16-720 Final Project Proposal: Motion Tracking and Motion Classification (E6)

1 Motivation

In homework 3, we walked through a basic algorithm [1, 2] for using optical flow histograms for classifying actions in the KTH action dataset. We computed histograms of the optical flow at 3D Harris keypoints throughout the image sequences and used a bag of words approach to describe actions from the training data. While this proved to be quite effective and ultimately yielded us an 80% classification success rate, there seem to be a few simple modifications that would intuitively improve the algorithm and potentially improve performance when generalized to other data sets. Namely, when it comes to identifying human motion templates, it would make sense to identify and track specific body parts [3]. In addition, there might be potent information in the periodicity of the motion throughout the sequence [4]. We would like to explore how we can use these concepts to extend the baseline implementation we had developed in the homework.

2 Proposed approaches

2.1 Labeling the keypoints

Instead of binning the optical flow descriptors from all the keypoints in an input sequence into a single histogram, we want to separately consider the flow descriptors from distinct body parts. For example, hand-waving may be characterized by lots of motion in the arms and relatively little motion in the feet, as compared to walking or running. [3] identifies body parts using silhouettes obtained from a known background, which cannot be applied to the KTH dataset. We plan to build an independent classification algorithm that first labels each keypoint as part of a leg, arm, torso, or image background. This would allow us to first discard the irrelevant (background) keypoints selected by the Harris detector and to then consider an intuitively more descriptive feature (optical flow histogram) vector that is three (one per body label) times the original length.

The challenge is to train an effective classifier that can correctly label the keypoints. The simplest approach would consider only one frame at a time. We can consider the number of keypoints in that frame (lots of points suggests noise), the relative positions of each keypoint within the frame, or the texture (using Gaussian or LoG filters) of a window around each keypoint. More intricate methods could also try to use context from preceding or subsequent frames and perhaps track the keypoints throughout the sequence. After designing the classifier, we would hand label the keypoints from multiple sequences and train this sub-algorithm with a standard technique like k-means or SVM.

2.2 Tracking of user-identified body parts

Since it may be hard to build an effective and general keypoint classifier, we alternatively consider a more direct way of tracking body parts for classifying human motion. At the start of each sequence, we can ask the user to define bounding boxes around the hands, feet, and torso of the subject. We can then apply Lucas

Kanade template tracking [5] to each window and save the estimated motions. Histograms summarizing the velocities of each body part (separately) can then be used to distinguish the motion templates.

2.3 Frequency analysis

Histogramming the velocities throws away temporal information that can in fact tell us a lot about the motion [4]. However, once we have the sequence of velocity vectors from section 2.2, we can directly apply Fourier analysis to compute characteristic frequencies of each body part for each motion template. This will give us an additional set of (histogram-able) features that may help us classify the motions.

3 Plan of action

The latter two ideas seem more practical for implementation right away, so we plan to start with those. If time permits, we may then try to build a keypoint classifier as described in section 2.1

References

[1] I. Laptev, "On space-time interest points," *Int. J. Comput. Vision*, vol. 64, no. 2-3, pp. 107–123, Sep. 2005. [Online]. Available: http://dx.doi.org/10.1007/s11263-005-1838-7

[2] I. Laptev and T. Lindeberg, "Local descriptors for spatio-temporal recognition," *Spatial Coherence for Visual Motion Analysis*, pp. 91–103, 2006.

[3] I. Haritaoglu, D. Harwood, and L. Davis, "W4: real-time surveillance of people and their activities," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 809–830, 2000.

[4] R. Cutler and L. Davis, "Robust real-time periodic motion detection, analysis, and applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 781–796, 2000.

