Using Video to Learn Faces

Robin Hewitt
rhewitt@acm.org

1. Overview

The goal of this project is to enhance the visual capabilities of Mabel – an autonomous, social robot – by adding motion segmentation and visual tracking. Segmentation and tracking have many potential uses in a robotics context, but the focus in this project will be on using these methods to improve face detection and recognition.

This project will begin with a literature survey to identify appropriate methods for motion-based segmentation and for visual tracking. Once the methods are selected, they will be implemented. The selected methods will then be tested to evaluate their ability to do the following:

- Track a face and head through a range of motions, in the presence of background clutter, and during a lighting change
- Separate image features into foreground, background, and contour classes
- Increase the number of useful positive examples available for learning

The next section describes the existing recognition method and the rationale behind it. Following that are details of this project’s objectives, then a description of the datasets to be used and the purpose of each. Test methodology and evaluation criteria follow that.

2. Existing recognition method

Mabel’s existing face detection and recognition system generates a face model based on one video frame. The user captures a single frame with a frontal face view, then draws a bounding rectangle to indicate its location. Users are also asked to mark the centers of both eyes and the nose tip.

The face image is then “parsed” to create a face model. The face model consists of two types of features – large-scale features that consist of simple, paired light and dark areas and small-scale features that are centered on interest points and are represented by a descriptor similar to the SIFT descriptor in [6].

During search, the large-scale features are used to rapidly filter out all but a few candidate locations and scales for a particular face. Each of these candidate matches are then separately evaluated by matching to the small-scale features. Closeness to the input model is evaluated as the sum of local deformations to bring the search region and the input model into correspondence.

This method differs from the usual approach to face recognition. The usual method is to do face detection separately using a statistical classifier. Detected faces are then aligned and passed to a second classifier for the recognition step. The statistical classifier for face detection is trained using a face database.

This approach can be problematic, however, if the database is biased in a way that’s unlike the population bias in the deployment context. An example of relevant database bias is camera position. Face databases typically contain images captured with the camera at eye level. Yet, for psychological and economic reasons, personal robots are usually short, and the camera is often well below eye level. Faces seen from this perspective look quite different. Other relevant biases in face databases include pale skin and frontal lighting.

Some deployment contexts, such as mugshot identification, are well characterized. In these contexts a statistical classifier makes sense. But the deployment context for a social robot is very open-ended. Whether developed as toys or as helpmates, these systems will likely be more successful (both in terms of entertainment and utility value) if they are able to adapt to varying circumstances.

Although Mabel’s current recognition system does not achieve the ultra-high detection levels of statistical classifiers that are tested and then trained on images drawn from the same database, it has the advantage of being entirely user driven. The current project will extend this system’s ability to learn flexibly by using continuity in the video feed to extract more information from the initial user input and to improve the quality of that information.

3. Project Objectives

The general goal for this project is to add motion-based segmentation and tracking capabilities to Mabel. The initial proposed use for these methods defines additional, more specific objectives. This initial use is to generate additional, more detailed data that can be used to improve the existing face-recognition system.
Select methods to implement The first objective for this project is to select and implement appropriate methods for both motion-based segmentation and tracking. This effort is expected to take about one or two weeks. Due to time constraints, it’s not intended that it be exhaustive. The outcome will be a high-level overview of the major approaches along with a list of pros and cons for each. One method or a combination of several methods will then be selected to implement in Mabel.

Many different approaches to object tracking have been developed. Candidate methods for this project include Predictive Filtering, as in [7] and [9]. Discriminative-Generative tracking, as in [5], and Mean-Shift tracking, presented in [2]. Two intriguing tracking methods were published recently: Jin et al. [4] model the effects of motion blur for tracking, while Shi and Karl[8] developed a real-time implementation for tracking multiple objects using level sets.

Tracking must begin with some sort of segmentation. For this project, a motion-based segmentation method will be used. A candidate segmentation method is described in [10].

Classify features as interior, exterior, or contour The small-scale features in the face model are located using interest-point detectors. Both corner and blob features are detected. Currently, all interest points within the bounding rectangle are considered face features. Some of these features are generated by background elements that happen to fall within the bounding rectangle. Eliminating these features from the face model should improve classification accuracy.

The remaining features are either interior points or are generated by the transition between foreground and background. The features that correspond to interior points are clearly good candidates for including in the face model. Contour points may also be useful features, as demonstrated in [3]. But their best representation and use may differ from that of interior features.

