Vision-based Contingency Detection

Jinhan Lee Jeffrey F. Kiser Aaron F. Bobick Andrea L. Thomaz

School of Interactive Computing
Georgia Institute of Technology
801 Atlantic Dr., Atlanta GA, 30302
{jinhlee, jfkiser, afb, athomaz}@cc.gatech.edu

ABSTRACT

We present a novel method for the visual detection of a contingent response by a human to the stimulus of a robot action. Contingency is defined as a change in an agent's behavior *within a specific time window* in direct response to a signal from another agent; detection of such responses is essential to assess the willingness and interest of a human in interacting with the robot. Using motion-based features to describe the possible contingent action, our approach assesses the visual self-similarity of video subsequences captured before the robot exhibits its signaling behavior and statistically models the typical graph-partitioning cost of separating an arbitrary subsequence of frames from the others. After the behavioral signal, the video is similarly analyzed and the cost of separating the after-signal frames from the before-signal sequences is computed; a lower than typical cost indicates likely contingent reaction. We present a preliminary study in which data were captured and analyzed for algorithmic performance.

Categories and Subject Descriptors

A.m [**Miscellaneous**]: Human-Robot Interaction—*Social Robots*; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Motion, Depth cues*

General Terms

Algorithms, Design, Experimentation

Keywords

Contingency Detection, Human Robot Interaction, Response Detection

1. INTRODUCTION

As robots migrate from controlled, constrained environments into the human world, many tasks will be performed through human-robot cooperation. To successfully execute these tasks, robots will need to identify willing interaction partners and to initiate interactions; the mere presence of humans is not sufficient to establish a task partner. Thus a required capability for such robots is the ability to detect whether a human is attending to or at least aware of the robot. To be effective such determination should be established quickly and through a natural action on the part of the robot. The robot, as an active agent, seeks an interactive partner by sending an interactive signal to an environment and checking presence of the response from other social agents.

Human developmental research has determined that as early as 10 months, infants use *contingency* to recognize social agents [1, 2]. Contingency is a change in an agent's behavior *within a specific time window* in direct response to a signal from another agent. In this paper we present a computational model of contingency detection, which we build in a supervised learning fashion with data collected from human subjects interacting with our upper-torso humanoid robot. We present a model that recognizes contingent events with 79% accuracy, and discuss the implications of this work and its future directions.

2. RELATED WORK

Contingency detection has been shown to be indispensable functionality to a social robot; it helps the robot to learn structured functional or social environments [6, 7, 9], and understand the social interaction [5, 8, 10]. The contingency detection can happen at discrete turn-based situation [6, 8] and at continuous interaction [3, 4]. For the former case it requires to define discrete states and learns functional mappings of state transition with a triggered signal. For the latter case, on the other hand, it needs a mechanism to determine the presence of the change as response to the signal. In this paper, we are interested in addressing the problem of the latter case.

Movellan [3] broke down the problem of contingency detection into two smaller problems: response detection and timing interpretation. He developed a model of behavior that optimally queried the environment with an auditory signal and detected a responsive agent with a simple binary sound sensor. Detection processing was limited to determining the source of the response—either the robot itself, a social agent, or background noise—to the robot's actuator signal. Movellan primary focus was on the timing constraints of the contingency problem. Similarly, Gold and Scasselati [4] focus on learning effective timing windows for contingency detection with an auditory signal.

Our work, in comparison, is concerned with the response detection aspect of the contingency problem. Real-world human responses, of course, include more than just audio signals potentially including visual and/or tactile signals. An ideal contingency response detector should be able to accept a variety of sensory cues.

The approach we design can be employed for any of these channels or their combination; in this paper, as our main contribution, we demonstrate the method applied to visually derived information. To be a contingent response to a given visual signal, something different should happen after the signal. This is why simple detection of abrupt change along the signal fails to detect contingent response, since abrupt change does not mean *different*. Our approach detects difference rather than abrupt change.

The response detection problem with visual cues differs from an *action recognition problem* in that responses cannot be formulated as predefined actions. For example, a human could respond to a robotic greeting by approaching the robot, speaking to the robot, stopping a previous action and looking at the robot, or by doing a number of other things. What makes an action a contingent response is that there is change in behavior at the appropriate time, not that there is some specific recognized action. The response detection problem is also different from the *saliency detection problem*. Saliency is defined as a region's difference from its spatial [12, 13] and /or temporal [11] neighbors. Saliency tends to be calculated only from immediate neighbors, which is too restrictive for responses that have a time duration of more than a fraction of a second.