[5] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

Implementation and Comparison of Human Action Recognition Algorithms

1 Abstract

Human action recognition in videos is a challenging problem with wide applications. In recent year, it becomes a hot topic of research in computer vision. A lot of effort has been devoted into both developing better tools and collecting larger and more challenge video datasets for action recognition. Yet, a comprehensive study of the state-of-art tools and datasets on action recognition has not been done yet.
The goal of this paper is two folds: 1) I will compare and combine the state-of-art features, feature representation methods, classifiers and fusion methods to get a best action recognition system and see how far we can go on action recognition using current tools we have; 2) I will evaluate the value of different datasets by studying the bias of each dataset.
I have got some preliminary results on the work goal. By using SIFT, STIP and MoSIFT and standard kernel SVM, I got 88% mean classification accuracy on UCF50 dataset and 48% mean classification accuracy on HMDB dataset. Both are better than the best reported results to date.

2 Related work

1) H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In CVPR, 2011.
2) Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo.Trajectory-based modeling of human actions with motion reference points. In ECCV, 2012.
3) A. Torralba and A. Efros. Unbiased look at dataset bias. In CVPR, 2011.

3 Tools, Code and Datasets

Tools: OpenCV, Matlab, C, C++
Code: Dense trajectory, STIP, MoSIFT, SIFT, CSIFT, SVM, Kernel Regression etc.
Dataset: UCF50, HMDB, KTH (maybe), Hollywood2(maybe)

Detect the State of Eyes

Basic Idea

The task at hand is to detect the state of a persons eyes; whether open or closed. This would be very useful to help detect if a driver is falling asleep while driving. Instead of the usual edge detection approach, a dual state model based system can be used. The parameters to be used are the iris and the inner and outer corners of the eyes.

Related work

The two main papers I will refer to are:

- Liu, H., Wu, Y., Zha., H.: Eye states detection from color facial image sequence. In: SPIE International Conference on Image and Graphics, vol. 4875. (2002) 693–698 8.
- Tian, Y., Kanade, T., Cohn, J.F.: Dual-state parametric eye tracking. In: Inter- national Conference on Automatic Face and Gesture Recognition. (2000)

Software

My understanding of the problem so far requires implementing Lucas-Kanade and some edge detectors to get the eye maps. I plan to use MATLAB for all of these.

Dataset

The authors on the second paper which came out of the Robotics institute at CMU mentioned that they used the Pitt-CMU Facial Expression AU Coded Database. I hope to get access to this.

Project Timeline

The two main tasks of this project are:

- Preprocessing- Assuming that the first frame is the initial position of the eye, normalizing the image for changes in brightness
- Detection- Detecting the inner corners of the eyes using some version of Lucas Kanade. Assuming that the outer corners are collinear, these can be detected after this.
- Iris detection-Using intensity and the presence of edges to detect the iris. If it is detected, the eyes are open, if not they are closed.

I do want to finish my project around Thanksgiving and so in the next 5 weeks, I anticipate that the initial stage of getting the database and preprocessing to take around a week. The next two tasks should take two weeks each.

Moving Object Detection Under Varying Environments Using Background Subtraction

Abstract

Background subtraction is a method typically used to segment moving regions in image sequences taken from a static camera by comparing each new frame to a model of the scene background. However, in real motion detection problems, the background will be changing, in terms of illumination, small motions on the "still" objects. To solve such tricky problems, firstly, we will follow the main reference papers to build a robust non-parametric model of the background used for moving object detection given a sequence of images, so that the model can handle situations where the background of the scene is not completely static but contains small motions such as tree branches and bushes. And then, we will follow the instructions in the papers to suppress detection of shadows, so that to improve the result of the detected moving object.

Expected Results

Use the video file we create on campus, which including moving objects (people, cars) and non-static background (with small motions of tree braches), to detect the moving targets with our algorithms. We can obtain sharp detection of the moving targets and the rest of background.

