Artificial Intelligence for Human-Robot Interaction
Papers from the AAAI 2015 Fall Symposium

Anticipation of Touch Gestures to Improve Robot Reaction Time

Cody Narber, Wallace Lawson, J. Gregory Trafton
Naval Center for Applied Research in Artificial Intelligence
Washington, DC

Abstract

Nonverbal communication is a critical way for humans to relay information and can have many forms including hand gestures, touch, and facial expressions. Our work focuses on touch gestures. In typical systems the recognition process does not begin until after the communication has completed, which can create a delayed response from the robot. It may take time for the robot to plan the appropriate response to touch, which could delay the reaction time. We have trained an artificial neural network on features extracted from the Leap Motion Controller, and successfully performed early recognition of touch gestures with high accuracy.

Introduction

Touch gestures are those gestures where a user will be coming in contact with the robot, such as touching, slapping, pushing, pulling, grabbing, poking, etc. For systems in which a user will be interacting with the robot via a touch gesture, the gesture recognition process will begin after a touch has occurred (Ji et al. 2011; Jung 2014). In both papers, large scale skin-like touch sensors were added to a robot where support vector machines and bayesian classifiers were used respectively to classify the types of touch. However, in cases where path planning is needed, processing the sensor data after a touch has occurred will create a delay in robot interaction. In certain human-robot interactions such as firefighting this delay can be much more critical to the performance of the system.

To minimize this type of delay and establish a real-time interaction between the user and the robot, we propose a system that anticipates user interaction as soon as possible in order to recognize an interaction gesture prior to touch. By recognizing the intended touch gesture early, the robot can begin the process for movement planning early so that execution of movement can occur immediately after contact and control between the user and the robot can be seamless.

Current research on anticipation of human activity has focused on a variety of topics including object path prediction for catching (Kim, Shukla, and Billard 2014), human focus and engagement towards a robot (Durdu et al. 2011; Vaufreydaz, Johal, and Combe 2015), human activity prediction in a specific environment (Koppula and Saxena 2013), or early face expression detection (Hoai and De la Torre 2014)

Our research makes use of the Leap Motion Controller where several features are extracted to determine proximity for engagement and to train an artificial neural network (ANN) for touch gesture anticipation. The number of image frames used to perform classification is a fraction of the total number of frames in the full interaction movement sequence which ensures that recognition will occur prior to touch. Current research using the Leap Motion has focused on recognizing non-contact hand gestures like pointing or sign language and not interaction gestures as done in our approach (Potter, Araullo, and Carter 2013; Marin, Dominio, and Zanuttigh 2014).

Data Collection

Data is collected from a Leap Motion Sensor, which is a compact (76 mm) stereo NIR camera, that is mounted on the backside of a humanoid robot just above waist height and angled upwards. In our research, we are processing the NIR images directly rather than using the extracted hand information provided by the Leap application programming interface (API). This done because the API may not recognize a hand shape early enough due to the hand's orientation or other factors. Therefore, the API may not provide any relevant data in the timeframe that we need.

The stereo NIR images that the Leap collects are preprocessed to enhance edges and minimize noise. An adaptive thresholding algorithm is then used to detect the hand/arm blobs. Convex hulls are created around the detected blobs using the QuickHull algorithm. Movement vectors are constructed by matching points within the current convex hull to edges in the subsequent detected convex hull. A histogram of angles is constructed containing 8 bins where each bin relates to a range of angles. The histogram is constructed by adding the magnitudes of the movement vectors to its corresponding angle bin. Figure 1 shows the preprocessed image, followed by the identified hand/arm blob. The rightmost image in Figure 1 shows the sequential convex hulls and mapped vectors that are used to construct the angle histogram.

The ratio of blob size to convex hull size estimates rough-

Figure 1: A. Preprocessed B. Blob Detection C. Vectors

ness for use as another feature. The number of pixels between the convex hulls in the two stereo images represents our approximate depth measure. Data was collected from several frames where an object was held at a specific distance from the sensor; comparing the predicted and actual distances, we have a 95% confidence interval of $[0.832, 0.902]$ percentage of alignment between the estimated and true values.

These measures are combined to create feature vector for a single frame. We gather 10 sequential frames in total and average together pairs of frames to construct a total of 5 separate vectors. These vectors are concatenated to create a single feature vector in 60 dimensional space, which is then normalized and used by our ANN. The number of features and frames to use were determined experimentally using both cross-validation and a separately collected test set. Our ANN is fully connected with a single hidden layer containing 60 nodes. Each node within the network uses a symmetric sigmoid activation function, and is trained using resilient backpropagation.

Experimental Results

For our training and testing purposes, we are assuming that the robot is a nozzle operator of a firefighting unit and the human supervisor is providing instruction on how to proceed. We also assume that the supervisor is either behind or to the side of the robot.