Some recent results in the computer vision community on the detection of abnormal or unusual events are related to the work presented here. Yu and Moon [16] compare every new observation to all previous observations using Principal Component Analysis. Their underlying observation representation is suited for detecting changes in motion direction rather than overall behavior, and the method of determining change is quite different that the metric we will derive, but the goal of detecting a change is similar to our task. Zhong et al. [17] splits a video into segments and clusters them using a spectral graph co-clustering method. A video segment is defined to be unusual if that segment is different from a majority of rest of the segments. Our work is distinguished from theirs by focusing on observations before a robot's interaction signal to observations after that signal. Finally, Boiman and Irani [18] builds a rich database of imagery to explain current segments of input video. Their notion of surprising or irregular video is that which cannot be synthesized from a composition of previous observed imagery. This is distinct form our approach in that explicitly measure direct similarity of behavior across a signal boundary.

3. APPROACH

We formulate the contingency detection problem as one of detecting a response — a behavior change — within a specified time interval after an interactive signal given by the robot. Our robot initiates an interaction with a person by sending interactive signals, such as waving or beckoning, and then detects the presence or absence of human response for a certain length of time after the signal. To detect human response, we look for a significant perturbation in the human behavior by modeling that behavior before and after the signal and looking for statistical differences. We assume that any observed perturbation happening within the allowed time interval after the robot's interaction signal is the human's response.

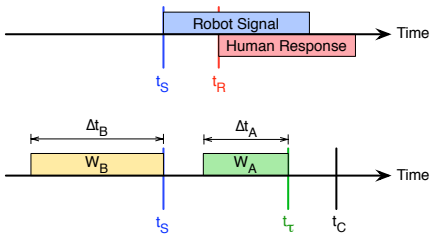


Figure 1: A Robot's Interactive Signal and a Human's Response. At top, the beginning of the robot signal, t_S, and the human response, t_R. The bottom graph shows the time windows over which we model human behavior before and after the signal.

Our approach to modeling human behavior is premised on two key assumptions. First, we assume that the human behavior prior to the interaction attempt is not erratic. This means that the background behavior consists of simple periodic or occasionally repeated motions. Intuitively, if the human behavior is continually changing abruptly many such changes could be confused for a response to the robot when in actuality are simply part of the ongoing behavior. Our system will fail to notice a contingent response in this case.

Second we assume that the cues measured by the detectors are sufficiently discriminative to adequately model the human's behavior and differentiate between any background behavior before the signal and a perturbation in behavior after the signal. In the example developed in this paper, we employ the spatial pattern of optic flow to characterize behavior. Therefore we assume that all relevant perturbations in human behavior which could be contingent responses are detectable as differences in optical flow. Other cues can easily be added to our behavior model, we demonstrate this by showing results of a multi-sensor experiment that uses both optical flow and depth cloud information from a Kinect senor.

Our formulation of the contingency problem is illustrated in Figure 1, where:

- t_S: time at which the robot initiates an interactive signal
- t_R: time at which the human initiates a contingent response to the robot's signal
- Δt_B: length of time over which human behavior is modeled prior to t_S
- Δt_A: length of time over which human behavior is modeled after t_S
- t_C: cutoff time when we stop looking for perturbations
- t_τ: current timestamp, ranging $t_S + \Delta t_A$ to t_C
- W_B: fixed window of time $[t_S - \Delta t_B, t_S]$, over which behavior before t_S is modeled
- W_A: sliding window of time $[t_\tau - \Delta t_A, t_\tau]$, over which behavior after t_S is modeled to represent the behavior at time t_τ

4. IMPLEMENTATION

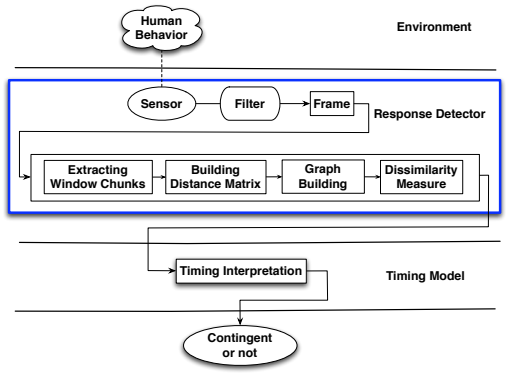


Figure 2: Our Contingency Detection Framework (Our main focus in this paper is the response detector in the blue box.)

Figure 2 shows our complete contingency detector framework. One or more sensor modules observe the human behavior and generate data; this data passes through a sensor-specific filter to remove background data which was not generated by the human agent in question.

The filtered data is stored in a feature vector, which we call a frame. One frame is sent to the response detector for each instant in time. We implemented a sensor module based on optical flow, which will be described in Section 4.1. If sensor data are represented as frames and a distance matrix over frames is defined, our framework can leverage that sensor. Even multiple sensor data can be merged and integrated into our framework. The data can be fused at least at three different stages: 1) at the frame calculation stage, 2) at the distance matrix calculation stage, and 3) at the dissimilarity calculation stage. In this paper, we present an example of the second case.