Accurately classifying interest-point features as interior, exterior, or contour is the second objective for this project. The proposed method for doing this is motion-based segmentation. The data for segmenting will consist of video feed captured while the user is interacting with the input interface. During this interaction, the user positions himself prior to capturing a video frame. This step is likely to involve some head motion. Afterwards, while drawing the bounding rectangle, the user is also likely to shift positions to some extent.

Capture additional face images Mabel’s face recognition method does not yet take advantage of more than one input example, but extending it to do so is a future goal. Although the user can certainly provide additional images by repeating the input process, it would be much nicer for the robot to automatically gather additional examples with no extra user effort. One way to do that is by automatically tracking the user’s face while it moves and storing example frames. Thus, the third project objective is to automatically capture face examples that 1) differ from what the user entered, 2) are positive examples with a high degree of certainty, and 3) are in some way typical of what the robot will likely see for this user.

Characterize failure conditions While tracking and motion-based segmentation can potentially be used to improve face recognition, these methods have their own limitations. If these methods go too badly astray, they can introduce significant errors into the face model and do more harm than good when used in the ways described above. To keep that from happening, it’s important to set “safe” limits when applying these methods.

Safe limits can be defined as the number of frames that can be tracked reliably and the minimum acceptable capture rate. The first step in selecting these limits is to acquire video sequences of face and head motion while systematically varying parameters likely to affect segmentation and tracking. These sequences will serve as the basis for testing both the segmentation and the tracking method. During test, different frame rates can be simulated by skipping frames in these stored sequences.

The actual method for setting the limits will to some extent depend on the results of testing. If the situations that cause these methods to fail can be characterized in ways that make them detectable, the number of tracked frames and the frame rate can be adjusted dynamically. If the failure mode cannot be reliably detected, safe values will need to be more conservative. In fact, one possible outcome is that the risk of introducing errors into the face model outweighs the potential benefits of using the selected methods.

Given the time constraints for completing the project, the objective for characterizing failure is limited to testing each sequence at full frame rate and visually identifying the frame number at which tracking failure is apparent.

4. Data

The test data for this project will consist of video sequences captured with the robot’s laptop. The video sequences will be stored in a non-lossy, compressed format so that raw bitmap data can be recreated for subsequences of arbitrary length. Three sets of video sequences will be used for testing.
Dataset 1  The first dataset will consist of a minimum of five video sequences of users initiating and using the input program for learning a face. When the user activates the interface, video feed is displayed. The user clicks to select one frame for learning, draws a bounding rectangle to specify the object to be learned, then clicks three alignment points (eye centers and nose tip). Their motions throughout this process will be captured and stored as part of each test sequence, but will not be displayed in the user interface. The purpose of these sequences is to provide test data to measure how accurately motion-based segmentation classifies features in the user’s input as exterior, interior, or contour.

Dataset 2  The second dataset will contain face sequences with specified head motions. The movements will increase in difficulty from simple to more complex to determine when face segmentation and tracking fail. The simplest case will be a slow, side-to-side swaying motion against a plain background. This motion will be repeated more rapidly. The next level of difficulty will be head rotations – side to side, then looking up and down. This series of motions will also be done twice – once slowly, then again more rapidly. The third type of motion will consist of head shots of a person leaning toward the camera, then backing away. Again, this will be at two speeds. The entire series will be repeated against a cluttered background.

The purpose of this dataset is to characterize when and how tracking and motion segmentation fail.

Dataset 3  The third dataset of video sequences will consist of head motions such that the face enters and leaves an area that has strong side lighting from a window. This is a difficult lighting situation for object recognition. It’s also a situation that an indoor robot is likely to encounter frequently. This third set of video sequences will be used to evaluate tracking and segmentation capabilities in this lighting context and to evaluate the effects of lighting transitions.

5. Testing

The tracking and segmentation methods will be evaluated in terms of how well they can support visual learning as described in Section 3.

5.1. Feature classification

The first test series will measure the accuracy of feature classification into background, foreground, and contour classes. Classification will be implemented using motion-based segmentation. These tests will use dataset 1. Performance will be evaluated as the percentage of correct classifications for each of the three feature classes.

5.2. Failure conditions

The next set of tests will compare tracking quality under various conditions. These tests will use datasets 2 and 3. In each sequence, the tracker will be initialized near the beginning of the sequence. The frame at which tracking begins to fail will be determined manually. The track length, in frames, for all tracking tests will be tabulated.