Software

MATALB

Timeline

<table>
<tr><th>Week 1</th><th>Week 2</th><th>Week 3</th><th>Week 4</th><th>Week 5</th><th>Week 6</th><th>Week 7</th><th>Week 8</th></tr>
<tr><td>10.21~10.28</td><td>10.29~11.4</td><td>11.5~11.11</td><td>11.12~11.18</td><td>11.19~11.25</td><td>11.26~12.2</td><td>12.3~12.9</td><td>12.10~12.13</td></tr>
<tr><td>1. Collect data;

2. Read main references and go through the math;

3. Exchange ideas on the baseline system.</td><td colspan="2">1. Pre-process the data;

2. Implement algorithms to build the background model;

3. Tune parameters.</td><td colspan="2">1. Debug

2. Implement algorithms for shadow detection</td><td>1. Algorithms Robustness testing and error analysis;

2. Debug and think out ways to improve performance.</td><td>1. Try to implement new possible ways;

2. Review the project works.</td><td>1. Poster preparation</td></tr>
</table>

Main References

Elgammal, D. Harwood, and L.S. Davis, "*Non-parametric Model for Background Subtraction*", ICCV Frame Rate Workshop, 1999.

Anurag Mittal, Nikos Paragios, "Motion-Based Background Subtraction using Adaptive Kernel Density Estimation", CVPR 2004

Implement Eulerian Video Magnification

Abstract

Videos captured in everyday life contain a lot of motion which we are not able to perceive. For example, the human eye cannot see oscillations at very high frequencies such as the movement of a guitar string, or low amplitude oscillations like a human pulse. In this project we will implement Eulerian Video Magnification [2]. This technique allows us to amplify temporal frequencies in an ordinary video (captured at 25-30 fps). We will use this technique (as demonstrated in the paper) to determine human heart rates from color variation due to blood flow in the face, or spatial deformations in the wrist. We will also use this technique to study correlations between vibrations and sound in commonly found objects such as clocks or speakers.

Related Work

Our primary reference is [2]. Poh et. al. [1] also use temporal variations to detect pulse.

Software Requirements

We plan to use OpenCV to implement the technique in [2]. We will use OpenCV to create a GUI that lets the user select temporal frequencies to amplify.

Dataset

We will capture videos of everyday objects and faces. We also plan to use videos available on YouTube.

Timeline

Date	Goal
2012-10-18 Thu	Proposal
2012-11-05 Mon	OpenCV Source
2012-11-19 Mon	Preliminary Results
2012-12-03 Mon	Final Report

References

[1] Ming-Zher Poh, Daniel J. McDuff, and Rosalind W. Picard. Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. *Opt. Express*, 18(10):10762–10774, May 2010.

[2] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William T. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph. (Proceedings SIGGRAPH 2012)*, 31(4), 2012.

Object Recognition Using Spatial Pyramid Matching Method

1 Background

The two main parts of the object recognition system are the image representation and classification. Currently, there are many efforts in devising powerful and smart representations and integrate them to detect and recognize objects in images. For the feature part, there are many features being designed which are not only discriminative but also be invariant to scale and orientation. Good examples are HOG, SIFT and SURF features. Based on these features, we could build the object recognition models using different image representations and classifications. Popular models include the Spatial Pyramid Matching[2] which uses pixel feature statistics at different levels and non-linear category SVM classifer, and Deformable Parts Model[1] which also takes into account the parts configuration and deformation. To better improve the accuracy and efficiency of object recognition, Yang et al.[4] proposed to make use of local Sparse Coding[3] to code features and then use a Linear SVM to do recognition task. Experimental results demonstrate that this method works pretty well in benchmark dataset Caltech 101 and Caltech 256[1] as achieving the state-of-art performances.

2 Project Milestones

In our project, we have three planned milestones marking different degrees of our goal, note that we will progress to the third step if we still have time when finishing the first two steps.
• Firstly, we are going to reproduce the result of the classical SPM method[2] on both caltech 101 and caltech 256 to construct the baseline.
• Then, we are going to implement the SPMSc method introduced in [4] and campare the result of SPM with and without local sparse coding scheme.
• In the third step, we plan to come up with some improved method based on our observations of the coding method of feature statistics to improve the two methods mentioned above.

