Augmented Reality: Active Appearance Model and Video Object Tracking

Term Paper (Spring 2005)

By

Chan Pik Wah

Supervised by Prof. Michael R. Lyu

Department of Computer Science and Engineering

The Chinese University of Hong Kong Shatin, N.T., Hong Kong

Abstract

Augmented Reality (AR) technology for digital composition of animation with real scenes is to bring new experience to the viewers. Augmented Reality is a form of human-machine interaction. The key feature of the Augmented Reality technology is to present auxiliary information in the field of view for an individual automatically without human intervention. To achieve the new Augmented Reality experience, object tracking is one of the hey point. Object tracking is the key component in Augmented Reality which provide the registration function to the system. To tackle this problem, we study the current technologies of augmented reality, object tracking and active appearance model. In this paper, a survey of augmented reality, object tracking and active appearance mode are presented and the proposed system architecture and its components are described in detail.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Augmented reality is a new area of research in which a virtual world is overlayed on top of the real world. Instead of the completely immersive environment of virtual reality, augmented reality attempts to use the flexibility of the digital world to enhance the environment in which we all live. Users of an augmented reality system are able to maintain the context of their surrounding environment, while still obtaining the benefit of additional sensory input and information.

One of the key step of augmented reality is registration to the system. Object tracking is the key technologies for this problem. In augmented reality system, a registration is needed such that the objects in the video is being detected and tracked. Then, the virtual object can be placed onto the objects of the video. In object tracking system, a model is need to be applied to estimate the shape and the motion of the objects. The Active Appearance

Model (AAM) algorithm has proved to be a successful method for matching statistical models of appearance to new images. Thus, in our newly proposed scheme, we applied the AAM in the object tracking system in the modelling part. Also, Kalamn filters are added to the fitting part of the AAM fitting algorithm to increase the speed of the tracking system.

1.2 Research Objective

Object tracking is an important techniques in the augmented reality area. There have been many approaches to object tracking described in the literature. The approaches differs in the goals of the tracking processes, the assumptions that are made about the scene, the types of moving objects and their motion characteristics and the camera motion. However, there are still number of problems need to be solved.

Active appearance model is a recent research area for modelling objects. It is a powerful tools for estimating the model and the appereance of the objects.

In our research, we analyze the current technologies of object tracking and the active appearance model. We propose a object tracking algorithm which applied the active apperance model, Kalman filter and occulsion detection.

1.3 Contribution

In the paper, we have the following main contributions:

- Survey on object tracking technologies

- Survey on Active Appearance Models (AAMs)
- Proposed a real time tracking system with AAMs and Kalman filters
- Proof of the general nature of AAMs

1.4 Structure of Report

This paper is organized into 4 chapters. The next chapter introduces the issues related to augmented reality, active appearance model and the object tracking techniques, a brief summary for current tracking techniques and a review for active appearance model are provided. A proposed object tracking system is described in chapter 3 and the experimental results are followed by. Finally, a conclusion and the future direction would be given in chapter 4.

Chapter 2

Background

Virtual Reality (VR) is stimulating the user's senses in such a way that a computer generated world is experienced as real. In order to get a true illusion of reality, it is essential for the user to have influence on this virtual environment. VR is used as a powerful interface with computers. By wearing a head mounted audio-visual display, position and orientation sensors, and tactile interface devices, one can actively inhabit an inclusive computer generated environment. With increasing computing power allowing for the processing of huge amounts of information in real time, VR technology has become more effective.

Augmented reality (AR) works on the same principles as virtual reality. However, different from VR where the user is immersed in a completely virtual environment, augmented reality overlays virtual objects and information over the real world. This is usually achieved by the use of see-through head mounted displays and tracking devices.

Let first have an overview of state-of-the-arts technologies in augmented reality and a literature review of current active appearance model and object tracking technologies.

2.1 Augmented Reality

2.1.1 Introduction

An augmented reality system supplements the real world with virtual objects that appear to coexist in the same space as the real world. An augmented reality system has the following properties

- Combine real and virtual objects in a real environment
- Runs interactively, and in real time
- Registers(aligns) real and virtual objects with each other

Ivan Sutherland, the computer graphics pioneer, is largely credited with the concept of Augmented Reality [2]. The Head-Mounted Display, which was prototyped by Ivan Sutherland in the late 1960s, started out as an Augmented Reality viewing device. It was so heavy that the device was called the "Sword of Damocles" because it had to be suspended from the ceiling to off set most of the weight from the head of the user. Since then Augmented Reality systems have come a long way and their progress has increased rapidly through the 1990's. The following sections take a look at some of the augmented reality applications that have been developed or are under research.

Augmented Reality is a variation of Virtual Reality as it is more commonly called. Virtual Reality technologies completely immerse a user inside a synthetic environment. While immersed, the user cannot see the real world around him. In contrast, Augmented Reality allows the user to see the real world, with virtual objects superimposed upon or composited with the real world. Therefore, AR supplements reality, rather than completely replacing it. Ideally, it would appear to the user that the virtual and real objects coexisted in the same space.

Milgram [3] defined a continuum of real-to-virtual environments, in which AR is one part of the general area of mixed reality, shown in Figure 2.1. In both augmented virtuality, in which real objects are added to virtual ones, and virtual environments (or virtual reality), the surrounding environment is virtual, while in AR the surrounding environment is real.

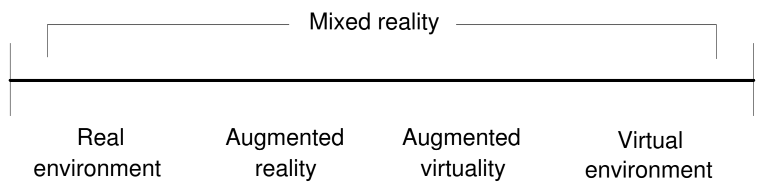

Figure 2.1: Milgrams real-to-virtual continuum

Augmented Reality enhances a user's perception of and interaction with the real world. The virtual objects display information that the user cannot directly detect with his own senses. The information conveyed by the virtual objects helps a user perform real-world tasks [4].

The critical problem with present augmented reality systems is the lack of real-time and accurate tracking. Since the information has to overlap with the real world, smallest errors in tracking information are detected by the human eye. Any mismatch between augmented objects and real objects can be discomforting and also result in incorrect information being given to the user.

There are numbers of enabling technologies for building compelling AR environments. Examples of these technologies include displays, tracking, registration and calibration.

2.1.2 Related Projects

In recent years, Augmented Reality technologies have been applied to different areas. Following are some projects related to Augmented Reality.

Boeing Computer Seattle is developing two real-time applications – one in virtual reality, the other one in augmented reality – aimed at putting more information directly in front of the engineers designing aircraft and the manufacturing workers who build them [5].

For many years, military aircraft and helicopters have used Head-Up Displays (HUDs) and Helmet-Mounted Sights (HMS) to superimpose vector graphics upon the pilot's view of the real world. Besides providing basic navigation and flight information, these graphics are sometimes registered with targets in the environment, providing a way to aim the aircraft's weapons. For example, the chin turret in a helicopter gunship can be slaved to the pilot's HMS, so the pilot can aim the chin turret simply by looking at the target. Future generations of combat aircraft will be developed with an HMD built into the pilot's helmet [6].

Augmented Reality technologies are also applied in satellites. The Autonomous Vision System on TeamSat (AVS) [7] is a fully autonomous star tracker and vision system. The objectives of the AVS were to verify, in space, multiple autonomous processes intended for spacecraft applications such as autonomous star identification and attitude determination, identification and tracking of non-stellar objects, imaging and real-time compression of image and science data for further ground analysis. The object tracking techniques are employed in the project.

2.1.3 Augmented Reality for Visualization

The use of augmented reality for visualization purposes has been restricted due to speed limitations in realistic real-time rendering. However, flat shaded renderings have been employed to use augmented reality for spatial visualization.

The collaborative design system developed at ECRC [8] is a demonstration of interactive graphics and real-time video for the purpose of interior design. The system combines the use of a heterogeneous database system of graphical models, an augmented reality system, and the distribution of 3D graphics events over a computer network [9].

The scenario for this application consists of an office manager who is working with an interior designer on the layout of a room. On a computer monitor they can see a picture of the room from the viewpoint of the camera. By interacting with various manufacturers over a network, they select furniture by querying databases using a graphical paradigm. The system provides descriptions and pictures of furniture that is available from the various manufactures who have made models available in their databases. Pieces or groups of furniture that meet certain requirements such as color, manufacturer, or price may be requested. The manager chooses pieces from this "electronic catalogue" and 3D renderings of this furniture appear on the monitor along with the view of the room. The furniture is positioned using a 3D mouse. Furniture can be deleted, added, and rearranged until the users are satisfied with the result; they view these pieces on the monitor as they would appear in the actual room. As they move the camera they can see the furnished room from different points of view Figure 2.2.

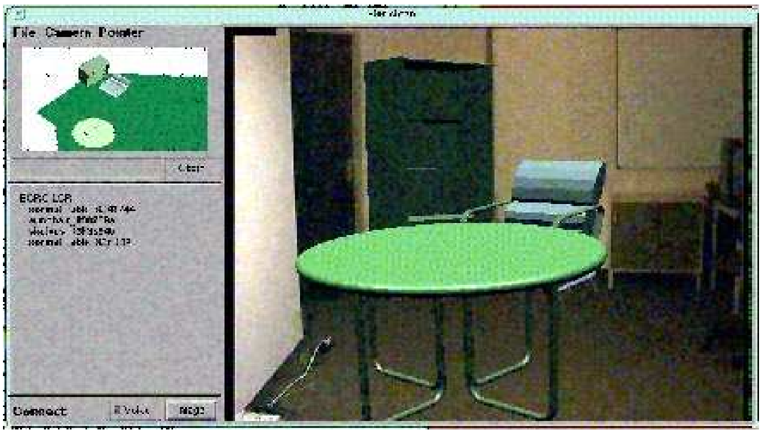

Figure 2.2: Collaborartive interioir design

The users can consult with colleagues at remote sites who are running the same system. Users at remote sites manipulate the same set of furniture using a static picture of the room that is being designed. Changes by one user are seen instantaneously by all of the others, and a distributed locking mechanism ensures that a piece of furniture is moved by only one user at a time. In this way groups of users at different sites can work together on the layout of the room. The group can record a list of furniture and the layout of that furniture in the room for future reference.

2.1.4 Augmented Reality for Maintenance/Repair Instructions

Various industries have been funding development of Augmented Reality systems for instruction/ maintenance.

Augmented Reality in Architectural Construction, Inspection, and Renovation

The Computer Graphics and User Interfaces Lab at Columbia University has worked on two augmented reality systems for use in structural engineering and architectural applications. The first, called "Architectural Anatomy," overlays a graphical representation of portions of the building's structural systems over a user's view of the room in which they are standing [10]. A see-through head-mounted display provides the user with monocular augmented graphics and tracks the position and orientation of their head with an ultrasonic tracking system Figure 2.3.