The response detector accepts frames from the sensor modules and is triggered when the robot begins to initiate an interaction signal at time t_S. For each instant in time from $t_S + \Delta t_A$ to t_C, the detector continues to receive frames but also outputs a scalar dissimilarity score; high values indicate that the behavior modeled during W_B — the interval before the signal — is different than the behavior modeled during W_A, the interval after the signal. These dissimilarity scores are modulated by a timing model, which applies a weight to each score based on the time elapsed, $t_\tau - t_S$; each weighted dissimilarity score is the contingency score. Finally, the response is declared contingent if any contingency score between $t_S + \Delta t_A$ and t_C exceeds a threshold.

As the response detector accumulates frames, it keeps only enough to fill the past Δt_B amount of time and discards frames which are older. When triggered at time t_S, the detector locks in the frames it already has as the frames for W_B and transitions to accumulating frames to fill W_A until time $t_S + \Delta t_A$. Once frames exist over all of W_B and W_A, the dimension of those frames is reduced using matrix factorization (Section 4.2). The detector groups them into overlapping clips of consecutive frames (Section 4.3). A distance matrix is calculated between the clips (Section 4.4). This distance matrix is converted to a graph (Section 4.5). Finally, we calculate a statistical difference between graph nodes representing clips from W_A and graph nodes representing clips from W_B; the scalar output of this process is the dissimilarity score (Section 4.6).

(a) An original image. A human subject is playing with a puzzle

(b) The color coded motion image. Two color blobs come from both the subject and a non-subject

(c) The corresponding disparity image

(d) The color coded motion image after eliminating background motion. Only motion from a subject remains

Figure 3: Motion Extraction and Background Motion Elimination

4.1 Sensors and Filters

Our framework allows for many types of human behavior sensors. A sensor module must be able to produce a feature vector which is filtered to only include information about the human agent in question in order to be used in our contingency detector. Our optical flow motion sensor uses depth information to filter background motion data.

4.1.1 Motion Extraction

We use a stereo camera to capture image data over time, which we then use to estimate motion by calculating the region-based dense optical flow as developed by Werlberger et al. [19]. This method minimizes an objective function which enforces consistency of pixel values along the flow vector as well as piecewise smoothness of the flow field; it uses a dense Gaussian pyramid with image warping. Figure 3(b) shows the resulting motion vector map for each pixel of a pair of images, one of which is 3(a).

4.1.2 Depth-Based Background Motion Elimination

As shown in the upper right of Figure 3(b), a blob of motion has been generated by a person walking in the background of the scene as well as by the human agent in the foreground. After extracting motion from the entire frame, we segment the motion image and eliminate background motion using the depth information from the stereo camera.

For motion segmentation, we adapt the graph-based color image segmentation method introduced by Felzenszwalb and Huttenlocher [14]. In their framework, an image is represented as a graph in which nodes are pixels and edges are defined by connecting neighboring pixels. A weight assigned to each edge indicates a distance between the corresponding nodes, where the distance metric is the difference of pixel value intensity. Segmentation is done by iteratively merging nodes. Please refer to [14] for details on merging. After ob-

taining motion segments, we remove those segments which have small motion magnitude as well as those which have a large depth. The filtered result is shown in Figure 3(d).

4.2 Frame Dimension Reduction

Based on our assumption that the background behavior is self-similar, the human agent's background behavior and response (if a response exists) can be embedded onto a low-dimensional subspace. We compare three different approaches: no dimensionality reduction, Principal Component Analysis (PCA) and Non-negative Matrix Factorization (NMF).

Both PCA and NMF generate a set of basis vectors and coefficients for the original frame data. Let $X \in \mathbb{R}^{D \times N}$ be the original high-dimensional, raw data of all frames in W_B and W_A combined, where D is the dimension of the data in each frame and N is the number of frames in W_B and W_A combined. The dimensionally reduced set of coefficients calculated for each frame become the elements of a new, smaller frame. Let $H \in \mathbb{R}^{R \times N}$ be all of the smaller frames, where R is the number of reduced dimensions; in the case of no dimension reduction, H is the original data X and R is equal to D.

4.2.1 PCA Dimension Reduction

Under the assumption that data comes from a unimodal Gaussian distribution and that the dimensions of largest variance of the data encode most of the information while removing noise, PCA finds a set of independent basis vectors whose directions maximize the variance of the frame data. The frame data is approximated a linear combination of these basis vectors. Let $\overline{X}$ represent the mean of X, and define $\widetilde{X} = X - \overline{X}$. Then, if we let U be the eigenvectors of $\widetilde{X}\widetilde{X}^T$ corresponding to the R largest eigenvalues, we compute the reduced frames as:

$$H = U^T \widetilde{X}$$

4.2.2 NMF Dimension Reduction

NMF [15] decomposes the frame data into a linear combination of basis vectors where all elements and coefficients of the basis vectors are non-negative. NMF has been used for part-based decomposition of an image. Since pixel values in a motion image are nonnegative and motion regions of the image can be viewed as parts, NMF is an appropriate choice for reducing the dimensionality of the motion data. Let $W \in \mathbb{R}^{D \times R}$ be a nonnegative basis of X, and $H \in \mathbb{R}^{R \times N}$ be coefficients of W. W and H are initialized to random nonnegative matrices. The following iterative process will converge to the set of reduced frames, H:

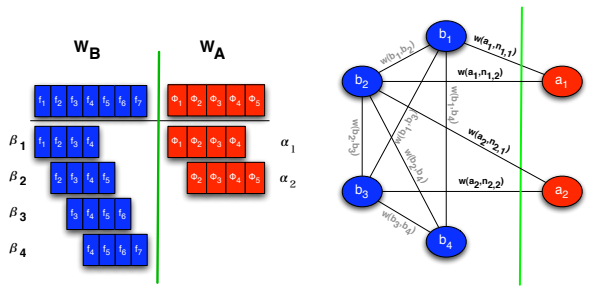
$$W \leftarrow W \otimes (XH^T) \oslash (WHH^T)$$

$$H \leftarrow H \otimes (W^T X) \oslash (W^T W H)$$

where $\otimes$ and $\oslash$ are element-wise matrix multiplication and division respectively. The details of the algorithm as well as discussions of its convergence are provided in [15].

4.3 Breaking Windows Into Clips of Frames

In order to adequately model temporal events, we compute distances between groups of consecutive reduced frames, which we call clips. Figure 4(a) shows how we make clips from reduced frames in W_B and W_A. We let f refer to a



(a) Grouping Frames to Clips with $\sigma = 4$ (b) Building a Dissimilarity Graph with $\kappa = 2$

Figure 4: Grouping into Clips and Building a Dissimilarity Graph

reduced frame in W_B, and ϕ refer to a reduced frame in W_A. We refer to the i^{th} clip of reduced frames in W_B as β_i, the j^{th} clip of reduced frames in W_A as α_j, the total number of clips in W_B as n, and the total number of clips in W_A as m. Given a clip size σ which is the number of reduced frames per clip, we define clips according to Equations 1 and 2:

$$\beta_i = \{f_k \mid k = i, \ldots, i + \sigma - 1\}, \quad i = 1, \ldots, n \qquad (1)$$

$$\alpha_j = \{\phi_k \mid k = j, \ldots, j + \sigma - 1\}, \quad i = 1, \ldots, m \qquad (2)$$

Notice that consecutive clips are overlapping, separated by only one frame, and that no clip is defined with frames outside of W_B and W_A or with frames from both W_B and W_A.

4.4 Calculating the Distance Matrix

The distance matrix $D \in \mathbb{R}^{(n+m)\times(n+m)}$ represents the distances between clips. Let $\beta'_i \in \mathbb{R}^{\sigma R}$ be a single vector made by concatenating the frames of β_i, and $\alpha'_j \in \mathbb{R}^{\sigma R}$ be a single vector made by concatenating the frames of α_j. Let $H' \in \mathbb{R}^{\sigma R \times (n+m)}$ be the matrix of all the β'_i's and α'_j's. The distance matrix is defined as:

$$D_{u,v} = \sum_{i=1}^{\sigma R} (H'_{i,u} - H'_{i,v})^2 \quad u, v = 1, \ldots, n+m \qquad (3)$$

4.5 Building the Dissimilarity Graph

We estimate the dissimilarity between the behavior in W_B and the behavior in W_A using the distance matrix D calculated in Equation 3 by constructing what we term a dissimilarity graph.

When the human has not generated a contingent response to the robot's signal, both β and α clips describe the same aspect of the self-similar behavior. In particular, an α clip is as likely to be similar to one β clip as to its closest neighbors. This should not happen in contingent cases, however, because when the human's background behavior has been disrupted by a contingent response, the β and α clips describe different behaviors.

Based on this intuition, we construct a weighted edge graph $G = (V, E)$. The nodes V represent clips from W_B and W_A. Let $B = \{b_i \mid i = 1, \ldots, n\}$ and $A = \{a_j \mid j = 1, \ldots, m\}$, where b_i is the node corresponding to the clip β_i in W_B and node a_j corresponds to α_j; then $V = A \cup B$.

To determine the edges E of the dissimilarity graph, we use the following properties:

1. Nodes in B are fully connected to each other
2. Nodes in A are never connected to each other
3. Nodes in A are only connected to the κ nearest nodes in B

Edge weights represent distances between clips. We denote the edge weight between connected nodes p and q as $w(p,q)$, and use the distance matrix D to determine the edge weight values. An example graph is shown in Figure 4(b).

To use the graph G to calculate the dissimilarity between A and B, we construct a probability distribution function for each node in B. Let $K_i(w)$ be the probability distribution function associated with node b_i.

We use a kernel density approximation of the probability distribution to calculate $K_i(w)$. We assume edge weights are independent and identically distributed samples.

