encouraging results when using a thermal camera under poor illumination, with shadows and cluttered backgrounds making the task more difficult.

3 Algorithms

The algorithms we use are based on those presented by Jung and Sukhatme in [1, 2]. We present and elucidate their approaches, as well as the variants we experiment with for the purposes of our implementation.

3.1 Ego-Motion Compensation

Ego-motion compensation can be understood as a "coordinate conversion procedure" [1] between images obtained at consecutive time steps, I_t and I_{t+1}. The camera undergoes some real world, objective transformation T^* between time steps t and $t+1$, which is unknown and must be approximated. As discussed above (Section 1) we pursue the *direct* approach to approximating T^* based on using salient features extracted from I_t and I_{t+1}. If our approximation T of the transformation is good, then the difference between I_{t+1} and $T(I_t)$ will be minimal. However, T will not be able to directly account for the motion of external objects because these objects move independently of the camera, and features extracted from them may lead to incorrect transformation approximations. This therefore necessitates a transformation model where the ego-motion compensation is not sensitive to external object motion. In practice, this is solved via an intuitive outlier detection algorithm for determining the transformation function.

In Section 3.1.4 we present the novel, more efficient approach we implemented, which is inspired by theirs. Our approach is characterized by obtaining decent transformation approximations without necessitating either an outlier detection step, nor an expensive least-squares optimization. This results in a more efficient algorithm for ego-motion compensation at the cost of slightly more error-prone transformation estimations, which are fortunately naturally compensated for in the particle filtering step.

3.1.1 Feature Extraction

Image feature extraction is an integral component for solving computer vision problems. Strong features provide the basis for many algorithms that seek to determine an alignment between two images, and also play an important role in object recognition problems. Historically, *corner features* have been used, which are obtained with algorithms such as the Harris corner detector [29] or the KLT (Kanade-Lucas-Tomasi) corner detector [3, 30]. These features are characterized by a sharp shift in intensity within a surrounding neighborhood around a certain pixel; i.e., when edges come together at a sharp angle.

More recent developments in image feature extraction have been largely influenced by Lowe's SIFT (Scale Invariant Feature Transform) features [31, 32], from which a vast amount of literature has emerged for extending and optimizing the original principles of SIFT. On a high level, these features are often claimed to be "more advanced" than traditional corner features [1] because these features indicate points of interest in an image in a way that is invariant to image scale, rotation, and are partially robust to other changes, such as illumination and noise. Despite these benefits, Jung and Sukhatme claim that the SIFT method of searching for keypoints and descriptors is too computationally expensive and "not suitable for real-time applications" [1]. With modern advances in technology, however, this argument against SIFT features is not as valid as it once was.

SURF (Speeded Up Robust Features) [33] features offer an improvement in the computational complexity of SIFT by extracting features much faster at the cost of slightly less robust features. Jung and Sukhatme use the KLT feature tracking algorithm (as described in [34, 30]) in their implementation "mainly due to its computational efficiency" [1]. In our implementation, we use SURF features to obtain a feature correspondence between images.

In both our implementation and Jung and Sukhatme's, the following general approach is invoked for the feature extraction process. Given two consecutive images I_t, I_{t+1}, a set of features F_t is extracted from I_t and a corresponding feature set F_{t+1} is obtained by tracking the same features found in I_t within I_{t+1}.

3.1.2 Transformation Estimation

Now that we have computed the feature correspondence $\langle F_t, F_{t+1} \rangle$, we can estimate the ego-motion of the camera by using a transformation model. Our aim is to compute a transformation matrix T such that the difference $F_{t+1} - T(F_t)$ is minimized. In other words, we want to determine a transformation function that is able to align (as much as possible) the image obtained at previous time step t to the image obtained at the next time step $t+1$. This transformation is obtained using the salient features rather than pixel values because features capture qualities and points of interest in the images that basic pixel values cannot capture on their own. Below we present the three transformation models explored in [1, 2]:

Affine model:

$$\begin{bmatrix} f_x^{t+1} \\ f_y^{t+1} \end{bmatrix} = \begin{bmatrix} a_0 f_x^t + a_1 f_y^t + a_2 \\ a_3 f_x^t + a_4 f_y^t + a_5 \end{bmatrix} \tag{1}$$

Bilinear model:

$$\begin{bmatrix} f_x^{t+1} \\ f_y^{t+1} \end{bmatrix} = \begin{bmatrix} a_0 f_x^t + a_1 f_y^t + a_2 + a_3 f_x^t f_y^t \\ a_4 f_x^t + a_5 f_y^t + a_6 + a_7 f_x^t f_y^t \end{bmatrix} \tag{2}$$

Pseudo-perspective model:

$$\begin{bmatrix} f_x^{t+1} \\ f_y^{t+1} \end{bmatrix} = \begin{bmatrix} a_0 f_x^t + a_1 f_y^t + a_2 + a_3 f_x^{t^2} + a_4 f_x^t f_y^t \\ a_5 f_x^t + a_6 f_y^t + a_7 + a_8 f_x^t f_y^t + a_9 f_y^{t^2} \end{bmatrix} \tag{3}$$

In each of these models, the constants a_i must be obtained according to some optimization algorithm. Jung and Sukhatme use a least squares optimization over all N feature correspondences $\langle f_i^{t+1} \in F_t, f_i^{t+1} \in F_{t+1} \rangle$, where the cost function to be minimized is defined as follows:

$$J = \sum_{i=1}^{N} \left(f_i^{t+1} - T(f_i^t) \right)^2 \tag{4}$$

When convergence is reached using some optimization software, the model parameters arrive at their optimal values. Each model, however, offers significantly different behaviors and assumptions inherent in T. For example, the affine model in Equation 1 assumes a linear transformation between images, and can be used to approximate most standard ego-motions of the camera, such as translation, rotation, and scaling [1]. However, when more complex changes occur (particularly if the real time difference between t and $t+1$ is larger than one second), the linear affine model is incapable of capturing the real translation. Indeed, often we observe motions that are projective operations, such as when one is in a moving car and closer objects appear to move more quickly than ones farther away. This motivates the use of a nonlinear transformation model, such as the bilinear model found in Equation 2 and the more complex pseudo-perspective model in Equation 3. However, more complex does not necessarily mean better; indeed, these models are prone to overfitting, particularly if the features they fit to correspond to moving objects [1], which we do not want to consider in the first place since moving objects are independent of the camera's ego motion. Jung and Sukhatme opt for the bilinear model [1]; preliminary experiments in our implementation and dataset found that the affine model offered the best performance (see Sections 3.1.4 and 4).

3.1.3 Outlier Detection

Jung and Sukhatme offer a compelling approach computing the transformation matrix without using features corresponding to moving objects, called outlier features [1]. This involves computing two transformation matrices T_0 and T. T_0 is computed as described above using the full feature set $F = \langle F_t, F_{t+1} \rangle$. The problem with T_0 is that it may have been computed according to outlier feature correspondences; i.e., according to feature correspondences on moving objects, which we do not want to take into account because they will cause an incorrect transformation to be computed. The next step is thus to partition F into two sets, F_{in}, F_{out} according to a pre-tuned hyperparameter ϵ, as described below in Equation 5:

$$\begin{cases} f_i \in F_{in} & \text{if } |f_i^{t+1} - T_0(f_i^t)| < \epsilon \\ f_i \in F_{out} & \text{otherwise} \end{cases} \tag{5}$$

We now have a set of M feature correspondences in F_{in} ($M < N$) that (hopefully) do not include correspondences between features of moving objects. The final step is therefore to obtain the final transformation matrix T with another least squares optimization, except this time it is computed only using the non-outlier features we partitioned into the set F_{in}.