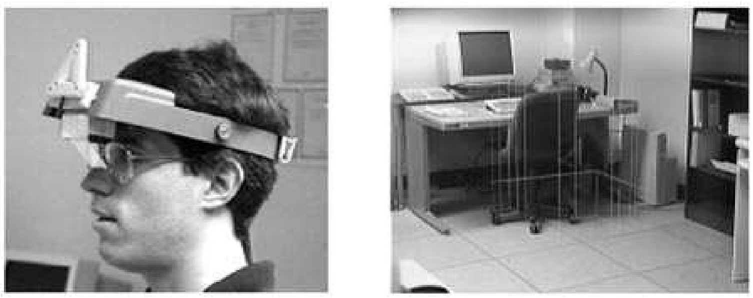

Figure 2.3: (a) The see through display (b) The structure overlapped over the real view

The above application can be used effectively in teaching architectural technology. The other augmented reality testbed system addresses spaceframe construction [11]. The spaceframe is assembled one component (strut or node) at a time. For each step of construction, the augmented reality system:

- Directs the worker to a pile of parts and tells which part to pick up by playing a sound file containing verbal instructions.
- Confirms that the worker has the correct piece by reading a barcode on the component.
- Directs the worker to install the component. A virtual image of the next piece, with a textual description fixed near it, indicates where to install the component, and verbal instructions played from a sound file explain how to install it.
- Confirms that the component is installed by asking the worker to place hand at a particular location on it (denoted by the barcode), then determining the position of the hand.

This research demonstrates the potential of augmented reality's x-ray vision and instructional guidance capabilities for improving architectural construction, inspection, and renovation.

Knowledge-based Augmented Reality for Maintenance Assistance (KARMA)

KARMA is a prototype system that a see-through head-mounted display to explain simple end-user maintenance for a laser printer. Several Logitech

3D trackers (the small triangles in the Figure 2.4 were attached to key components of the printer, allowing the system to monitor their position and orientation [12].

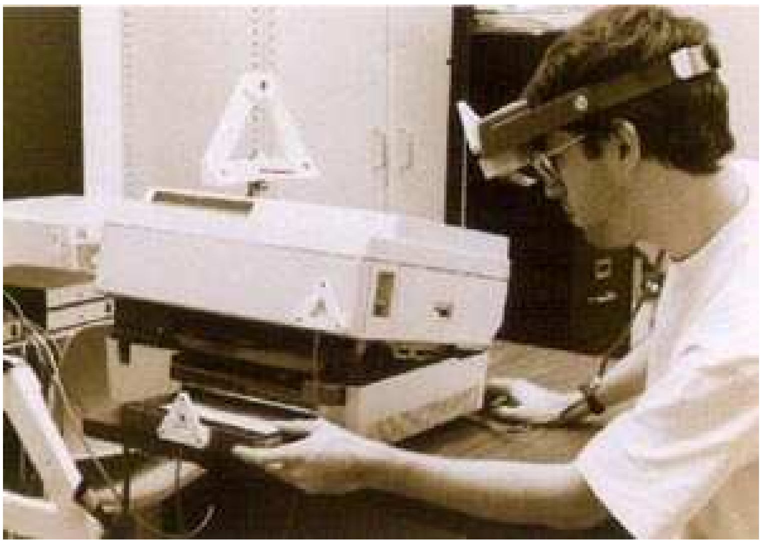

Figure 2.4: (a) Triangular trackers attached to various parts

AR in Aircraft industry

The Integrated Media Systems Center at University of Southern California is working with McDonnell Douglas engineers to create an AR system that will display text and graphics so aircraft assemblers at its Douglas Aircraft Co. facility in Long Beach can build planes more quickly and accurately. Such a system would help workers by relieving them of the need to refer back and forth to blueprints or instruction manuals.

At Boeing, David Mizell is using a grant from the Defense Advanced Research Projects Agency to try to use AR to simplify the process of bundling hundreds of wires. Traditionally, workers use foam boards with complicated pre-printed diagrams to lace the wires into a bundle. AR might allow a worker to use a blank board and rely on graphics in a head-mounted display to show where each wire should go.

2.1.5 Augmented Reality for Outdoor Applications

A Mobile Augmented Reality Systems for Exploring the Urban Environment(MARS)

This prototype system developed by the Computer Graphics and User Interfaces Lab, Columbia University acts as a campus information system, assisting a user in finding places and allowing to query information about items of interest, like buildings, statues, etc. [13]. The user carries a backpack computer with a wireless network and wears a head-mounted display. The position of the user is tracked by differential GPS while orientation data is provided by the head-mounted display itself. As the user looks around the campus, the see-through headworn display overlays textual labels on campus buildings. The user can interact with the system to bring up related information about any building.

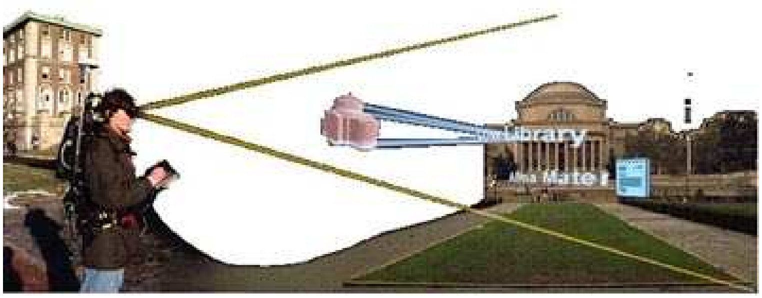

Figure 2.5: MARS - Mobile Augmented Reality System

Geospatial Registration of Information for Dismounted Soldiers (GRIDS)

GRIDS is an Augmented Reality system currently under research, that can offer an intuitive, natural way for dismounted soldiers to understand electronic information [14]. An infantryman wears a see-through Head-Mounted Display (HMD) hat which overlays computer graphics directly upon his view of the surrounding environment. The graphics are spatially registered with objects in the environment. A soldier wearing an HMD sees information labels from the tactical database directly superimposed over seen or unseen individuals and objects. The crucial requirement is that the graphic labels be properly registered to the correct objects in the environment. Accurate registration requires accurate tracking of the user's location and direction of gaze.

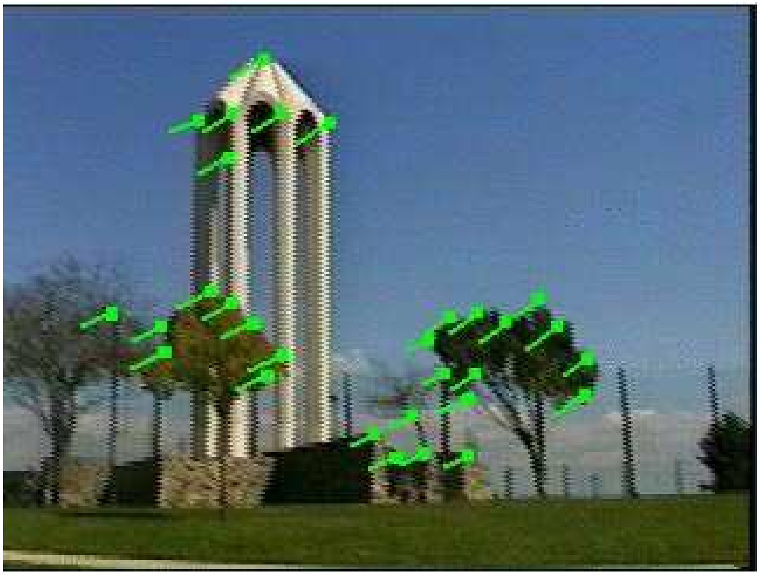

Figure 2.6: GRIDS - Showing Geospatial registration of information

2.1.6 Application for Entertainment

Augmented Reality Computing Arena for Digital Entertainment (ARCADE)

Augmented Reality Computing Arena for Digital Entertainment (ARCADE) [15] is an environment to develop and execute Augmented Reality applications targeted for digital entertainment. The computing arena is the collection of software tools, algorithms, designs, and configurations within a highly automated execution environment for Augmented Reality entertainment applications. Figure 2.7 shows the logical framework of ARCADE.

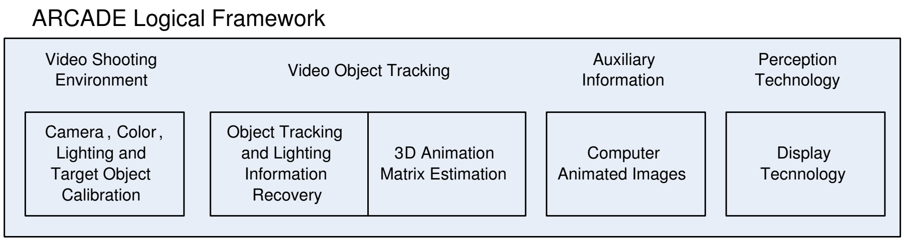

Figure 2.7: The Generic Framework of ARCADE

Since Augmented Reality entertainment applications will be executed across different hardware platforms, and some of the hardware may be proprietary, ARCADE intends to ease the hurdle of applying Augmented Reality technology for digital entertainment sectors through this generic logical framework.

ARCADE aids developers in deploying Augmented Reality entertainment experience and applying Augmented Reality technology for content creation. We deliver the architecture design of the ARCADE along with a fully tested implementation including the following components:

1 Video Object Tracking Engine (VOTE)

2 Video Object Tracking Techniques

3 Pilot Augmented Reality Entertainment Applications

2.1.7 Summary

Augmented reality is a newly research area and there are number of research project that are undergoing. There are many application that augmented reality would applied on.

2.2 Survey on Video Object Tracking

2.2.1 Introduction

Accurately tracking the user's viewing orientation and position is crucial for AR registration. Two major components can be distinguished in a typical object tracker. Target representation and localization is mostly a bottom-up process which has to cope with changes in the appearance of the target. Filtering and data association is mostly a top-down process dealing with the dynamics of the tracked object, learning of scene priors, and evaluation of different hypotheses. The way the two components are combined and weighted is application-dependent and plays a decisive role in the robustness and efficiency of the tracker.

Videos can be considered aa a sequences of images, which called a frame. The video frames are displayed in a fast enough frequency so that human eyes percept the continuity of its content. It is obvious that all image processing techniques can be applied to individual frames, including object detection and tracking. Also, the contents of two consecutive frames are usually closely related, which can be a useful information for many techniques.

Object detection in videos involves verifying the presence of an object in image sequences and possibly locating it precisely for recognition. Object tracking is to monitor an object's spatial and temporal changes during a video sequence, including its presence, position, size, shape, etc. This is done by solving the temporal correspondence problem, the problem of matching the target region in successive frames of a sequence of images taken at closely-spaced time intervals. These two processes are closely related because tracking usually starts with detecting objects, while detecting an object repeatedly in subsequent image sequence is often necessary to help and verify tracking.

2.2.2 Object tracking

Object tracking techniques can be divided into two main streams: marker-based object tracking [16], [17], [18] and marker-less-based object tracking [19]. General tracking technologies include mechanical arms and linkages: accelerometers and gyroscopes, magnetic fields, radio frequency signals, and acoustics. Tracking measurements are subject to signal noise, degradation with distance, and interference sources. Active tracking systems require calibrated sensors and signal sources in a prepared and calibrated environment. Among passive tracking approaches, computer vision methods can determine pose as well as detect, measure, and reduce pose tracking errors derived by other technologies. The combined abilities to both track pose and manage residual errors are unique to vision-based approaches.

Several different approaches for object tracking exist. Comaniciu [20] proposed a new approach toward target representation and localization of the central component in visual tracking of nonrigid objects. The feature histogram-based target representations are regularized by spatial masking with an isotropic kernel. The masking induces spatially-smooth similarity functions suitable for gradient-based optimization; hence, the target localization problem can be formulated using the basin of attraction of the local maxima.