The kernel used is a Gaussian function with bandwidth h, which is adaptively estimated based on sample variance. Please refer to [20] for details. We define $K_i(w)$ as:

$$K_i(w) = \frac{1}{h(n-1)} \sum_{\ell=1,\ell\neq i}^{n} \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{(w(b_\ell,b_i)-w)^2}{2h^2}\right)} \quad (4)$$

Let $C_i(w)$ be the cumulative distribution function of $K_i(w)$, which represents the percentage of nodes in B that are closer than w to b_i. Let $n_{j,k}$ represent the k^{th} nearest neighbor of a_j. We use this to calculate the dissimilarity between A and B, represented as d_τ (where the current time is t_τ), which is a measure of the dissimilarity between the behavior observed during W_B and the behavior observed during W_A:

$$d_\tau = \frac{1}{m\kappa} \sum_{j=1}^{m} \sum_{k=1}^{\kappa} C_k(w(a_j, n_{j,k})) \quad (5)$$

4.6 Determining Contingency

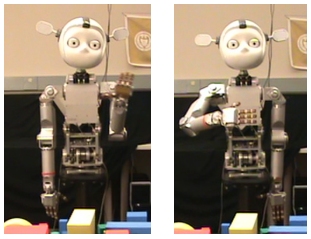
As the current time t_τ increases from $t_S + \Delta t_A$ to t_C, many dissimilarity scores d_τ are computed. To compute the contingency scores, we weight the dissimilarity scores based on a timing model. We used a uniform distribution for our timing model because we did not collect enough data to adequately model human response times (see Section 7.2). To decide that a human has reacted contingently to a robot signal, we threshold the contingency scores and consider cases where the threshold is surpassed as contingent.

5. DATA COLLECTION

To build a computational model of social contingency detection we use a supervised learning approach. To establish the appropriate parameters of our model we require a realistic dataset of contingent and non-contingent responses to a robot issuing different interactive signals. We designed a study to collect this data from human subjects.

Our study consisted of two groups: contingent and non-contingent. We ran a within groups study with five people, each of whom was asked to conduct a short interaction with our upper-torso humanoid robot, Simon. People were given one of three background tasks: playing with toy blocks, playing with a Rubik's cube puzzle, or talking on a cell phone. They were given the following instructions prior to starting the interaction:

- Contingent Group: While you are doing task X, Simon will try to get your attention, please respond to the robot in any way that you feel is appropriate.
- Non-Contingent Group: While you are doing task X, Simon will try to get your attention, please ignore this and do not respond.



(a) Waving (b) Beckoning

Figure 5: Two robot signals used in data collection

During the interaction Simon sent one of two interaction signals to participants: *Waving* and *Beckoning* as shown in Figure 5. The Waving and Beckoning behaviors took about 9-10 seconds.

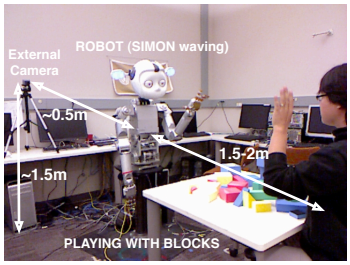


Figure 6: Environment setup for data collection

An interaction was considered terminated 10 seconds after the robot stopped moving or the human stopped responding, whichever came later. The participant was then given instructions for the next interaction (i.e., given a background task and the contingency-based instructions). Each person completed 8 interactions with the robot.

From these interactions we logged time-stamped video data, which allows us to link the time of the robot's signal with the video data. As an vision sensor, we used an external stereo camera [1]. We collected images and disparity values. As shown in Figure 6 it was placed behind the robot at shoulder height such that camera maintains a good view of the human subjects (at distance of 1-3 meters from the robot) and robot self-motion is not in the camera view. During the study, people were located close enough to the robot that we could assume that any motions detected in the near field would come only from the participant; we remove background noise based on depth of motion.

As a summary of the data, we collected 43 test cases: 20 contingent and 23 non-contingent. These 43 cases included different robot signal events (22 waves, 21 beckoning gestures). The cases also varied by background task (17 playing

[1]The stereo camera used is a Videre STOC which has a baseline of 9cm, 640 by 480 image resolution and runs at 25-30 frames per seconds.

with blocks, 16 doing puzzle, and 10 talking on phone).[2]

As our proof of concept of extension of our detector to handle multiple sensors, we collected 24 additional test cases from 3 subjects using the same experimental protocol: 12 contingent and 12 non-contingent using same robot gestures. The cases varied by background task (12 playing with blocks, 12 playing with legos). With a Kinect sensor, we logged not only images from the camera sensor, but also 3D point clouds obtained from the depth sensor. [3]

6. THE CONTINGENCY MODEL

6.1 Setup

We used the collected video data and robot action logs to train our contingency detection model. To build a contingency model, we run each video and calculate contingency scores for frames after the robot action signal and found a contingency threshold which produces the highest classification accuracy. Due to the small size of our dataset, we performed leave-one-out cross validation on the best contingency detection model. We investigated the effect of changing two parameters in our contingency detector: the dimension reduction technique, and the connectivity of the graph. Throughout the experiments, we set Δt_B to 9 seconds, Δt_A to 2.5 seconds, t_C to 15 seconds; these values were found empirically and we considered them to be constants. We tried three dimension reduction methods: no dimension reduction, PCA , and NMF (see Sec. 4.2). For PCA and NMF, we set the number of reduced dimensions, R, to 20. We also varied the number of nearest neighbors, κ, used in determining the connectivity of our dissimilarity graph. We varied κ between 5%, 10%, 20%, 50%, and 100% of the number of clips in W_A.