3.1.4 Our Single-Step Implementation

In our view, Jung and Sukhatme's approach described above is sub-optimal for several reasons. Firstly, it treats all features equally, which is inadvisable since certain feature correspondences are bound to be more correct than others. Secondly, there is no guarantee that the outlier features actually correspond to external object movement; hand-tuning of the ϵ parameter according to the designer's intuitions is the only way to have at least a semblance of a guarantee for correctness, which is a dubious necessity. Thirdly, it is computationally expensive; two least squares optimizations have to be performed (nonlinear ones at that, if using the bilinear or pseudo-perspective models), and the algorithm for outlier detection in Equation 5 requires N matrix multiplications between T_0 and each feature correspondence f_i. It is for these reasons that we propose a more efficient approach to ego-motion compensation, which takes inspiration from what is presented above.

Rather than using the entire set of features (some of which may be incorrect or misinformative), we only use the best three feature correspondences made with the SURF key point features, where goodness is measured by the level of similarity between the areas matched across the images. By using only the three best feature correspondences, we are almost always guaranteed to be using only features that representing stationary objects, since features extracted from the moving objects are likely not linking as similar areas in the image as those that link stationary ones. With these, we compute the transformation matrix with a closed form solution since, with the three correspondences, we are naturally able to generate six equations to solve for the six unknown variables in the affine model (see Equation 1). Although this may seem like quite a small amount of information to compute such an important transformation, it turns out that the ego-motion is compensated relatively well, as can be observed in the results explained in Section 5; additionally, our compensated frame differences look comparable to the frame differencing results presented by Jung and Sukhatme in [1] in Figure 5.

3.2 Motion Detection with Adaptive Particle Filter

After the ego-motion is compensated, the presence of motion of an external object can be calculated by image differencing which is given by Equation 6 below:

$$I_d(x,y) = \begin{cases} |(I_c(x,y) - I^t(x,y))| & if(x,y) \in R \\ 0 & otherwise \end{cases} \tag{6}$$

However such frame differencing inherently contains two types of errors. The first error results from imperfect ego motion compensation, and the second error occurs due to the introduction of new information in the image because the camera position was changed. To deal with these two errors, a probabilistic method is used to approximate movement of external object. The normalized pixel values in the difference images can be interpreted as a representation of the probability that there exists a moving object in that pixel position. This is modeled using a Bayesian formulation, where a state x_t is to be calculated. The posterior probability distribution of the state is given by Equation 7 below:

$$p_m(x_t) = n^t P(I_d^t|x_t) \int P(x_t|x_{t-1}) P_m(x_{t-1}) dx_{t-1} \tag{7}$$

Where the posterior probability distribution is updated recursively using motion model $P(x_t|x_{t-1})$ and the measurement or perception model, $P(I_d^t|x_t)$ where I_d^t is image difference at the time t.

We implement this recursive approach using a particle filter, where the state of the system is $s_t = [x, y]$, and the motion model is given by the Equation 8 below:

$$s_i^{t+1} = s_i^t + \delta t * \dot{s_i^t} + noise \tag{8}$$

An initial uniform distribution over all particle is considered; the particle moves in the environment using the above motion model, and belief of the state is updated using the measurement model given by image differencing. In the belief update step the weights of all the particles are recalculated and same number of particles are picked with replacement according to probability proportional to the weight. The weight of each particle is given by Equation 9 below:

$$w_i^t = \frac{1}{m^2} \sum_{j=-\frac{m}{2}}^{j=\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{k=\frac{m}{2}} I_{diff}^t(s_i^t(x) - j, s_i^t(y) - k) \tag{9}$$

4 Dataset

Images from the PARSE-27k (Pedestrian Attribute Recognition in Sequences) data set[1] created by the Visual Computing Institute at RWTH Aachen university (Rheinish-Westphalian Technical University)[2] was used for the project. The dataset was specically created for catering to the constraints of mobile vision applications. The data is based on eight video sequences of varying length taken by a moving camera in a city environment where every 15^{th} frame is used. We use these images to demonstrate working of ego-motion compensation and external object tracking using particle filter. However, one difficulty found in this

[1] See `http://www.vision.rwth-aachen.de/parse27k/`

[2] See `http://www.vision.rwth-aachen.de`

data is that the external moving objects often do not stay in the camera's reference frame for more than a few frames, which means that the particle filter must adapt to these rapid changes quite quickly.