Oron [21] proposed an algorithm for multi-object tracking method for video images. This algorithm is based on motion estimation and image difference techniques. It provides an improved capability for object tracking, especially for cases of non-stationary background conditions due to radiometric changes, camera motion instability and motion of different objects within the images.

Natural feature tracking is another challenging topic. Natural feature tracking algorithms detect and track target objects in the scene automatically without any pre-defined figures or objects. In [22], a natural feature tracking algorithm is proposed. Natural scene features stabilize and extend the tracking range of augmented reality pose-tracking systems. Point and region features are automatically and adaptively selected for properties that lead to robust tracking.

2.2.3 Current Techniques on Object Detection and Tracking

Template-based object detection

Template-based object detection is used when a template describing a specific object is available. In such case, object detection becomes a process of matching features between the template and the image sequence under analysis. Object detection with an exact match is computationally expensive and the quality of matching depends on the details and the degree of precision provided by the object template. In general, there are two types of object template matching, fixed and deformable template matching.[37]

Fixed template matching Fixed templates are useful when object shapes do not change with respect to the viewing angle of the camera. Two major techniques have been used in fix template matching, image subtraction and correlation.

In Image subtraction, the template position is determined from minimizing the distance function between the template and various positions in the image. Although image subtraction techniques require less computation time than the following correlation techniques, they perform well in restricted environments where imaging conditions, such as image intensity and viewing angles between the template and images containing this template are the same.

For correlation, the position of the normalized cross-correlation peak between a template and an image to locate the best match is used. This technique is insensitive to noise and illumination effects in the images, but suffers from high computational complexity caused by summations over the entire template. Point correlation can reduce the computational complexity to a small set of carefully chosen points for the summations. [39]

Deformable template matching Deformable template matching approaches are more suitable for cases where objects vary due to rigid and non-rigid deformations. These variations can be caused by either the deformation of the object per se or just by different object pose relative to the camera. Because of the deformable nature of objects in most video, deformable models are more appealing in tracking tasks.

In this approach, a template is represented as a bitmap describing the characteristic contour/edges of an object shape. A probabilistic transformation on the prototype contour is applied to deform the template to fit salient edges in the input image. An objective function with transformation parameters which alter the shape of the template is formulated reflecting the cost of such transformations. The objective function is minimized by iteratively updating the transformation parameters to best match the object [37]. The most important application of deformable template matching techniques is motion detection of objects in video frames which we will review in the following section. [51, 53]

Feature-based object detection

In feature-based object detection, standardization of image features and registration (alignment) of reference points are important. The images may need to be transformed to another space for handling changes in illumination, size and orientation. One or more features are extracted and the objects of interest are modeled in terms of these features. Object detection and recognition then can be transformed into a graph matching problem.

Shape-based approaches Shape-based object detection is one of the hardest problems due to the difficulty of segmenting objects of interest in the images. In order to detect and determine the border of an object, an image may need to be preprocessed. The preprocessing algorithm or filter depends on the application. Different object types such as persons, flowers, and airplanes may require different algorithms. For more complex scenes, noise removal and transformations invariant to scale and rotation may be needed.

Once the object is detected and located, its boundary can be found by edge detection and boundary-following algorithms. The detection and shape characterization of the objects becomes more difficult for complex scenes where there are many objects with occlusions and shading. [31, 48]

Color-based approaches Unlike many other image features (e.g. shape) color is relatively constant under viewpoint changes and it is easy to be acquired. Although color is not always appropriate as the sole means of detecting and tracking objects, but the low computational cost of the algorithms proposed makes color a desirable feature to exploit when appropriate. [52]

[33] developed an algorithm to detect and track vehicles or pedestrians in real-time using color histogram based technique. They created a Gaussian Mixture Model to describe the color distribution within the sequence of images and to segment the image into background and objects. Object occlusion was handled using an occlusion buffer. [32] achieved tracking multiple faces in real time at full frame size and rate using color cues. This simple tracking method is based on tracking regions of similar normalized color from frame to frame. These regions are defined within the extent of the object to be tracked with fixed size and relative positions. Each region is characterized by a color vector computed by sub-sampling the pixels within the region, which represents the averaged color of pixels within this region. They even achieved some degree of robustness to occlusion by explicitly modeling the occlusion process.

Motion detection

Detecting moving objects, or motion detection, obviously has very important significance in video object detection and tracking. A large proportion of research efforts of object detection and tracking focused on this problem in last decade. Compared with object detection without motion, on one hand, motion detection complicates the object detection problem by adding object's temporal change requirements, on the other hand, it also provides another information source for detection and tracking.

A large variety of motion detection algorithms have been proposed. They can be classified into the following groups approximately.

Thresholding technique over the interframe difference These approaches [30] rely on the detection of temporal changes either at pixel or block level. The difference map is usually binarized using a predefined threshold value to obtain the motion/no-motion classification.

Statistical tests constrained to pixelwise independent decisions These tests assume intrinsically that the detection of temporal changes is equivalent to the motion detection [44]. However, this assumption is valid when either large displacement appear or the object projections are sufficiently textured, but fails in the case of moving objects that preserve uniform regions. To avoid this limitation, temporal change detection masks and filters have also been considered. The use of these masks improves the efficiency of the change detection algorithms, especially in the case where some a priori knowledge about the size of the moving objects is available, since it can be used to determine the type and the size of the masks. On the other hand, these masks have limited applicability since they cannot provide an invariant change detection model (with respect to size, illumination) and cannot be used without an a priori context-based knowledge.

Global energy frameworks The motion detection problem is formulated to minimize a global objective function and is usually performed using stochastic (Mean-field, Simulated Annealing) or deterministic relaxation algorithms (Iterated Conditional Modes, Highest Confidence First). In that direction, the spatial Markov Random Fields [49] have been widely used and motion detection has been considered as a statistical estimation problem. Although this estimation is a very powerful, usually it is very time consuming.

2.2.4 Object tracking using motion information

Motion detection provides useful information for object tracking. Tracking requires extra segmentation of the corresponding motion parameters. There are numerous research efforts dealing with the tracking problem. Existing approaches can be mainly classified into two categories: motion-based and model-based approaches [47]. Motion-based approaches rely on robust methods for grouping visual motion consistencies over time. These methods are relatively fast but have considerable difficulties in dealing with non-rigid movements and objects. Model-based approaches also explore the usage of high-level semantics and knowledge of the objects. These methods are more reliable compared to the motion-based ones, but they suffer from high computational costs for complex models due to the need for coping with scaling, translation, rotation, and deformation of the objects.

Tracking is performed through analyze geometrical or region-based properties of the tracked object. Depending on the information source, existing approaches can be classified into boundary-based and region-based approaches.

Boundary-based approaches

Also referred to as edge-based, this type of approaches rely on the information provided by the object boundaries. It has been widely adopted in object tracking because the boundary-based features (edges) provide reliable information which does not depend on the motion type, or object shape. Usually, the boundary-based tracking algorithms employ active contour models, like snakes [45] and geodesic active contours. These models are energy-based or geometric-based minimization approaches that evolve an initial curve under the influence of external potentials, while it is being constrained by internal energies.

Snakes Snakes is a deformable active contours used for boundary tracking which was originally introduced by Terzopoulos et al [40]. In segmentation and boundary tracking problems, these forces relate to the gradient of image intensity and the positions of image features. One advantage of the force-driven snake model is that it can easily incorporate the dynamics derived from time-varying images. The snakes are usually parameterized and the solution space is constrained to have a predefined shape. So these methods require an accurate initialization step since the initial contour converges iteratively toward the solution of a partial differential equation [25, 41, 50].

Considerable work has been done to overcome the numerical problems associated with the solution of the equations of motion and to improve robustness to image clutter and occlusions. [27] proposed a B-spline representation of active contours, [29] employed polygonal representation in vehicle tracking problems, and [43] proposed a deformable superquadric model for modeling of shape and motion of 3D non-rigid objects.

Geodesic active contour models These models are not parameterized and can be used to track objects that undergo non-rigid motion. In [28], a three step approach is proposed which start by detecting the contours of the objects to be tracked. An estimation of the velocity vector field along the detected contours is then performed. At this step, very unstable measurements can be obtained. Following this, a partial differential equation is designed to move the contours to the boundary of the moving objects. These contours are then used as initial estimates of the contours in the next image and the process iterates. More recently, in [26], a front propagation approach that couples two partial differential equations to deal with the problems of object tracking and sequential segmentation was proposed. Additionally, in [34], a new, efficient numerical implementation of the geodesic active contour model has been proposed which was applied to track objects in movies.

Region-based approaches

These approaches rely on information provided by the entire region such as texture and motion-based properties using a motion estimation/segmentation technique. In this case, the estimation of the target's velocity is based on the correspondence between the associated target regions at different time instants. This operation is usually time consuming (a point-to-point correspondence is required within the whole region) and is accelerated by the use of parametric motion models that describe the target motion with a small set of parameters. The use of these models introduces the difficulty of tracking the real object boundaries in cases with non-rigid movements/objects, but increases robustness due to the fact that information provided by the whole region is exploited.

Optical flow [42, 46] is one of the widely used methods in this category. In this method, the apparent velocity and direction of every pixel in the frame have to be computed. It is an effective method but time consuming. Background motion model can be calculated using optic flow, which serves to stabilize the image of the background plane. Then, independent motion is detected as either residual flow, the flow in the direction of the image gradient that is not predicted by the background plane motion. Although slightly more costly to compute, this measure has a more direct geometric significance than using background subtraction on a stabilized image. This method is very attractive in detecting and tracking objects in video with moving background or shot by a moving camera.

2.2.5 Challenges

Although has been studied for dozens of years, object detection and tracking remains an open research problem. A robust, accurate and high performance approach is still a great challenge today. The difficulty level of this problem highly depends on how you define the object to be detected and tracked. If only a few visual features, such as a specific color, are used as representation of an object, it is fairly easy to identify all pixels with same color as the object. On the other extremity, the face of a specific person, which full of perceptual details and interfering information such as different poses and illumination, is very hard to be accurately detected, recognized and tracked. Most challenges arise from the image variability of video because video objects generally are moving objects. As an object moves through the field of view of a camera, the images of the object may change dramatically. This variability comes from three principle sources: variation in target pose or target deformations, variation in illumination, and partial or full occlusion of the target.

There are two sources of information in video that can be used to detect and track objects: visual features (such as color, texture and shape) and motion information. Combination of statistical analysis of visual features and temporal motion information usually lead to more robust approaches. A typical strategy may segment a frame into regions based on color and texture information first, and then merge regions with similar motion vectors subject to certain constraints such as adjacency. A large number of approaches have been proposed in literature. All these efforts focus on several different research areas each deals with one aspect of the object detection and tracking problems or a specific scenario. Most of them use multiple techniques and there are combinations and intersections among different methods. All these make it very difficult to have a uniform classification of existing approaches. So in the following sections, we would review most of the approaches separately in association with different research highlights.

2.2.6 Summary

Augmented reality becomes more and more popular and the success of the augmented reality is highly depends on the accuracy of the object tracking result. Although so much work has been done in object tracking, it still seems impossible so far to have a generalized, robust, accurate and real-time approach that will apply to all scenarios. There are number of difficulties, including: noisy background, moving camera or observer, bad shooting conditions, object occlusions, etc. Thus, object tracking is still a open research area. As the computing power keeps increasing and network keeps developing, more complex problem may become solvable.