To prove the feasibility of extending our detector to handle multiple sensors, we built contingency models with only one sensor (image only) and with multiple sensors (image and depth). For the multiple sensor case, we processed each sensor data independently up to distance matrix calculation and merged those matrices before building a graph structure.

We implemented the contingency detection with GPU programming, particularly for the optical flow calculation and NMF decomposition. It runs at 10 to 15Hz, which we believe is fast enough for real-time Human-Robot Interaction.

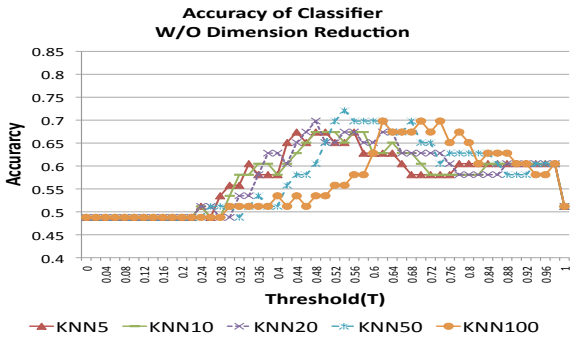
6.2 Results

Figures 7(a), 7(b), and 7(c) provide the results of a variety of experiments in which the dimensionality reduction method, the number of nearest neighbors, and the threshold used for detecting contingent responses were varied in the single sensor case. The measure of performance was percent accuracy which was defined to be correctly determining whether a contingent response occurred or not.

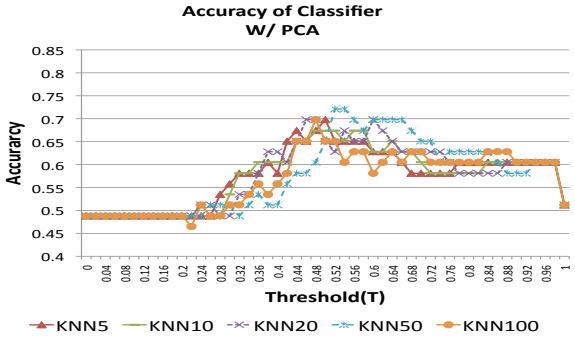
The first observation we make is that the classifiers built with the NMF-based dimension reduction performed better than those built with PCA or no dimension reduction. As the basis vectors of NMF directly model regions of motion, the NMF is suited better to represent motion and as

[2]We initially collected data from 7 subjects but had to remove data from two subjects due to data logging errors. This is how we ended up with the unequal number of background tasks across the dataset.

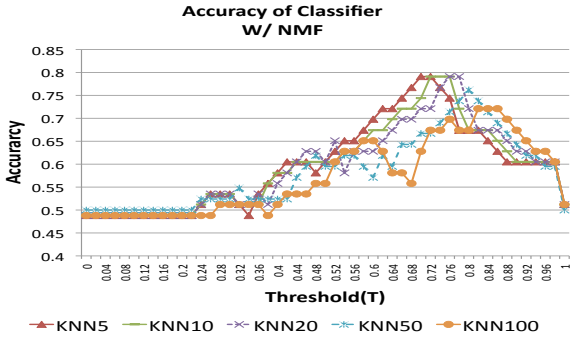
[3]We placed a Kinect sensor at the same location as the Videre STOC in Fig. 6.



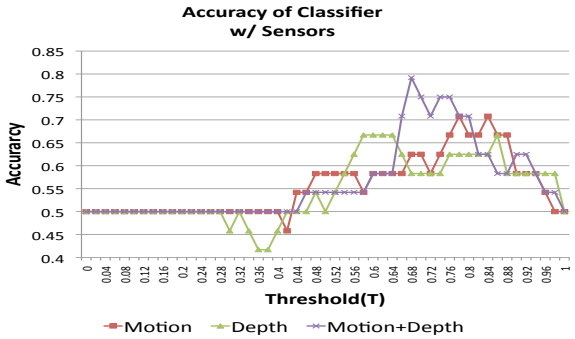
(a) Accuracy of contingency detectors with no dimension reduction. The best accuracy is 72.1% at $T = 0.54$ with $\kappa = 50\%$.



(b) Accuracy of contingency detectors with PCA dimension reduction. The best accuracy is 72.1% at $T = 0.52$ to $T = 0.54$ with $\kappa = 50\%$.