5 Results

In Figure 1 below[3] we observe an example sequence of six images in our dataset. One can observe that the camera is moving forward, and that the pedestrians in the image at $t = 0$ move entirely out of the frame by $t = 5$, meaning that they must be localized rapidly in order to be detected.

In Figure 5 we observe the feature correspondence according to the top 3 best corresponding features across the images from the two time steps. We compute the transformation matrix directly with these feature correspondences. For example, in the correspondence from $t = 0$ to $t = 1$ we observe one of the best features, according to SURF, was the distinctive and unique window in the top of the leftmost building.

In Figures 2 and 3 we observe the performance of our ego-motion compensation algorithm. At first look, one may be quite surprised that such a small number of feature correspondences can lead to a transformation function that creates better compensations than simple frame differencing with the original images.

In the first three time steps we observe that a significant amount of compensation is performed because the top left corner almost completely blacks out when compared to the uncompensated frame differencing. This thus means that the transformation model correctly reorients the image to align with the one in the next time step at those locations. We also observe in the latter three time steps that much of the buildings and ground is blacked out when compared to what occurs without the compensation. In addition, we find that the moving pedestrians are more clearly observable and highlighted in the compensated differencing. Nonetheless, we do observe that there is a significant portion of the image that is not blacked out, despite the fact that these parts of the image do not correspond to moving objects. This leads to difficulties for the particle filter.

In Figure 4 we observe the performance of our particle filter algorithm. It is expected that in the first few time steps the particles are randomly distributed. However, we had hoped that by around $t = 4$ or $t = 5$ the particles would converge around the moving pedestrians. This, however, was not the case — many particles incorrectly tracked the errors caused by improper compensation, such as the bricks on the ground or the top of the tree.

6 Conclusion

We have presented an approach to ego-motion compensation for external movement detection on a moving camera, inspired by the work of Jung and Sukhatme in [1, 2]. Our literature review and experience in implementing this work has led us to arrive at certain conclusions and reflections on the work we have pursued.

It is undeniable that this is a difficult task; however, a process of abstraction and idealization should have been more attentively pursued. This would have allowed us to develop our system more thoroughly, as has been historically done in computer vision research. Indeed, the results we present show that a more complete understanding of the inner workings of

[3]Figures begin after the references section.

our algorithms could have been elucidated had we, for example, obtained the data ourselves in idealized environments, such as ones in simple halls with basic camera motions and external object movement. This process of idealization would have allowed us to isolate specific aspects of our approach because if the algorithms do not work in idealized environments, they certainly will not work in complex ones.