2.3 Survey on Active Appearance Model

2.3.1 Introduction

The Active Appearance Model (AAM) algorithm is a powerful tool for modelling images of deformable objects. AAM combines a subspace-based deformable model of an object's appearance with a fast and robust method of fitting this model to a previously unseen image.

The Active Appearance Model (AAM) algorithm, since its introduction by Edwards et al. [74, 57], has found many applications in a variety of areas such as face tracking, face recognition, and medical imaging [79]. AAM uses Principal Component Analysis (PCA) based linear subspaces to model the 2D shapes and textures of the images of a target object class. Such a representation allows AAM to represent a certain image with a very small number of parameters. Given a previously unseen image that belongs to the same object class, AAM finds the optimal parameters to represent the target image by using an iterative scheme that is fast and robust. The speed of this algorithm comes from the assumption that the gradient matrix is fixed around the optimal coefficients for all images. AAM numerically estimates this fixed gradient matrix by estimating it for a set of training images around their optimal parameters and averaging the results. Different variations of the basic AAM algorithm have been proposed [58, 84, 89, 90]

The Active Appearance Model (AAM) algorithm has proved to be a successful method for matching statistical models of appearance to new images.

There has been a great deal of research into using deformable models to interpret images. Reviews are given in [57]. Active Shape Models were developed by Cootes et.al.[57] to match statistical models of object shape to new images. They have been used successfully in many application areas, including face recognition, industrial inspection [57] and medical image interpretation. They have been extended to search 3D images. Active Appearance Models were introduced more recently. They have proved very successful for interpreting and tracking images of faces, and have been applied to medical image interpretation. They been extended to model and search colour images.

The time line of development in Active Shape Models(ASMs) and Active Appearance Models(AAMs) is shown in

2.3.2 Basic Active Appearance Models

Active Appearance Models were developed by Gareth Edwards et al. in 1998 and ever since been a valuable extension to the extensively used Active Shape Models. This was a proposal and implementation of a statistical entity capable of capturing full appearance of some object – an appearance that can be faithfully described by the generic object shape mapped with some overlaid textures. Such models expressed not only the variation of shape, but also pixel intensities that are vital for full reconstruction and synthesis of valid realistic model instances.

Shape Model

The creation of such a model firstly relies on landmarking, much as in the case of shape models. Annotation of edges, corners and T-junctions in the

Table 2.1: Timeline for development of AAMs and ASMs

Year	Shape Modelling	Appearance Modelling
1991	First parametric statistical shape model, using PCA on inter-point distances Cootes et. al.[55, 56]	Craw and Cameron warp faces to a reference shape before doing PCA [73]
1992	Point Distribution Model [60] Active Shape Model, search using edge detectors [59]	
1993	Statistical profile models for ASM [61, 70] ASM with varying weights at each point [61, 70] 3D ASM [79]	
1994	Combining statistical and elastic shape deformation [62, 64] Multi-resolution ASMs [69] Non-linear ASM using Polynomial Regression [85] B-spline based ASMs [54]	Statistical models of shape and texture [63] Statistical Appearance Models [80] Shape model with statistical model of concatenated profiles [77]
1995	Optimal estimation of shape and pose fit to points [78]	
1996		Statistical models of (x,y,I) surface [82]
1997	Mixture models of shape distributions in ASM [65, 66]	
1998		Active Appearance Model introduced [74, 57] Shape-AAM (shape driven by residuals) [58]
1999	Non-linear ASMs using Kernel PCA [83] 2D+time ASMs [76, 75]	Wavelet compression of AAMs [87]
2000		View-based AAMs [71] Local refinement of AAM search [67] Coupled models of appearance [72]
2001		Constrained AAMs [68] Inverse Compositional updating scheme [31]
2002	Use of classifiers to detect features [86] Robust estimation of shape parameters [81]	
2004		Real time combined 2D+3D active appearance model [88]

image identifies unique attributes in some image that can be consistently located across a whole set of examples. Furthermore, what in fact classified such models as ones of full appearance is their ability to extract intensity (brightness) values from a shape normalised to fit a global mean (Procrustes analysis is used here to apply translation, rotation and scaling). Although originally the technique was implemented in greyscale images, Stegmann et. al. now provide an open source API that supports a corresponding RGB appearance for face datasets. Not surprisingly, it usually required 3 times the amount of space and time to process and any solution for grayscale data usually extends to colour by breaking up pixel elements into 3 components.

The 2D shape of an AAM is defined by a 2D triangulate mesh and in particular the vertex locations of the mesh. Mathematically, we define the shape $\boldsymbol{s}$ of an AAM as the 2D coordinates of the n vertices that make up the mesh:

$$\mathbf{s} = \begin{pmatrix} u_1 & u_2 & \dots & u_n \\ v_1 & v_2 & \dots & v_n \end{pmatrix} \tag{2.1}$$

AAMs allow linear shape variation. This means that the shape matrix $\boldsymbol{s}$ can be expressed as a base shape s_0 plus a linear combination of m shape matrices s_i

$$s = s_0 + \sum_{i=1}^{m} p_i s_i \tag{2.2}$$

where the coefficients p_i are the shape parameters.

AAMs are normally computed from training data consisting of a set of images with the shape mesh (usually had) marked on them. Principal component Analysis (PCA) is then applied to the training meshes. The base shape s_0 is the mean shape and the matrices s_i are the (reshaped) eigenvectors corresponding to the m largest engienvalues. An example independent AAM shape model is shown in Figure 2.8. The triangulated base mesh s_0 is plotted in the left of the Figure 2.8. In the remainder of the figure, the base mesh $s0$ is overlayed with arrows corresponding to each of the first four shape vectors s_1, s_2, s_3 and s_4.

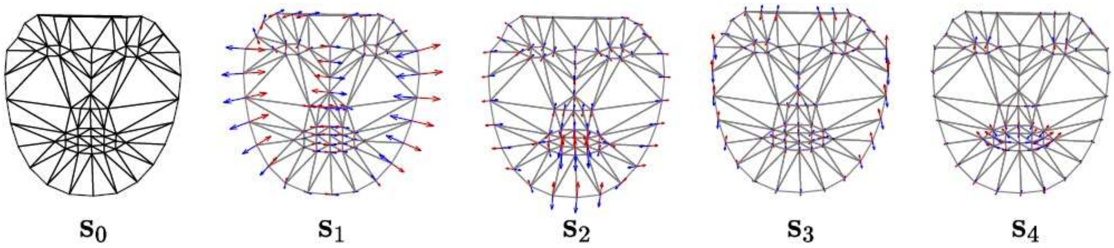

Figure 2.8: The linear shape model of an independent AAM. The model consists of a triangulated base mesh s_0 plus a linear combination of shape vectors s_is .

s_0 is the mean shape and the matrices s_i are the eigenvectors corresponding to the m largest eigenvalues. The resulted shape is the linear combination of the base mesh and other shape vector. Example is shown in Figure 2.9.

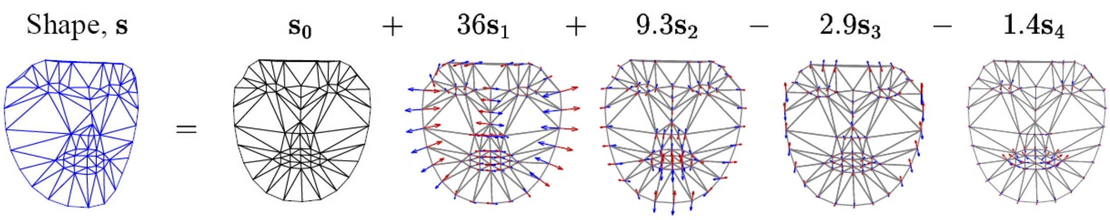

Figure 2.9: Example of the combination of the 2D AAMs shape models

Appearance Model

The appearance models encompass both shape and texture which are coded in a single vector. The means of finding correlation between the two is eigenanalysis of the covariance matrix where Principal Component analysis gives encouraging results and can reduce the dimensionality of the data considerably well while still accounting for much of the variation (but not all of it of course). There is a significant improportion between the space required (hence speed) and the loss that Principal Component analysis imposes. What is seen in practice that the components in the analysis quickly shrink, that is, they have a very small discriminatory power and when values become almost negligible they can be discarded. That, of course, will depend on the requirments of the system. For industrial inspection where quality is crucial or in medical image analysis, low error rates are usually required and the presence of abnormality is difficult to spot. On the contrary, if real-time object tracing in a video sequence is required, subsequent framed can compensate for incorrect location and efficiency is at a premium.

The appearance of the AAM is defined within the base mesh s_0. Let s_0 also denote the set of pixels $\mathbf{u} = (u, v)^T$ that lie inside the base mesh s_0, a convenient abuse of terminology. The appearance of the AAM is then an image $\mathbf{A(u)}$ defined over the pixels $\mathbf{u} \in s_0$. AAMs allows linear appearance variation. This means that the appearance $A(u)$ can be expressed as a base appearance $A_0(u)$ plus a linear combination of l appearance images $A_i(u)$:

$$A(u) = A_0(u) + \sum_{i=1}^{l} \lambda_i A_i(u) \tag{2.3}$$

where the coefficients λ_i are the appearance parameters. Since we can easily perform a linear reparameterization, wherever necessary we assume that the images A_i are orthonormal. As with the shape, the base appearance A_0 and appearance images A_i are usually computed by applying PCA to the shape normalized training images.

The base appearance is set to be the mean image and the images A_i to be the m eigenimages corresponding to the m largest eigenvalues. The appearance of an example independent AAM is shown in Figure 2.10.The appearance of an example independent AAM is shown in Figure 2.10. The base appearance A_0 is plotted on the left of the Figure 2.10. On the right, the first four of the appearance images is plotted, A_1, A_2, A_3 and A_4. And the sample of the linear combination of the appearance of the AAM is shown in Figure 2.11.

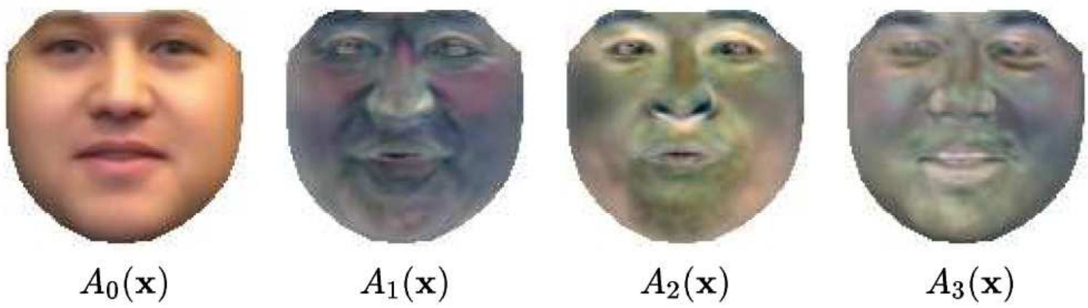

Figure 2.10: The linear appearance variation of an independent AAM. The model consists of a base appearance image A_0 and A_i s

As with the shape component, the base appearance A_0 and the appearance images A_i are normally computed by applying PCA to a set of shape normalised training images. Each training image is shape normalised by

Figure 2.11: Example of appearance models in the AAM

warping the (hand labelled) training mesh onto the base mesh s_0 Usually the mesh is triangulated and a piecewise affine warp is defined between corresponding triangles in the training and base meshes [11] (although there are ways to avoid triangulating the mesh using, for example, thin plate splines rather than piecewise affine warping [10].) The base appearance is set to be the mean image and the images A_i to be the m eigenimages corresponding to the m largest eigenvalues. The fact that the training images are shape normalised before PCA is applied normally results in a far more compact appearance eigenspace than would otherwise be obtained.