(c) Accuracy of contingency detectors with NMF dimension reduction. The best accuracy is 79.1% at $T = 0.70$ with $\kappa = 5\%$ to $\kappa = 20\%$.



(d) Accuracy of contingency detectors with multiple sensors. The best accuracy is 79.2% at $T = 0.68$ with $\kappa = 20\%$.

Figure 7: Three robot signals used in data collection

expected provided a better cue from which to determine contingent responses.

The best classifier performance was achieved using the NMF dimension reduction method and a connectivity of $\kappa = 5\%$, $\kappa = 10\%$, or $\kappa = 20\%$; for the different κ a slightly different threshold T was required to achive this level of performance. The accuracy obtained with these classifiers is 79%, where the binary threshold T is between 0.70 and 0.78 as shown in Figure 7(c). To check that we were not overfiting to the training data we performed leave-one-out cross validation where all but one of the data sequences was used to establish the best parameter settings, and then applied to the remaining sequences; this test produced the same accuracy of 79%. The precision of the classifier is 1.0, which means that the false positive rate is 0; i.e.: all non-contingent cases were correctly classified. Among the 20 contingent test cases, 11 were correctly identified as being contingent.

The classifier built with NMF dimension reduction but with a connectivity of $\kappa = 100\%$ (fully connected) had a precision of 0.85, meaning that some non-contingent cases were classified as contingent. The reason for this is that making the dissimilarity graph fully connected requires the behavior after the signal to be similar to all of the behavior before the signal, not just some of the behavior before the signal, in order to be classified as non-contingent.

Upon examining the data, we saw two explanations for the false negative errors. First, responses from subjects were sometimes communicated with different cues such as audio and social rather than motion cues. We observed that some contingent subjects briefly shifted their gaze to look at the robot, but did so while continued their background behaviors. Additionally, occasionally the contingent response motion occurred in image regions where the on-going background behavior was performed. In these cases the background behavior masked the contingent response.

Figure 8 shows the timing of our contingent data. The average human response delays after robot waving and beckoning are 4.05 (s.d. 1.54) seconds and 5.73 (s.d. 3.61) seconds respectively. The average durations of the human responses to waving and beckoning are 3.64 (s.d. 2.56) seconds and 3.27 (s.d. 1.68) seconds respectively. As Figure 8 shows, human response does not appear to adhere to a Poisson distribution timing model, even within the same robot signal and human background task. However, the limited number of trials prohibits robustly estimating a true probability density over response delay.

Figure 7(d) shows results from our multi-sensor experiment, demonstrating the generality of our approach. We find that the two sensors together had slightly better performance than either on their own. The multi-sensor based contingency detector performed best with accuracy of 79.2%. We observed that the best threshold for multi-sensor based detector is located between the best thresholds for single-sensor detectors.

7. DISCUSSION

7.1 False Positive versus False Negative

Throughout the experiment, the classifier which achieves the best accuracy does not have false positives — cases which were not actually contingent but which our algorithm incorrectly labeled as being so. Instead, all errors are false negatives. The true cost of such failures is, of course, determined by the task cost of each type of error.

One possible strategy is to introduce a loss function and prior belief about the outcome of the interaction and minimize that function. Prior belief about the outcome of the interaction can be a learned model like timing interpretation, or can be built based on interaction over time. A loss function should depend on both the task and the task context. In the situation where the robot's task requires long interaction with human and a number of humans are present, the robot should require a high confidence of the human's response. In less crowded environments with shorter interactions, the robot can initiate interactions with lower confidence.

Another strategy is to postpone making a decision for or against contingency when the contingency score is neither low nor high. Rather than making a single binary decision, *contingent* or *not contingent*, a classifier could allow a third state, *more queries needed*. When the contingency score is smaller than a lower boundary or larger than a higher boundary, the robot could make a contingency decision. When the score is between boundaries, the robot could do another signal/query to the human to gain confidence.

7.2 Human Response Delay

We observed relatively long amount of delay, about 5000 milliseconds on average, before the human responded to the robot's *visual* signal, when compared with the delay time (800 to 2400 milliseconds) modeled for *sound* signals [3]. We hypothesize that the difference in delay results from the time required for a person to determine the meaning of the behavior. This recognition delay partly comes from the slow speed of robot motions compared with human motions, which may serve to slow down the entire interaction. To test our hypothesis, we could change the speed of motions and/or augment them with speech to help the human understand the meaning of them faster and check whether those changes significantly reduce delay.

7.3 Sufficiency of the Motion Feature for Response Detection

Our method allows a variety of sensor modules to be used. The accuracy of our motion-only classifier demonstrates that motion features provide a significant amount of information in our experimental situation, and our multi-sensor experiment showed that this can be combined with a depth sensor to achieve better performance. However, there are scenarios where motion-based classifiers make mistakes. We observed at least one case where the subject was contingent to the robot, but did not alter their motion pattern; instead, they responded verbally. This suggests that a combination of sound and motion could improve accuracy.