References

[1] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[2] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[3] H. Lee, A. Battle, R. Raina, and A.Y. Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19:801, 2007.

[4] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[1] http://www.vision.caltech.edu/archive.html

Using Kinect to Detect Intent to Interact in 3D Interfaces

Abstract
One challenge of bringing 3D interfaces into multi-user environments is in determining who is trying to interact. One approach is to use a marker-based tracking system, however this is expensive, cumbersome and becomes infeasible in public spaces such as at malls or airports. Another approach is to use a depth-based skeletal tracking system which requires no user instrumentation, such as the Microsoft Kinect. Current Kinect-enabled interactive systems either assume that the person closest to the screen is trying to interact with the system, or looks for an engagement gesture such as a wave or raising one's hand. Unfortunately, heuristics such as raising a hand or using the person closest to the screens lead to many false positives, while an engagement gesture can be time-consuming and difficult to trigger. The aim of this project is to develop an algorithm for automatically determining who is trying to interact with an interface without requiring any upfront gesture such as a wave. I will develop and compare several algorithms for detecting engagement: a baseline heuristic (closest person whose hand is raised), a more refined heuristic approach, and finally a machine learning based approach. The contributions of this project will be new algorithms for automatically detecting intention to interact as well as a comparison of how well existing new algorithms work.

Related Work
Bohus, D., & Horvitz, E. (2009). Learning to predict engagement with a spoken dialog system in open-world settings. *... Special Interest Group on Discourse and Dialogue*, (September), 244–252. Retrieved from http://dl.acm.org/citation.cfm?id=1708411

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., et al. (2011). Real-time human pose recognition in parts from single depth images. *Cvpr 2011*, 1297–1304. doi:10.1109/CVPR.2011.5995316

Song, Y., Demirdjian, D., & Davis, R. (2012). Continuous body and hand gesture recognition for natural human-computer interaction. *ACM Transactions on Interactive Intelligent Systems*, *2*(1), 1–28. doi:10.1145/2133366.2133371

Dataset
I will collect about 50 recording of just one person playing games which will be used for my preliminary train and test set. I plan to collect data using Kinect Studio, a recording program that ships with Kinect for Windows SDK.

I plan to label data using Visual Gesture Builder, which is a program that ships with the Xbox SDK. This generates labeled videos files, which I will then post-process into a format that labels each user frame with engaged or not engaged.

I also plan to collect around 50 recordings of multiple people playing games using the Microsoft Kinect with various distractors (other people around the player) and with primary player switching (primary player goes from one person to another).

I will write my own software to generate labels for each player in the scene as engaged or not engaged. (stretch for the project timeframe).

Software
Feature extraction. Takes depth, skeleton, user mask frames and generates per-frame features.
Software to generate player labels (stretch for project timeframe).
Algorithm evaluation will be done using weka.
Implementing optimal machine learning algorithm into a program that does live classification.
Implement a heuristic algorithm that does live classification, can also take as input pre-labeled data for evaluation.
Implement baseline algorithm for live classification, can also take as input pre-labeled data for evaluation.

Timeline
by October 25 collect initial data set (single, multi user)
by November 8 labeling finished (including software development)
by November 22 heuristics developed (including live recognition), feature extraction done.
by December 6 accuracy #s for all algorithms, live classifiers implemented
by December 13 project presentation done

American Sign Language Recognition System with SIFT Descriptors
16-720 Project Proposal

Proposal

Vision-based gesture recognition shows promise for many human-computer interaction applications [1]. One such application is sign language translation, which can be used as an interactive educational tool or simply to help a hearing-impaired person communicate more effectively with someone who does not know sign language. For this project, the team intends to create an American Sign Language (ASL) recognition system which can classify sign gestures captured by a webcam in near real time. For the scope of this project the team will focus on the static signs for letters of the alphabet but will expand to dynamic word signs if time permits.