Principle Component Analysis

Throughout the process of PCA, dimensionality reduction is initially performed to make the shape representation more compact, but secondly to reduce the dimensionality of the vector describing texture variation (with the mean shape available for normalisation) in the observed (training) data.

To obtain a model that accounts for both the above variations, namely shape and intensity, Principal Component Analysis is again used to reduce the dimensionality of the aggregation of the two. During this process, the correlation between both of these is learned and a combined vector is formed. To account for different representation of texture and shape, i.e. axis aligned, normalised and centred coordinates versus 8 bit (24 bit for RGB by most conventions) encoding of colour, a matrix that scales both components by some given weighing is used. The matrix elements of this weighing component, W, define some type of transformer that improves consistency of value range in the currently handled solumn vector. As a result of the process, a vector which is rather compact can be obtained which describes full appearance (shape and texture). Typically it is larger in size than its two original merged components. That is made mandatory in order to account for the same amount of variation as before. It is rather obvious though that any level of model fidelity can be chosen and it has a direct connection with the number of elements it comprises.

A linear PCA is used to recursively find the direction in which the variation of some data is maximal. Sometimes (for a manageable number of dimensions) we can visualise all vectorised data in space so that an imaginary cloud of points is formed. PCA is able to identify the component whose removal would be the most harmful to classification of that data, i.e. the direction that distinguishes different data instances most effectively. The eigenvalues corresponding to the data in hand indicate how significant each eignevector is with respect to data discrimination. Hence, not all existing eigenvectors (which are linearly dependent on the data dimensionality) are equally useful in some new, more succinct vector representation. Some of them can be found to be 0 in which case they can be fully ignored and dimesionality reduction that is not lossy becomes available.

The allowed range of values for each parameter in the resulting appearance model indicates a general variability property. The modes of variation, that is, the collection of n modes is sorted in descending order of influence on the overall appearance. Mode n will in fact be the n^{th} vector element. The variation of the model and the allowed range is restricted by a set of parameters (virtually a column vector) bi that can describe a legal state of the model when considered collectively.

Model Instantiation

To traverse image structures, the models produced are usually stretched to fit an image under a standard optimisation routine where image differences (pairwise difference) is first and foremost taken into account. It is not the most attractive feature of this novel technique, but uses of this ability begin to emerge. Interpretation of gestures through the variables b and motion tracking are among the more interesting directions that fitting model to an image took. Measure, inspection and diagnosis are some of the more useful directions.

A somewhat detailed and irrelevant aspect of AAM search is to do with optimizations, off-line training and speed-up. To allow quick and reliable convergence between a model and an image, the relationship between parameter values and the effects they have on the error measure (inferred from image differences) is learned before searching takes place. Not only parameters are taken into account, but also rigid transformations that are vital for matching, let us say, if we know very little about the size of a target object in an image.

To achieve the above a long sequence of alterations to the models is applied and the effects on intensities is learned and recorded in some matrix A. A collection of matrices eventually guides the steps taken in each iteration to achieve better conversion. This matrices can be thought of as masks which allow any real number and the matrices form a virtual image overlay.

Good initialization is normally required when it comes to the placement of a model in some image. The search will inspect nearby pixels more than distant ones and if nearby pixels show little potential (for fitting), if any at all, then the algorithm will converge in some local minima (or run forever, or maximum number of iterations will be reached). To allow for robust performance, different resolutions of the image as well as scaled models of appearance can be used. Gaussian averaging is normally used to produce such analogous simpler (coarser) elements of the original data. The assumption is that given a coarse scale the problem is simplified and something can be learned and passed forward to the later iterations that deal with finer image resolutions.

Equations 2.2 and 2.3 describe the AAM shape and appearance variation. However, they do not describe how to generate a model instance. Given the AAM shape parameters $\mathbf{p} = (p_i, p_2, ..., p_i)^T$ we can use Equation 2.2 to generate the shape of the AAM $\mathbf{s}$. Similarly, given the AAM appearance parameters $\lambda = (\lambda_1, \lambda_2, ..., \lambda_m)^T$. we can generate the AAM appearance $A(x)$ defined in the interior of the base mesh $\mathbf{s}_0$. The AAM model instance with shape parameters $\mathbf{p}$ and appearance parameters λ is then created by warping the appearance A from the base mesh $\mathbf{s}_0$ to the model shape $\mathbf{s}$. This process is illustrated in Figure 2.12 for concrete values of $\mathbf{p}$ and λ.

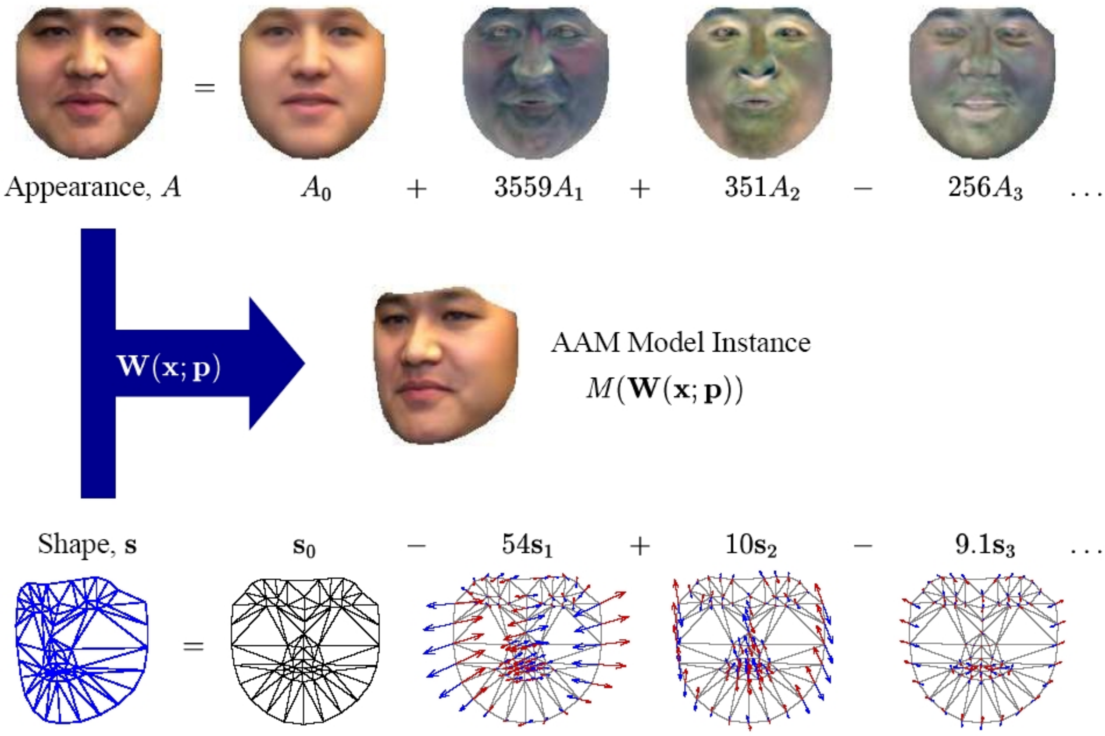

Figure 2.12: Sample of linear combination of appearance of the AAM

Fitting Algorithm

For fitting the AMMs, there are number of proposed algorithms. Here we describe the popular ones and discuss their advantages and disadvantages.

Basic Search The basic search of the AAMs if using the following algorithm. This is repeated utill no improvement is made to the error, $|\Delta I|^2$

Sub-sampling During Search During the search, all the points in the model to obtain I_s is sampled with the prediction on the change to the model parameters. The change in the i^{th} parameter, Δc_i, is given by

Algorithm 1 Iterative Search for Fitting Active Appearance Model

Require: m: number of sample

Evaluate the error vector $\Delta I_0 = I_m$
Evaluate the current error $E_0 = |\Delta I_0|^2$
Compute the predicted displacement, $\Delta c = R\Delta I_0$
Set $k = 1$
Let $c_1 = k\Delta c$
Sample the image at this new prediction, and calculate a new error vector, ΔI_1
if $|\Delta I_1| \leq E_0$ **then**
    Accept the new estimate, $c1$
else
    try at $k = 0.5$, $k = 0.25$ etc.
end if

$$\Delta c_i = A_i \Delta I \tag{2.4}$$

where A_i if the i^{th} row of A.

To select a useful subset for all parameters, the best $u\%$ of elements for each parameter are computed, then the union of such sets are generated. If u is small enough, the union will be less than all the elements. Then, a new multi-variate regression is preformed to compute the relationship. A' between the changes in the subset of samples, $\Delta I'$, and the changes in parameters

$$\Delta c = A' \Delta I' \tag{2.5}$$

Search Using Shape Parameters An alternative approach is to use image residuals to drive the shape parameters, p_s, and thus c directly from the image given the current shape, This approach ay be useful when there are few shape modes.

The update equations in this case has the form

$$\Delta p_s = P \Delta I \tag{2.6}$$

where in this case ΔI is given by the different between the current image sample g_s and the best fit of the grey-level model to it, g_m. The test for convergence is by monitoring changes in the shape parameters, or simply apply a fixed number of iterations at each resolution.

2.3.3 Extension of AAMs

Extensions to appearance models span a large range of applications and purposes. Some work of Cootes et al. extended the application of AAM's to faces so that one can switch between different models depending on the view point. Each of these models requires a separate training and learning process as well as relevant data that can be hard to collect. This work intended to allow greater flexibility as the head moves and rotates. This may be of some interest if access control systems exploit AAM's and an almost strictly orthogonal view on a face is difficult to acquire. The normal assumptions of the model usually break when some landmarks get occluded. According to Lanitis et al. this happens at when the angle that entends between the frontal view and the aperture location extends over 22.5 degrees. More work earlier on took place to account for 3-D data and slicing was a common requirement as in the case for brain model (atlas) fitting. Current work attempts to automate much of the process, annotation being a particular problem. When this problem is solved, human intervention will become minimal and the cost of model acquisition will go down considerably.

Chapter 3

Object Tracking with Active Appearance Model and Kalman Filter

3.1 Introduction

Object tracking is an important component to augmented reality. In augmented reality system, a registration is needed such that the objects in the video is being detected and tracked. Then, the virtual object can be placed onto the video. In object tracking system, a model is need to be applied to estimate the shape and the motion of the objects. The Active Appearance Model (AAM) algorithm has proved to be a successful method for matching statistical models of appearance to new images. Thus, in our newly propose scheme, we applied the AAM in the object tracking system in the modelling part. Also, 3D-constraints are added to the fitting part of the AAM fitting algorithm to increase the speed of the tracking system.

In the following sections, the details of the proposed object tracking system is described.

3.2 Scheme details

We propose a object tracking scheme that the active appearance model is applied. The system includes the following parts: object matching, motion modelling, and occlusion detection. The main contribution is propose a scheme to apply the active appearance model for tracking purpose. The overview of the system is shown in Figure 3.1.