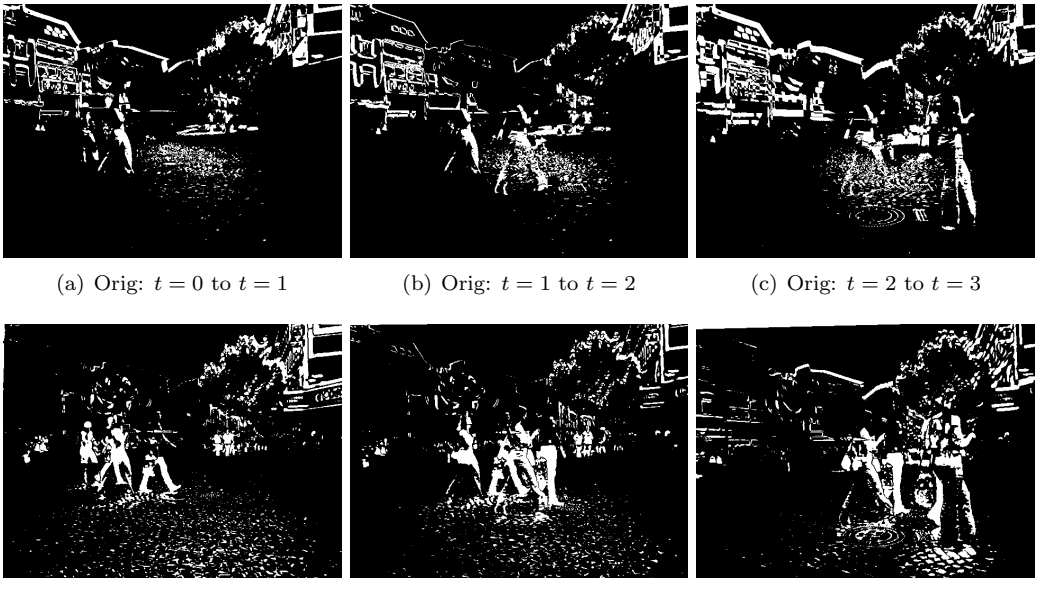
Although this alternative experimental design would have been desirable, we discovered through the literature review that there is no systematic way to evaluate these kinds of systems. Indeed, in many expositions of this work there is no universally accepted way to objectively assess the quality of this kind of system. It was therefore unclear as to how to pursue an objective analysis of our system, thus leading us to have to rely on randomly sampling a sequence of images and analyzing them subjectively.

The task of ego-motion compensation for external movement detection is challenging, but rewarding if done correctly. Indeed, in the future when this problem is perfected we can expect more technological innovation, from self-driving cars to enhanced surveillance systems. Our exposition of this problem has elucidated and informed our understanding of the historical development of computer vision technology, and has been an invaluable experience in understanding the particular evolution of implementing a system to solve these kinds of problems.

(a) $t = 0$ (b) $t = 1$ (c) $t = 2$

(d) $t = 3$ (e) $t = 4$ (f) $t = 5$

Figure 1: The original images from $t = 0$ to $t = 5$.



(a) Orig: $t = 0$ to $t = 1$ (b) Orig: $t = 1$ to $t = 2$ (c) Orig: $t = 2$ to $t = 3$

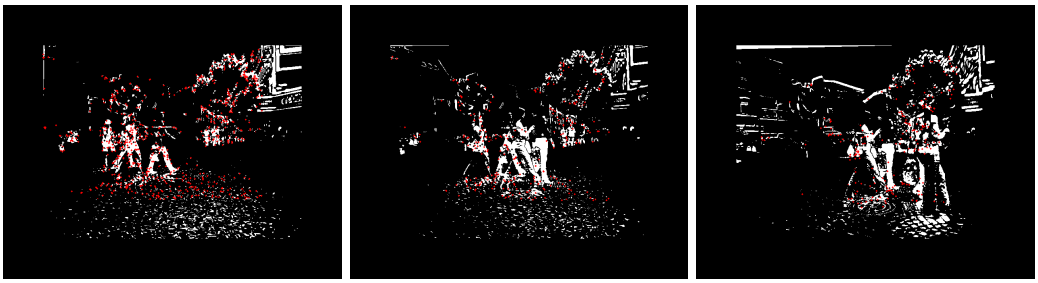
(d) Comp: $t = 0$ to $t = 1$ (e) Comp: $t = 1$ to $t = 2$ (f) Comp: $t = 2$ to $t = 3$

Figure 2: Comparison between frame difference of the original (Orig) images and compensated (Comp) images from $t = 0$ to $t = 3$.