The purpose of these tests is to prepare a visual guide for identifying any conditions under which the selected tracking method is particularly suspect. Depending on the tracker’s performance, these values may be sufficient to set a safe maximum for the number of tracked frames. The video data and initial tracking results can also serve as a relevant basis for comparing tracking methods in future.

5.3. Generating face examples

The final set of tests will provide insight into whether tracking plus frame extraction can be used to automatically generate additional face examples that are useful for learning. It will also provide information about how these examples can best be utilized. These tests will use all three datasets. In each sequence, tracking will proceed as in the previous tests. It will continue up through the interval where tracking quality is good.

The tracked region will be extracted from every Nth frame of the video sequence. (N is a parameter that will be based on the results of the previous test.) Each of these extracted regions will be classified as match or non-match using the face model produced by this user’s initial input frame.

For each sequence, the first frontal-view face image (if any) that exceeds the recognition threshold will be used to create a second face model. This second model will be used to classify each of the non-matches from the previous step. The results from this process will be reported as 1) the number of sequences in which the initial face model correctly classified all of the extracted frontal-view face images, 2) the number of frontal face views in each video sequence that were originally misclassified, and 3) the number of corrected misclassifications in each video sequence. The second result is only applicable to sequences that include more than one misclassified frontal-view face image.

The outcome of these tests will be used to guide the decision on how (and whether) to use views generated during tracking to improve the initial face model. If the percentage of sequences in which the initial face model correctly classified all of the extracted frontal-view face images is high, that suggests the best use for the tracker may be to generate a visual manifold of poses. If, however, a large number of misclassified frontal face views are found, and many of these are corrected with the second face model, that suggests that tracking is a useful way to extract additional, in-
formative, frontal face views.

6. Schedule

6.1. Time Estimates

This project’s tasks and estimated time to complete each are summarized below:

Task 1 – Data acquisition This task consists of acquiring 15-20 video recordings. The recording time itself is quite short. The shots may need to be scheduled over a period of several weeks, however. For planning purposes, this process is scheduled to run concurrently with the literature survey. But it may extend an additional week or two.

Task 2 – Literature survey The literature survey is expected to take from one to two weeks. Due to time constraints, it’s not intended that it be exhaustive.

Task 3 – Implementation This task consists of implementing the selected motion-based-segmentation and tracking methods. If implementations for the selected methods are available as Matlab scripts or as executable code, these implementations will be used for testing. If both the segmentation and the tracking method must be implemented from scratch, time constraints may make it necessary to choose just one of these methods to implement. In that case, motion-based segmentation will be the most likely candidate, since it’s potentially useful on its own whereas tracking relies on prior segmentation. But in this situation, the results of the literature survey will also help guide the decision as to what to implement.

Task 4 – Test feature classification This task consists of running the tests described in Section 5.1. Steps include generating the small-scale features for the face model and hand labeling each of these as interior, exterior, or contour. The motion-based segmentation method will then be applied to separate the motion region from the background. The class assignment for each feature will require a mat-lab script. Total time for this task is estimated as about three days. For planning purposes, it’s scheduled to take one week.

Task 5 – Evaluate failure conditions This task consists of running the tracking program to mark each frame in several video sequences, then manually selecting the frames where tracking failure is visible. The main effort for the task will be in writing a C++ or Java utility to sequence through the captured frames. Since this utility is similar to programs I’ve already developed, this task is expected to take not more than about three days.

Task 6 – Generate face examples The steps for generating the face examples are given in Section 5.3. This task is given one week in the planning schedule.

6.2. Milestone Deadlines

- 16 Jan: task 1, Literature survey, completed
- 23 Jan: task 2, Data acquisition, completed
- 17 Feb: task 3, Implementation, completed
- 28 Feb: tasks 4-6, Testing, completed
- 6 Mar: Data analysis and draft report completed

7. Qualifications

I’ve completed two computer vision courses at UCSD: CSE 152, Intro Computer Vision (Spring 2004) and CSE 252C, Selected Topics in Vision & Learning (Fall 2004). For 252C, I wrote a face finder that generalizes from a small number of face examples (about five) to detect faces of previously unseen individuals.

I’m currently part of a hobbyist team that’s designing and building a social robot. I developed the vision framework for this project. The framework uses DirectShow for video capture and includes both realtime visual feedback and logging for monitoring visual behaviors. It provides an external API for the robot’s personality controller to make visual requests such as “check for obstacles” or “look for object X.” The existing functionality is based on simple methods to support initial development and software integration. My current efforts are directed towards improving and replacing these methods with more capable ones. This project is one of those improvements.

References