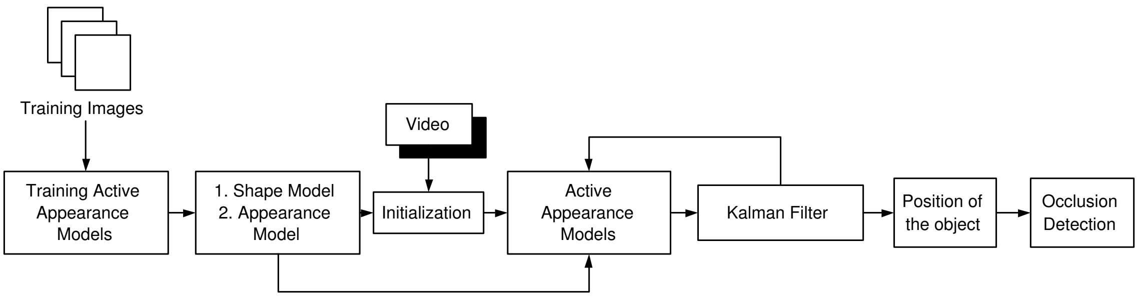

Figure 3.1: Overview of the tracking system

Active appearance model is applied for object matching parts. When the video is input to the system, the position of the object estimated by the AAM. The fitting speed of the AAM algorithm is inherently dependent on good initialization. A multi-scale AAM initialization is applied. Then motion is modelled by Kalamn filter. Finally, the occlusion in the system is detected.

3.2.1 Object Matching by Active Appearance Model

A low cost active appearance model is applied in this object matching. To match a object in a video, a training set of number of images are needed to train the shape model and the appearance model in the AAM. The following sections describe the AAM applied and the initialization of the AAM.

The Active Appearance Model (AAM) is a generalization of the widely used Active Shape Model approach, but uses all the information in the image region covered by the target object, rather than just that near modelled edges.

Shape Formation

AAMs handle planar shape as a finite set of landmarks, i.e. corresponding points between and within populations. The representation used for a single n-point shape is defined as Equation 3.1

$$\mathbf{s} = \begin{pmatrix} u_1 & u_2 & \dots & u_n \\ v_1 & v_2 & \dots & v_n \end{pmatrix} \tag{3.1}$$

For dealing with redundancy in multivariate data, AAMs utilise the linear orthogonal transformation; principal component analysis (PCA). In this setting a shape of n points is thus considered one observation, x_i, in a $2n$ dimensional space.

The shape PCA is essentially an eigen-analysis of the covariance matrix of the shapes aligned w.r.t. position, scale and rotation, i.e. after a Procrustes analysis.

New shape instances can thus be synthesised by deforming the mean shape, s_0, using a linear combination s_1, s_2..., and s_i with shape parameter p_i:

$$s = s_0 + \sum_{i=1}^{m} p_i s_i \tag{3.2}$$

Texture Formulation

Contrary to the prevalent understanding of the term texture in the computer vision community, this concept will be used somewhat differently below. Here we define texture as "The pixel intensities across the object in question (if necessary after a suitable normalisation)". For l samples over the object surface, the texture is represented as:

$$A = [A_0, A_2, ..., A_l]^T \tag{3.3}$$

In the shape case, the data acquisition is straightforward because the landmarks in the shape vector constitute the data itself. In the texture-case one needs a consistent method for collecting the texture information between the landmarks, i.e. an image sampling function needs to be established. This can be done in several ways. Here, a piece-wise affine warp based on the Delaunay triangulation of the mean shape is applied.

Following the warp from an actual shape to the mean shape, a normalization of the g-vector set is performed to avoid the influence from global linear changes in pixel intensities. Hereafter, the analysis is identical to that of the shapes. Hence, a compact representation is derived to deform the texture in a manner similar to what is observed in the training set:

$$A(u) = A_0(u) + \sum_{i=1}^{l} \lambda_i A_i(u) \tag{3.4}$$

where the coefficients λ_i are the appearance parameters.

Independent AAMs have separate shape p and appearance λ parameters. On the other hand, combined AAMs just use a single set of parameters $c = (c_1, c_2, ..., c_l)^T$ to parameterized shape:

$$s = s_0 + \sum_{i=1}^{l} c_i s_i \tag{3.5}$$

and appearance:

$$A(u) = A_0(u) + \sum_{i=1}^{l} c_i A_i(u) \tag{3.6}$$

Optimization

In AAMs, the search is treated as an optimization problem in which the difference between the synthesized object delivered by the AAM and an actual image is to be minimized. By adjusting the AAM-parameters (c and pose) the model texture, I_{model}, can be deformed to fit the image, I_{image}, in the best possible way. In this case the quadratic error norm is applied as optimization criterion.

$$E = \sum_{i=1}^{m} (I_{model} - I_{image})^2 = \sum_{i=1}^{m} (\Delta A_i)^2 = \|\Delta A\|^2 \tag{3.7}$$

Though the parameterisation of the object class in question can be compressed markedly by the principal component analysis it is far from an easy task to optimize the system. This is not only computationally cumbersome but also theoretically challenging since it is most likely non-convex. AAMs handle these potential problems by assuming a linear relationship between parameter changes, Δc, and pixel differences, ΔI.

$$\Delta c = R\Delta I \tag{3.8}$$

We can use 3.8 in an iterative search algorithm. Given the current estimate of model parameters, c_0, and the normalised image sample at the current estimate, I_s, each iteration proceeds as follows:

Algorithm 2 Iterative Search for Fitting Active Appearance Model

Require: m: number of sample

1: Evaluate the error vector $\Delta I_0 = I_s - I_m$
2: Evaluate the current error $E_0 = |\Delta I_0|^2$
3: Compute the predicted displacement, $\Delta c = R\Delta I_0$
4: Set $k = 1$
5: Let $c_1 = k\Delta c$
6: Sample the image at this new prediction, and calculate a new error vector, ΔI_1
7: **if** $|\Delta I_1| \leq E_0$ **then**
8: Accept the new estimate, $c1$
9: **else**
10: try at $k = 0.5$, $k = 0.25$ etc.
11: **end if**

This is repeated until no improvement is made to the error, $|\Delta I|^2$, and convergence is declared.

Initialization

The basic AAM optimization scheme is inherently dependent on good initialization. To accommodate this, we devise the following search-based scheme thus rendering the use of AAMs fully automated. The technique is inspired by the work of Cootes et al. [57] who use a pixel difference evaluation criteria and a threshold estimation for detecting multiple object instances.

The fact that AAMs are self-contained is exploited in the initialization. They can fully synthesize photorealistic objects of the class that they represent concerning shape and textural appearance. Hence, we use the model without any additional data to perform the initialization.

The idea is to exploit an inherent property of the AAM optimization V i.e. convergence within some range from the optimum. This is utilized to narrow down an exhaustive search from a dense to a sparse population of the hyperspace spanned by pose- and c-parameters. In other words, normal AAM optimizations are performed sparsely over the image using perturbations of the pose and model parameters.

This has proven to be feasible, fast and robust. A set of relevant search configuration ranges is established and the sampling within this set is done as sparsely as possible. Further, this is done in scale-pyramid to increase speed. Any available prior knowledge about pose is utilized when determining search ranges

In pseudocode, the initialization scheme for detecting one object per image is:

Algorithm 3 Multi-scale AAM Initialization

Require: m: max number of transient iterations

Require: k: max number of final iterations ($k > m$)

Require: n: number of initialization candidates

Require: $\{\Omega\}$: An empty candidate set with room for n result entries, $\{\omega_i\}_{i=1}^{n} = \{\theta_i, E_i\}_{i=1}^{n}$

Require: $\{\Psi\}$: A set of application specific search ranges for each model parameter

Populate the space spanned by $\{\Psi\}$ as as sparsely as the linear regression allows by a set of search configurations $\Theta = \{\theta_1, ..., \theta_n\}$
for each vector in Θ **do**
  Run AAM optimization at θ_i (max$m$ iterations)
  Calculate the fir, $E \leftarrow \|\Delta I\|^2$
  $\omega_{max} \leftarrow max_E\{\Omega\}$
  if $E < E_{max}$ **then**
    if the number of elements in $\{\Omega\} == n$ **then**
      remove $max_E\{\Omega\}$
    end if
    add $(\theta_i, E) to \{\Omega\}$
  end if
end for
for each element in $\{\Omega\}$ **do**
  Run AAM optimization at θ_i (max k iterations)
  calculate and update the fit, $E \leftarrow \|\Delta I\|^2$
end for
$\omega_{initial} \leftarrow min_E\{\Omega\}$
for each scale level **do**
  Run AAM optimization at $\theta_{initial}$ (max k iterations)
end for
return $\theta_{initial}$

3.2.2 Motion Modelling

When used for tracking in a video sequence, the initial estimate in a frame should be better predicted than just the adaptation from the previous frame. This could be done, for example, with a simple motion estimation in a few points, a Kalman filter, or a combination thereof.

A Kalman filter is an adaptive filter used to model the state of a discrete dynamic system. The technique was originally developed in 1960 to filter out noise in electronic signals [91], but has since found tracking and modelling applications in computer vision [92, 93].

Given a set of measurements taken from the system, the filter can estimate the next state of the system and adjust its model to allow for changes in the system behaviour [94].

The way the problem is formulated depends on the measurements that can be made and the results that are required. For instance, if one is interested in positional coordinates, rotation and scale, each of these can have a value and a first and second derivative with respect to time. This could be formulated as a single twelve dimensional state vector $x(n)$ with a corresponding state transition matrix.

Alternatively, each attribute could be considered independently. There would be four Kalman filters, each with a three dimensional state vector. This greatly simplifies the corresponding calculations for the Kalman gain coefficients. This is the method adopted here.

The output of the Kalman filters predicts the position of the object, which increase the robustness of the tracking system. The information output can minimize the search area of the next input frame. The search space of the fitting algorithm of the active appearance model is minimizes and improve the performance of the AAM and increase the accuracy of object tracking system.

In the Kalman filters, it assume that the system is linear, that observations of it are linear functions of the underlying state, and that noise, both in the system and in the measurement, is white and Gaussian. Formally, we have the model

$$x_{k+1} = A_k x_k + w_k \tag{3.9}$$

where the matrix A_k describe the evolution of the underlying model state, w_k is zero mean Gaussian noise.

$$z_k = H_k x_k + v_k \tag{3.10}$$

where the H_k are the measurement matrices, describing how the observations are related to the model, v_k is another zero mean Gaussian noise factor.

In our system, we assume the object move randomly, thus the motion model of the object is:

$$\mathbf{x_n} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} \tag{3.11}$$

$$\mathbf{x_k} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \end{pmatrix} + w_{k-1} \tag{3.12}$$

The measurement equation matrix is:

$$\mathbf{z_k} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} + v_k \qquad (3.13)$$

The complete picture of the operation of the Kalman filter is shown in Figure 3.2. In our system, x is the position of the object. In the time update part, it project the state and covariance estimates from the previous time $k-1$ to the current time step k. Function f predict the current position of the object with the information in the pervious frame. A_k and W_k are the process Jacobians at step k and Q_k is the process noise covariance at step k.

The measurement update equations shown in Figure 3.2, correct the state and covariance estimates with the measurement z_k. h is the measurement matrix, H_k and V are the measurement Jacobians at step k, and R_k in the measurement noise covariance at step k.