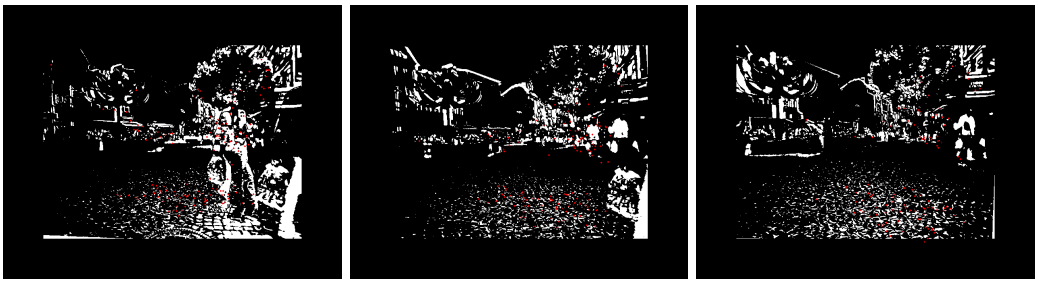
(a) Orig: $t = 3$ to $t = 4$ (b) Orig: $t = 4$ to $t = 5$ (c) Orig: $t = 5$ to $t = 6$

(d) Comp: $t = 3$ to $t = 4$ (e) Comp: $t = 4$ to $t = 5$ (f) Comp: $t = 5$ to $t = 6$

Figure 3: Comparison between frame difference of the original (Orig) images and compensated (Comp) images from $t = 3$ to $t = 6$.

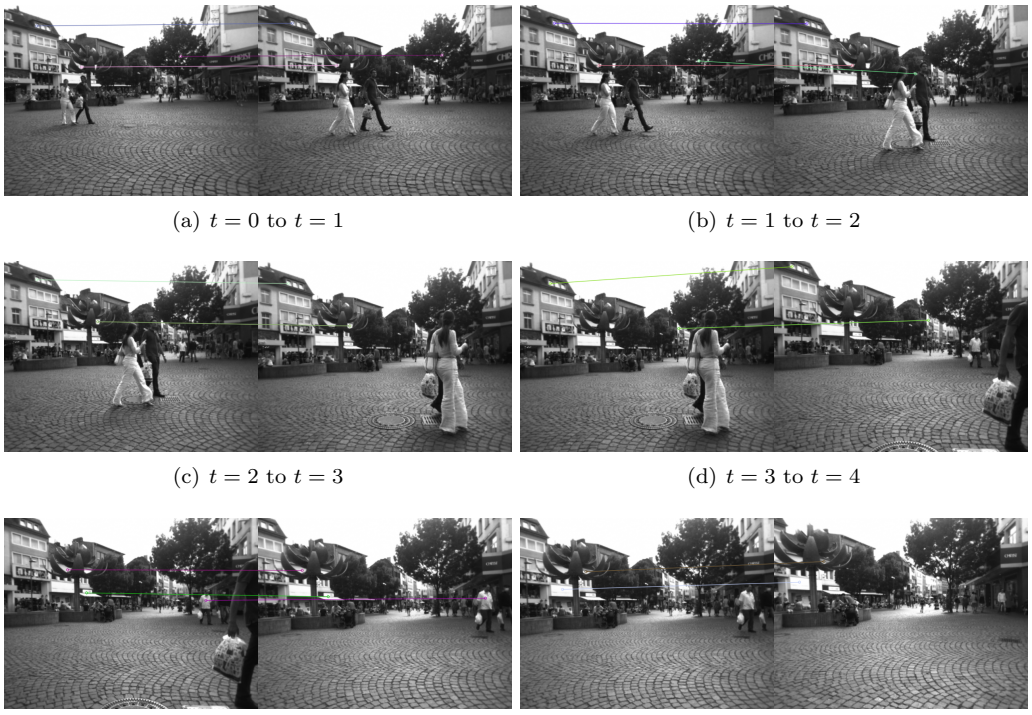


(a) $t = 0$ to $t = 1$ (b) $t = 1$ to $t = 2$ (c) $t = 2$ to $t = 3$



(d) $t = 3$ to $t = 4$ (e) $t = 4$ to $t = 5$ (f) $t = 5$ to $t = 6$

Figure 4: Particle filter applied to the compensated images from $t = 0$ to $t = 6$.