General Approach

The team will generate images to establish a training database for each ASL sign in the alphabet. Because of differences in hand shapes and sign techniques between subjects, it is important to extract features that are scaling and rotation invariant. The team will rely on Scale Invariant Feature Transform (SIFT) features since the method meets both invariance requirements and has been proven to work in recognition problems [2]. In addition, SIFT features are partially invariant to changes in illumination which should allow us to supplement our image database with images from the Purdue RVL-SLLL American Sign Language Database [3]. Once the features are extracted for the training images, the team will cluster the data. Images for testing will be taken in from a webcam for feature extraction and classification using a nearest neighbor classifier.

Additional Remarks

In Spring 2012, the current team (along with Matt Eicholtz) worked on this problem in 24-787 Artificial Intelligence and Machine Learning. A Matlab GUI was created to capture images and perform the live classification. It's our intention to reuse these parts of the project. Since the focus of the previous project was on machine learning, the implementation focused on the building of a neural network. The features were simply the threshold binary intensities of a 50x75 image. Since comprehensive computer vision components were absent in our previous implementation, we intend to focus this project on extracting more meaningful features and improving our recognition system.

1. Wu Y and Huang TS. Vision-Based Gesture Recognition: A Review. *Lect Notes Comput Sci*

1739, 1999.

2. Distinctive image features from scale-invariant keypointsDavid G. Lowepreprint, to appear International Journal of Computer Vision. 2003.

3. R. B. Wilbur and A. C. Kak, "Purdue RVL-SLLL American Sign Language Database," School of Electrical and Computer Engineering Technical Report, TR-06-12, 2006, Purdue University, W. Lafayette, IN 47906.

Out of Sight, Out of Hand: Visual Tracking for Manipulation

Uncertainty is unavoidable in manipulation due to perception errors and an imperfect model of the arm's kinematics. Nonprehensile actions, such as push-grasping and sweeping, are effective at manipulating objects under uncertainty. Imagine dragging a book to the edge of a table to more easily grasp it or using a quick sweep to clear space on a cluttered desk. These actions are currently executed as open-loop routines and, thus, depend on overly conservative assumptions to guarantee success. Visually tracking the object during manipulation would allow for less conservative planning and reduce the amount of uncertainty introduced by executing these actions.

Visual tracking is difficult during manipulation because the robot's arm and hand necessarily occlude much of the object. Our key insight is the structure of the problem aids tracking in two ways: (1) approximate arm and hand occlusions are known from the robot's forward kinematics and (2) a physical model of the object's motion provides a strong prior on its movement in the image. We will use knowledge of the arm kinematics and geometry to compensate for occlusions in a template tracking technique, such as Lucas-Kanade. Unlike model-based techniques, template tracking will allow us to successfully track unmodeled objects. The output of the template tracker will fused with our physics-based motion model using a Bayesian filter to smooth tracking over time and to improve tracking of heavily occluded objects.

The first step of our algorithm will utilize the kinematics of the arm to build an occlusion mask for the image. This mask will be used to remove occluded regions from consideration during template matching. A template tracking algorithm will compute a displacement vector describing the alignment of the template between successive image frames from unoccluded regions. In addition, a confidence in the displacement measurement will be computed based on the amount of occlusion, with high confidence indicating low occlusion.

Loutas et. al. [3] proposed using a Kalman filter to aid template tracking. Their system had no control over or prior knowledge of the movement of the tracked object, and thus used a constant acceleration model to forward-predict the template's displacement. This is a fairly uninformative model, and thus the authors only used the Kalman filter during periods of full occlusion. In contrast, Dellaert and Thorpe [1] used a Kalman filter during the entire process of tracking a car. In this case, an accurate motion model of the vehicle was available making the Kalman filter useful even if the entire template was visible in a frame. Our problem is similar because we have a realistic, physics-based model of the object's movement [2]. Because of this similarity, we propose using a Kalman filter even during periods of partial or no occlusion. Our physics model of the object's motion will serve as the prediction step and the output of the template tracker as the measurement step. We will use the confidence value returned by the template tracker to dynamically update the noise associated with the observation model.