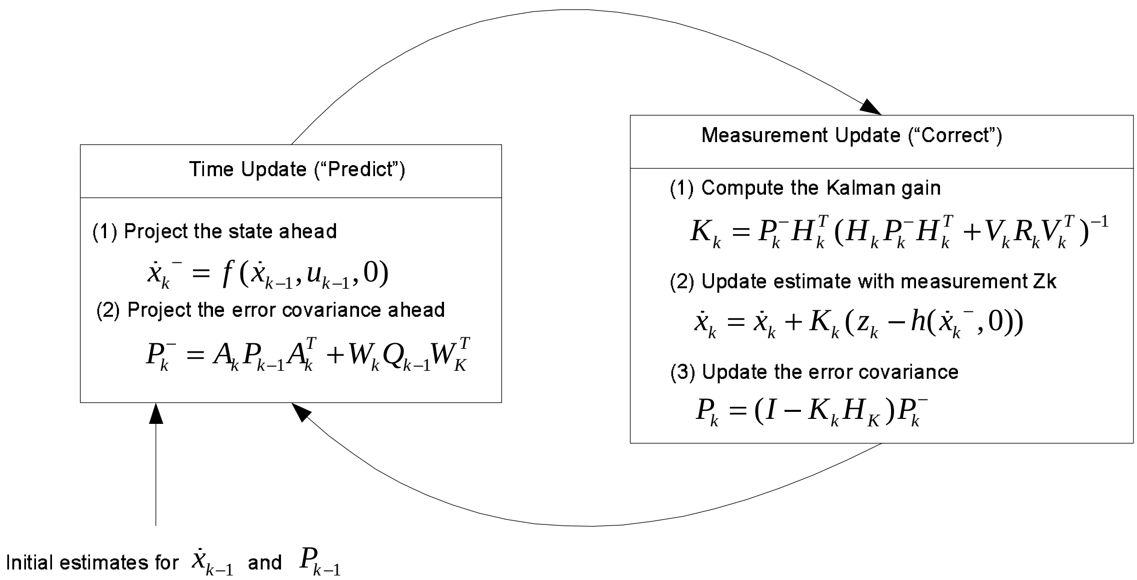

Figure 3.2: The complete picture of the operation of the Kalman filter

The matrix for estimating the motion of the object and the matrix for measuring the error can be obtained by doing the experiments. We will preform the experiment to obtain this two matrix.

3.2.3 Combining the Active appearance model and the Kalman filter

The output of the Kalman filter predicts the position of the object in the next video frame. In the fitting algorithm of the AAM, the shape model parameter and the appearance parameter are search iteratively in the search space. With the information provided by the Kalman, the searching space of the AAM fitting cane be reduced. Also, the fitting algorithm would fall into local minimum. However, the Kalman can predict the position of the object, this can reduce the chance of trapping in the local minimum.

3.3 Design Experimentation

Experiments will be done on the proposed scheme. The tracking system will implemented by using the AAM-API provided by DTU and the Kalman Filter provided by OpenCV. The system is implemented according to the architecture shown in Figure 3.1

The following steps will be done to conduct the experiments on the object tracking system:

- The training image set is extracted form the video.
- The object is annotated using 12 landmarks.

- Five training images are input to the AAM to train the shape model and the appearance model.

- Video with the object in the training image is input to the system and perform the object tracking.

The system will be run on the environment with the following configuration: Web cam as the input device of the system, a desktop computer with Pentium 4 CPU 2.00GHz and 512MB RAM. Also, evaluation on the performance of the scheme will be preformed.

3.4 Summary

In this section, an object tracking system with active appearance model and Kalman filter is proposed. The performance of the proposed scheme is improved. Experiments will be preformed to demonstrate it. Also, the occlusion detection would be added to the system, such that the accuracy of the system would be increased.

Chapter 4

Conclusion and Future Work

In this paper, we first reviewed the current technologies in the field of augmented reality, object tracking and the active appearance model. Then we proposed a scheme for object tracking. The scheme included several components: active appearance model, initialization, Kalman filter for prediction and occlusion detection. Experiment will done to demonstrate the improvement made by the proposed scheme.

The future of the work include:

- Implement the proposed object tracking scheme to demonstrate the improvement made.
- Include the occlusion detection to improve the accuracy of the system.
- Model the problem for Kalman filter more accurately.
- Improve the speed of the fitting algorithm in the active appearance model by using multi-resolution.

Acknowledgement

Prof. Michael R. Lyu provides valuable suggestions and comments and these give significant direction to my research.

Bibliography

[1] Mann, S., Smart Clothing: The "Wearable Computer" and WearCam, Personal Technologies, Vol. 1, No. 1, March 1997, Springer-Verlag

[2] Sutherland, I., A Head-Mounted Three Dimensional Display Proc. Fall Joint Computer Conference 1968, Thompson Books, Washington, DC, 1968, 757V764.

[3] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Transactions on Information Systems*, Vol. E77-D, No. 12, pp. 1321-1329.

[4] R. Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, Vol. 6, No. 4, pp. 355-385, August 1997.

[5] D. Sims, "New realities in aircraft design and manufacture," *IEEE Transactions on Computer Graphics and Applications*, Vol. 14, No. 2, pp. 91, March 1994.

[6] B. Wanstall, "HUD on the head for combat pilots," *Interavia 44*, pp. 334-338, April 1989. [A89-39227].

[7] M. Betto, J. Jorgense, and T. Riis, "The autonomous vision system on TeamSat," *Proceedings 1999 IEEE Aerospace Conference*, Aspen, Colorado, 6-13 March 1999.

[8] This system was developed by the User Interaction and Visualization group at the European Computer-Industry Research Centre (ECRC), which has now been transferred to the Fraunhofer Project Group for Augmented Reality in ZGDV.

[9] Ross T. Whitaker, Chris Crampton, David E. Breen, Mihran Tuceryan and Eric Rose, Object Calibration for Augmented Reality, ECRC-95-04, 1995. (Also appeared in Eurographics '95 Proceedings, pp. C-15 - C-27, September 1995).

[10] Feiner, S., Webster, A., Krueger, T., MacIntyre, B., and Keller, E., Architectural anatomy, In Presence, 4(3), Summer 1995, 318-325.

[11] Webster, A., Feiner, S., MacIntyre, B., Massie, W., and T. Krueger., Augmented reality in architectural construction, inspection and renovation, Proc. ASCE Third Congress on Computing in Civil Engineering, Anaheim, CA, June 17-19, 1996, 913-919.

[12] KARMA, Computer Graphics and UI Lab, Columbia University, Jan 2000 http://www.cs.columbia.edu/graphics/projects/karma/karma.html

[13] Feiner, S., MacIntyre, B., Hollerer, T., and Webster, T., A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment, In Proc. ISWC '97 (Int. Symposium on Wearable Computers), October 13-14, 1997, Cambridge, MA.

[14] GRIDS, Computer Graphics and Immersive Technologies Lab, University of Southern California, Jan 2000, http://www.usc.edu:80/dept/CGIT/

[15] M.R. Lyu, I. King, T.T. Wong, E. Yau, and P.W. Chan, “ARCADE: Augmented Reality Computing Arena for Digital Entertainment,” Proceedings 2005 IEEE Aerospace Conference, Big Sky, Montana, U.S.A., March 5-12 2005.

[16] L. Davis, E. Clarkson, and J. Rolland, “Predicting accuracy in pose estimation for marker-based tracking,” *Proceedings The Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 28-35, 7-10 October 2003.

[17] R. Azuma and C. Furmanski, “Evaluating label placement for augmented reality view management,” *Proceedings The Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 66-75, 7-10 October 2003.

[18] M. Ribo, P. Lang, H. Ganster, M. Brandner, C. Stock, and A. Pinz, ”Hybrid tracking for outdoor augmented reality applications,” *IEEE Computer Graphics and Applications*, Vol. 22, No. 6, pp. 54-63, November-December 2002.

[19] D. Beier, R. Billert, B. Bruderlin, D. Stichling, and B. Kleinjohann, “Marker-less vision based tracking for mobile augmented reality,” *Proceedings The Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 258-259, 7-10 October 2003.

[20] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-Based Object Tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligience*, Vol. 25, No. 5, pp. 564-577, May 2003.

[21] E.Oron, “Motion estimation and image difference for multi-object tracking,” *Proceedings 1999 IEEE Aerospace Conference*, Aspen, Colorado, March 6-13, 1999.

[22] U. Neumann and S. You, “Natural feature tracking for augmented reality,” *IEEE Transactions on Multimedia*, Vol. 1, No. 1, pp. 53-64, March 1999.

[23] P. Viola and M.Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp.511-518, 8-14 December 2001.

[24] Rainer Lienhart and Jochen Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," *Proceedings IEEE Iternational Conference in Image Processing* 2002, Vol. 1, pp. 900-903, September 2002.

[25] Shoichi Arakil, Takashi Matsuoka, Haruo Takemura, and Naokazu Yokoya, Real-time Tracking of Multiple Moving Objects in Moving Camera Image Sequences Using Robust Statistics. 1051-4651/98, 1998 IEEE.

[26] M. Bertalmio, G. Sapiro, and G. Randall, Morphing Active Contours. Proc. Int'l Conf. Scale-Space Theories in Computer Vision, pp. 46-57, 1999.

[27] R. Curwen and A. Blake, Dynamic Contours: Real-Time Active Splines. Active Vision, A. Blake and A. Yuille, eds., pp. 39-58. MIT Press, 1992.

[28] V. Caselles and B. Coll, Snakes in Movement. SIAM J. Numerical Analysis, vol. 33, pp. 2,445-2,456, 1996.

[29] M.P. Dubuisson, S. Lakshmanan, and A.K. Jain, Vehicle Segmentation and Classification using Deformable Templates. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 3, pp. 293-308, 1996.

[30] N. Diehl, Object-Oriented Motion Estimation and Segmentation in Image Sequences. IEEE Trans. Image Processing, vol. 3, pp. 1,901-1,904, Feb. 1990.

[31] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, Efficient and Effective Querying by Image Con-tent. J. Intelligent Information Systems, vol. 3, no. 1, pp. 231-262, 1994.

[32] Paul Fieguth, Color-Based Tracking of Heads and Other Mobile Objects at Video Frame Rates. Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97).

[33] T.D.Grove, K.D.Baker, and T. N. TAN, Color based object tracking. 14th International Conference on Pattern Recognition (CV41).

[34] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, Fast Geodesic Active Contours. Proc. Int'l Conf. Scale-Space Theories in Computer Vision, pp. 34-45, 1999.

[35] Gregory D. Hager and Peter N. Belhumeur, Efficient Region Tracking With Parametric Models of Geometry and Illumination. IEEE transactions on pattern analysis and machine intelligence, vol. 20, no. 10, pp. 1025-39 October 1998.

[36] Ryan C. Jones, Daniel DeMenthon, David S. Doermann, Building mosaics from video using MPEG motion vectors. 1999.

[37] A.K. Jain, Y. Zhong, and S. Lakshmanan, Object Matching Using Deformable Templates. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 3, pp. 267V278, Mar. 1996.

[38] Javed I. Khan, Zhong Guo and Wansik Oh, Motion based object tracking in MPEG-2 video stream for perceptual region discrimination rate transcoding. Proceedings of the 9th ACM international conference on multimedia. Pp. 572-576.