(a) $t = 0$ to $t = 1$

(b) $t = 1$ to $t = 2$

(c) $t = 2$ to $t = 3$

(d) $t = 3$ to $t = 4$

(e) $t = 4$ to $t = 5$

(f) $t = 5$ to $t = 6$

Figure 5: Feature correspondence across images (showing only the top 3 features used for computing transformation).

References

[1] Boyoon Jung and Gaurav S Sukhatme. Real-time motion tracking from a mobile robot. *International Journal of Social Robotics*, 2(1):63–78, 2010.

[2] Boyoon Jung and Gaurav S Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *International Conference on Intelligent Autonomous Systems*, pages 980–987, 2004.

[3] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[4] Thomas B Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer vision and image understanding*, 81(3):231–268, 2001.

[5] Thomas B Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2):90–126, 2006.

[6] Joseph O'rourke and Norman I Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 522–536, 1980.

[7] Byoung Chul Ko, Joon-Young Kwak, and Jae-Yeal Nam. Human tracking in thermal images using adaptive particle filters with online random forest learning. *Optical Engineering*, 52(11):113105–113105, 2013.

[8] Masataka Ozaki, Kei Kakimuma, Masafumi Hashimoto, and Kazuhiko Takahashi. Laser-based pedestrian tracking in outdoor environments by multiple mobile robots. *Sensors*, 12(11):14489–14507, 2012.

[9] Michihiro Shimada, Yuichiro Yoshikawa, Mana Asada, Naoki Saiwaki, and Hiroshi Ishiguro. Effects of observing eye contact between a robot and another person. *International Journal of Social Robotics*, 3(2):143–154, 2011.

[10] Haibin Yan, Marcelo H Ang, and Aun Neow Poo. A survey on perception methods for human–robot interaction in social robots. *International Journal of Social Robotics*, 6(1):85–119, 2014.

[11] Zhigang Zhu, Guangyou Xu, Yudong Yang, and Jesse S Jin. Camera stabilization based on 2.5 d motion estimation and inertial motion filtering. In *IEEE Int. Conf. on Intelligent Vehicles*, pages 329–334, 1998.

[12] Alberto Censi, Andrea Fusiello, and Vito Roberto. Image stabilization by features tracking. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 665–667. IEEE, 1999.

[13] Carlos Morimoto and Rama Chellappa. Evaluation of image stabilization algorithms. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 5, pages 2789–2792. IEEE, 1998.

[14] Tarak L Gandhi, Sadashiva Devadiga, Rangachar Kasturi, and Octavia I Camps. Detection of obstacles on runway using ego-motion compensation and tracking of significant features. In *Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on*, pages 168–173. IEEE, 1996.

[15] Alper Yilmaz, Khurram Shafique, and Mubarak Shah. Target tracking in airborne forward looking infrared imagery. *Image and Vision Computing*, 21(7):623–635, 2003.

[16] Tarak Gandhi and Mohan Trivedi. Parametric ego-motion estimation for vehicle surround analysis using an omnidirectional camera. *Machine Vision and Applications*, 16(2):85–95, 2005.

[17] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 246–252. IEEE, 1999.

[18] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE, 2004.

[19] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780, 2006.

[20] Andrew B Godbehere, Akihiro Matsukawa, and Ken Goldberg. Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. In *American Control Conference (ACC), 2012*, pages 4305–4312. IEEE, 2012.

[21] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[22] Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. 1995.

[23] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997.

[24] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee, 2000.

[25] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1-2):99–141, 2001.

[26] Pierre Del Moral. Non-linear filtering: interacting particle resolution. *Markov processes and related fields*, 2(4):555–581, 1996.

[27] Michael Isard and Andrew Blake. Condensation—conditional density propagation for visual tracking. *International journal of computer vision*, 29(1):5–28, 1998.

[28] Dieter Fox. Kld-sampling: Adaptive particle filters. In *NIPS*, volume 14, pages 713–720, 2001.

[29] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[30] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. 1991.

[31] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[32] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[33] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. *Computer vision–ECCV 2006*, pages 404–417, 2006.

[34] Jean-Yves Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5(1-10):4, 2001.