We will implement this algorithm on HERB and evaluate the performance of the algorithm on sequences of images captured while sliding books on table. Ground truth pose estimates of the tracked object will be measured using visual fiducials affixed to the objects. If time permits, we will perform additional evaluation on the Autonomous Robotic Manipulation (ARM-S) robot while clearing clutter from the workspace by sweeping objects with the robot's hand.

1 Timeline

1. **Oct. 31:** Generate occlusion masks from the arm kinematics
2. **Oct. 31:** Implement pushing.
3. **Nov. 9:** Collect a dataset of HERB pushing books.
4. **Nov. 23:** Implement template tracking on planar objects.
5. **Nov. 30:** Generalize template tracking for non-planar objects.
6. **Dec. 13:** Validate the algorithm on HERB and/or the ARM-S robot.

References

[1] Frank Dellaert and Chuck Thorpe. Robust car tracking using kalman filtering and bayesian templates. In *Conference on Intelligent Transportation Systems*, 1997.

[2] Robert D. Howe and Mark R. Cutkosky. Practical Force-Motion Models for Sliding Manipulation. *International Journal of Robotics Research*, 1996.

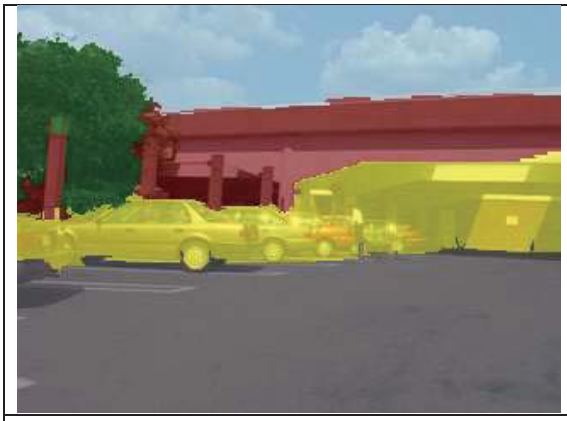
[3] E. Loutas, K. Diamantaras, and I. Pitas. Occlusion resistant object tracking. In *International Conference on Image Processing*, volume 2(7), pages 65–68, 2001.

Improve the bounding box detector with inference machines

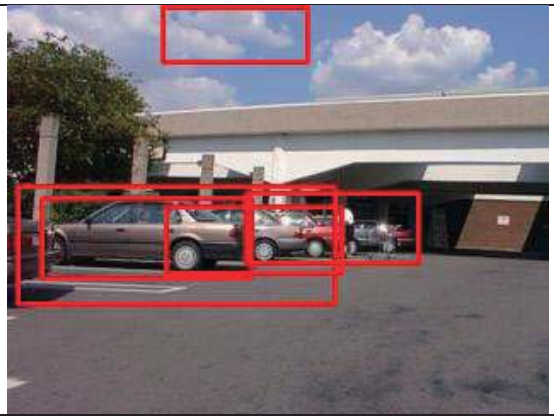
Object detection has been a challenging problem since the birth of computer vision. The most common approach is to apply trained detector on a given window of the image. Such approaches are often suboptimal because they ignore the contextual information within the window and from neighboring area. Recent researches have shown that utilizing contextual information is an effective method for developing per pixel semantic classification of scene [1].

We propose to use semantic segmentation output from an inference machine to provide contextual information to object detectors. We will investigate the performance of a raw object detector evaluated at various thresholds of its confidence, as well as performance of the detectors reweighted with the output of the inference machine.

We will also use the object detections as weighted features for training the inference machine and investigate performance improvements.



Inference Machine output



Detector output

Fig 1: The left image shows the output from an inference machine. On the right side is the output from running a car detector.

Reference

[1] Daniel Munoz and J. Andrew Bagnell and Martial Hebert, "Stacked Hierarchical Labeling," in *European Conference on Computer Vision (ECCV)*, 2010.