[39] W. Krattenthaler, K.J. Mayer, and M. Zeiller, Point Correlation: A Reduced-Cost Template Matching Technique. Proc. ICIP, pp. 208V212, 1994.

[40] M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active Contour Models. Int'l J. Computer Vision, vol. 1, pp. 321-332, 1988.

[41] Yun-Ting Lin and Yuh-Lin Chang, Tracking Deformable Objects with the Active Contour Model. 1997 IEEE.

[42] L. Wixson, Detecting Salient Motion by Accumulating Directionally-Consistent Flow. IEEE transactions on pattern analysis and machine intelligence, vol. 22, no. 8, August 2000.

[43] D. Metaxas and D. Terzopoulos, Shape and Nonrigid Motion Estimation Through Physics-Based Synthesis. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, no. 6, pp. 580-591, 1993.

[44] H.H. Nagel, G. Socher, H. Kollnig, and M. Otte, Motion Boundary Detection in Image Sequences by Local Stochastic Tests. Proc. European Conf. Computer Vision, vol. II, pp. 305-315, 1994.

[45] Natan Peterfreund, Robust Tracking of Position and Velocity With Kalman Snakes. IEEE transactions on pattern analysis and machine intelligence, vol. 21, no. 6, June 1999.

[46] Robert Pless, Tomas Brodsky, and Yiannis Aloimonos, Detecting Independent Motion: The Statistics of Temporal Continuity. IEEE transactions on pattern analysis and machine intelligence, vol. 22, no. 8, August 2000

[47] Nikos Paragios and Rachid Deriche, Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects. IEEE transactions on pattern analysis and machine intelligence, vol. 22, no. 3, pp. 266-280, march 2000.

[48] A. Pentland, R. Picard, and S. Sclaroff, Photobook: Tools for Content-Based Manipulation of Image Databases. Storage and Retrieval of Image and Video Databases II, Paper No. 2185-05, San Jose, Calif., pp. 34-47, SPIE, Feb. 1994.

[49] N. Paragios and G. Tziritas, Adaptive Detection and Localization of Moving Objects in Image Sequences. Signal Processing: Image Comm., vol. 14, pp. 277-296, 1999.

[50] Lili Qiu, Li Li, Contour Extraction of Moving Objects.

[51] Stan Sclaroff and Lifeng Liu, Deformable Shape Detection and Description via Model-Based Region Grouping. IEEE transactions on pattern analysis and machine intelligence, vol. 23, no. 5, pp. 475-489, May 2001.

[52] V. V. Vinod and Hiroshi Murase, Video Shot Analysis using Efficient Multiple Object Tracking. 1997 IEEE.

[53] Yu Zhong, Anil K. Jain, M.-P. Dubuisson-Jolly, Object Tracking Using Deformable Templates. IEEE transactions on pattern analysis and machine intelligence, vol. 22, no. 5, pp544-549, May 2000.

[54] A. Baumberg and D. Hogg. Learning flexible models from image sequences. In J.-O. Eklundh, editor, 3nd European Conference on Computer Vision, volume 1, pages 299V308. Springer-Verlag, Berlin, 1994.

[55] T. F. Cootes, D. H. Cooper, C. J. Taylor, and J. Graham. A trainable method of parametric shape description. In P. Mowforth, editor, 2nd British Machine Vison Conference, pages 54V61. Springer-Verlag, Sept. 1991.

[56] T. F. Cootes, D. H. Cooper, C. J. Taylor, and J. Graham. A trainable method of parametric shape description. Image and Vision Computing, 10(5):289V294, June 1992.

[57] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In H.Burkhardt and B. Neumann, editors, 5th European Conference on Computer Vision, volume 2, pages 484V498. Springer, Berlin, 1998.

[58] T. F. Cootes, G. J. Edwards, and C. J. Taylor. A comparative evaluation of active appearance model algorithms. In P. Lewis and M. Nixon, editors, 9th British Machine Vison Conference, volume 2, pages 680V689, Southampton, UK, Sept. 1998. BMVA Press.

[59] T. F. Cootes and C. J.Taylor. Active shape models. In D. Hogg and R. Boyle, editors, 3rd British Machine Vision Conference, pages 266V275. Springer-Verlag, Sept. 1992.

[60] T. F. Cootes, C. J.Taylor, D. Cooper, and J. Graham. Training models of shape from sets of examples. In D. Hogg and R. Boyle, editors, 3rd British Machine Vision Conference, pages 9V18. Springer-Verlag, Sept. 1992.

[61] T. F. Cootes and C. J. Taylor. Active shape model search using local grey-level models: A quantitative evaluation. In J. Illingworth, editor, 4th British Machine Vision Conference, pages 639V648. BMVA Press, Sept. 1993.

[62] T. F. Cootes and C. J. Taylor. Combining point distribution models with shape models based on finite-element analysis. In E. Hancock, editor, 5th British Machine Vison Conference, pages 419V428. BMVA Press, Sept. 1994.

[63] T. F. Cootes and C. J. Taylor. Modelling object appearance using the grey-level surface. In E. Hancock, editor, 5th British Machine Vison Conference, pages 479V488, York, England, Sept. 1994. BMVA Press.

[64] T. F. Cootes and C. J. Taylor. Combining point distribution models with shape models based on finite-element analysis. Image and Vision Computing, 13(5):403V409, 1995.

[65] T. F. Cootes and C. J. Taylor. A mixture model for representing shape variation. In A. Clarke, editor, 8th British Machine Vison Conference, pages 110V119. BMVA Press, Essex, Sept. 1997.

[66] T. F. Cootes and C. J. Taylor. A mixture model for representing shape variation. Image and Vision Computing, 17(8):567V574, 1999.

[67] T. F. Cootes and C. J. Taylor. Combining elastic and statistical models of appearance variation. In 6th European Conference on Computer Vision, volume 1, pages 149V163. Springer, 2000.

[68] T. F. Cootes and C. J. Taylor. Constrained active appearance models. In 8th International Conference on Computer Vision, volume 1, pages 748V754. IEEE Computer Society Press, July 2001.

[69] T. F. Cootes, C. J. Taylor, and A. Lanitis. Active shape models : Evaluation of a multi-resolution method for improving image search. In E. Hancock, editor, 5th British Machine Vison Conference, pages 327V336, York, England, Sept. 1994. BMVA Press.

[70] T. F. Cootes, C. J. Taylor, A. Lanitis, D. H. Cooper, and J. Graham. Building and using flexible models incorporating grey-level information. In 4th International Conference on Computer Vision, pages 355V365. IEEE Computer Society Press, May 1993.

[71] T. F. Cootes, K. N. Walker, and C. J. Taylor. View-based active appearance models. In 4th International Conference on Automatic Face and Gesture Recognition 2000, pages 227V232, Grenoble,France, 2000.

[72] T. F. Cootes, G. Wheeler, K. Walker, and C. J. Taylor. Coupled-view active appearance models. In M. Mirmehdi and B. Thomas, editors, 11th British Machine Vison Conference, volume 1, pages 52V61, Bristol, UK, Sept. 2000. BMVA Press.

[73] I. Craw and P. Cameron. Parameterising images for recognition and reconstruction. In 2nd British Machine Vison Conference, pages 367V370. Springer-Verlag, London, 1991.

[74] G. Edwards, C. J. Taylor, and T. F. Cootes. Interpreting face images using active appearance models. In 3rd International Conference on Automatic Face and Gesture Recognition 1998, pages 300V305, Japan, 1998.

[75] G. Hamarneh and T. Gustavsson. Deformable spatio-temporal shape models: Extending asm to 2d+time. In T. Cootes and C. Taylor, editors, 12th British Machine Vison Conference, pages 13V22, 2001.

[76] G. Harmarneh. Deformable spatio-temporal shape modeling. Masters thesis, Department of Signals and Systems, Chalmers University of Technology, Sweden, 1999.

[77] J. Haslam, C. J. Taylor, and T. F. Cootes. A probabalistic fitness measure for deformable template models. In E. Hancock, editor, 5th British Machine Vison Conference, pages 33V42, York, England, Sept. 1994. BMVA Press, Sheffield.

[78] A. Hill, T. F. Cootes, and C. J. Taylor. Active shape models and the shape approximation problem. In 6th British Machine Vison Conference, pages 157V166. BMVA Press, Sept. 1995.

[79] A. Hill, A. Thornham, and C. J. Taylor. Model-based interpretation of 3D medical images. In J. Illingworth, editor, 4th British Machine Vision Conference, pages 339V348. BMVA Press, Sept. 1993.

[80] A. Lanitis, C. Taylor, and T. Cootes. Automatic tracking, coding and reconstruction of human faces using flexible appearance models. IEE Electronic Letters, 30:1578V1579, 1994.

[81] M.Rogers and J.Graham. Robust active shape model search. In 7th European Conference on Computer Vision, volume 4, pages 517V530. Springer, 2002.

[82] C. Nastar, B. Moghaddam, and A. Pentland. Generalized image matching: Statistical learning of physically-based deformations. In 4th European Conference on Computer Vision, volume 1, pages 589V598, Cambridge, UK, 1996.

[83] S. Romdhani, S. Gong, and A. Psarrou. A multi-view non-linear active shape model using kernel PCA. In T. Pridmore and D. Elliman, editors, 10th British Machine Vison Conference, volume 2, pages 483V492, Nottingham, UK, Sept. 1999. BMVA Press.

[84] S.Baker and I.Matthews. Equivalence and efficiency of image alignment algorithms. In Computer Vision and Pattern Recognition Conference 2001, volume 1, pages 1090V1097, 2001.

[85] P. Sozou, T. F. Cootes, C. J. Taylor, and E. D. Mauro. A non-linear generalisation of pdms using polynomial regression. In E. Hancock, editor, 5th British Machine Vison Conference, pages 397V406, York, England, Sept. 1994. BMVA Press.

[86] B. van Ginneken, A.F.Frangi, J.J.Stall, and B. ter Haar Romeny. Active shape model segmentation with optimal features. IEEE-TMI, 21:924V933, 2002.

[87] C. Wolstenholme and C. J. Taylor. Wavelet compression of active appearance models. In MICCAI, pages 544V554, 1999.

[88] Real-time combined 2D+3D active appearance models Jing Xiao; Baker, S.; Matthews, I.; Kanade, T.; Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on Volume 2, 27 June-2 July 2004 Page(s):II-535 - II-542 Vol.2

[89] X. Hou, S. Li, H. Zhang, and Q. Cheng. Direct appearance models. In Computer Vision and Pattern Recognition Conference 2001, volume 1, pages 828V833, 2001.

[90] S.Yand, C.Liu, S.Z.Li, H.Zhang, H.-Y. Shum, and Q.Cheng. Texture-constrained active shape models. In A.Pece, editor, Workshop on Generative Model-Based Vision, pages 107V113, Copenhagen, 2002.

[91] KALMAN, R. E. A new approach to linear filtering and prediction problems. Transactions of the ASME, Journal of Basic Engineering. 1960.

[92] DERICHE, R., AND FAUGERAS, O. Tracking line segements. Image and Vision Computing. 8. 261-270. 1990.

[93] HARRIS, C., AND STENNETT, C. RAPID - a video rate object tracker. In Proceedings of the British Machine Vision Conference, 73-77. 1990.

[94] DOBIE, M. R., AND LEWIS, P. H. The application of a Kalman filter to object tracking. 1992 Research Journal, Department of Electronics and Computer Science, University of Southampton. 45-47. 1992.