We examine 9 types of interaction differentiating them by urgency, standing position, and interaction type. Each of these differentiating factors can take on two meaningful values, and an additional value where the interaction is not well-defined. The urgency can be non-urgent or urgent. The position can be assigned as either to the side or back of the robot. The interaction type is used for designating what type of gesture the operator is performing, either pulling the robot back or pushing the robot forward.

Classification can be performed on 9 separate classes where they are designated as SidePush, UrgentSidePush, SidePull, UrgentSidePull, BackPush, UrgentBackPush, BackPull, UrgentBackPull, and NonIntegration. The NonIntegration class is used to denote movement in proximity of the robot where direct interaction is not happening such as walking past the robot, reaching or grabbing for other proximal objects. The NonIntegration class catches all of the movements that have at least one differentiating factor that is not well-defined. For our training set we have gathered 550 samples, with approximately 120 samples for the NonIntegration class, and 50 samples for each of the other classes. Using 10-fold cross validation to recognize all 9 classes separately, we have an overall accuracy of 82.2%, but we get improved results when we run recognition separately on each of the 3 differentiating factors: urgency, position, and movement type.

	Neither	Non-Urgent	Urgent	Acc
Urgency (R)	0.913	0.862	0.929	0.902
Urgency (P)	0.967	0.876	0.889	
	Neither	Side	Behind	Acc
Position (R)	0.906	0.838	0.943	0.891
Position (P)	0.920	0.894	0.872	
	Neither	Pulling	Pushing	Acc
Movement (R)	0.898	0.973	0.953	0.952
Movement (P)	0.983	0.932	0.965	

Table 1: (R)ecall, (P)recision, and Overall (Acc)uracy

In Table 1, we show the results classifying each differentiating factor using 10-fold cross validation. We can see that our technique works very well for recognizing the type of movement. This shows that our technique is well-balanced at separating out similar classes, but performs even better when the 3 dimensional output vector is examined to isolate the factors of the classes. The output vector can be used by our robot control unit to make informed decisions about motion planning: speed, operator location, and target direction; thus allowing a level of planning to occur even when one of these factors is known.

On average, our algorithm finishes recognition after 52.3% of the full movement has been completed, with urgent gestures completing a larger percentage of movement (73.8%) due to the nature of the urgent gestures taking less time to perform. However, recognition occurs after only 39.5% of the total movement for non-urgent gestures. This equates to recognition occurring between 62 to 269 ms prior to touch.

Discussion

We have demonstrated the ability to recognize different classes of touch gestures prior to the moment of touch, which can reduce robot response time to achieve a more fluid interaction. Our future work will incorporate an additional Leap sensor on the other side of the robot and introduce other classes to check for turn and poke commands. We will also incorporate feedback from tactile sensors for terminating the touch gesture signal.

We will then verify experimentally that our touch anticipation system performs faster than our existing system, which uses only the tactile sensors as a synthetic skin. We will then perform a set of user studies to determine a how natural the interaction feels between these two systems.

Acknowledgment

This work was supported by the Office of Naval Research (GT). The views and conclusions contained in this paper do not represent the official policies of the U.S. Navy.

References

Durdu, A.; Erkmen, I.; Erkmen, A. M.; and Yilmaz, A. 2011. Morphing estimated human intention via human-robot interactions. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1.

Hoai, M., and De la Torre, F. 2014. Max-margin early event detectors. *International Journal of Computer Vision* 107(2):191–202.

Ji, Z.; Amirabdollahian, F.; Polani, D.; and Dautenhahn, K. 2011. Histogram based classification of tactile patterns on periodically distributed skin sensors for a humanoid robot. In *RO-MAN, 2011 IEEE*, 433–440. IEEE.

Jung, M. M. 2014. Towards social touch intelligence: developing a robust system for automatic touch recognition. In *Proceedings of the 16th International Conference on Multimodal Interaction*, 344–348. ACM.

Kim, S.; Shukla, A.; and Billard, A. 2014. Catching objects in flight. *Robotics, IEEE Transactions on* 30(5):1049–1065.

Koppula, H. S., and Saxena, A. 2013. Anticipating human activities for reactive robotic response. In *IROS*, 2071.

2013. Leap motion controller. http://www.leapmotion.com/. Accessed: 2015-06-30.

Marin, G.; Dominio, F.; and Zanuttigh, P. 2014. Hand gesture recognition with leap motion and kinect devices. In *Image Processing (ICIP), 2014 IEEE International Conference on*, 1565–1569.

Potter, L. E.; Araullo, J.; and Carter, L. 2013. The leap motion controller: a view on sign language. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, 175–178. ACM.

Vaufreydaz, D.; Johal, W.; and Combe, C. 2015. Starting engagement detection towards a companion robot using multimodal features. *Robotics and Autonomous Systems*.