8. CONCLUSION

We introduce a motion-based contingency detector that leverages visual motion features and the known time of the robot's interaction signal to estimate a statistical difference between motions before and after the robot's signal, and detect the presence or absence of a human response. One application of such a detector is finding a human partner to initiate task-related interaction with. The result of our experiment shows that statistical change in motion can indicate the presence or absence of human response. Our resulting model performs with 79% accuracy at detecting contingent responses, in cross validation testing.

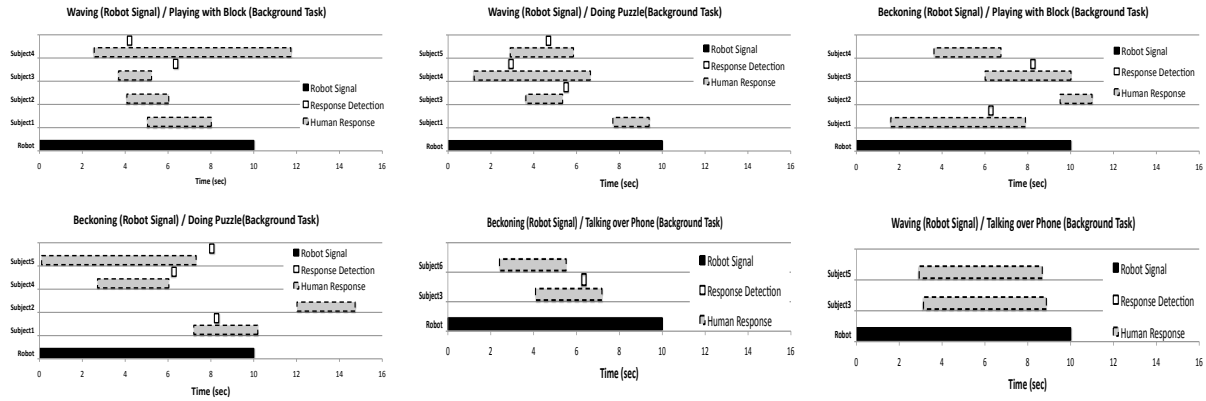


Figure 8: Timing of the robot's signal, the human's response, and the detector's recognition of a contingent behavior for our contingency cases. The detector shown uses NMF with $\kappa = 10\%$, with an accuracy of 0.791%. Cases without the detector trigger were not recognized as contingent cases by the detector, but note that none of the non-contingent cases was labeled contingent by the detector.

Based on our pilot data, which contained 43 test cases (20 contingency and 23 non-contingency) from 5 subjects, we hypothesize a relationship between response time delay, speed of the robot's interactive signal, and the human understanding of the robot's signal. We also hypothesize that a contingency detector built on our method but which uses both sound and motion features will improve accuracy results, particularly in reducing false negatives.

9. REFERENCES

[1] J.S. Watson, "Smiling, cooling, and "the game"", *MerrillPalmer Quarterly*, 1972.

[2] J.S. Watson, "The perception of contingency as a determinant of social responsiveness", *Origins of the Infant's Social Responsiveness*, 1979.

[3] J.R. Movellan, "An Infomax Controller for Real Time Detection of Social Contingency", *International Conference on Development and Learning (ICDL)*, 2005.

[4] K. Gold and B. Scassellati. "Learning acceptable windows of contingency", *Connection Science*, Vol. 18(2), p. 217-228, June 2006.

[5] G. Csibra and G. Gergely, "Social learning and social cognition: The case for pedagogy", *Processes of Changes in Brain and Cognitive Development. Attention and Performance*, Oxford University Press, 2006.

[6] H. Sumioka, Y. Yoshikawa, and M. Asada, "Reproducing Interaction Contingency Toward Open-Ended Development of Social Actions: Case Study on Joint Attention", *IEEE Transactions on Autonomous Mental Development*, 2010.

[7] J. Triesch, C. Teuscher, G. Deak, and E. Carlson, "Gaze following: why (not) learn it?", *Developmental Science*, 2006.

[8] K. Lohan, A. Vollmer, J. Fritsch, K. Rohlfing, B. Wrede, "Which ostensive stimuli can be used for a robot to detect and maintain tutoring situations?", *ACII Workshop*, 2009.

[9] N.J. Butko, J.R. Movellan, "Learning to Look.", *Proceedings of the 2010 IEEE International Conference on Development and Learning (ICDL)*, 2010.

[10] K. Pitsch, H. Kuzuoka, Y. Suzuki, L. Sussenbach, P. Luff, and C. Heath, "*IEEE International Symposium on Robot and Human Interactive Communication (ROMAN)*, 2009

[11] R.P. Wildes, "A measure of motion salience for surveillance applications", *International Conference on Image Processing (ICIP)*, 1998.

[12] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[13] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1998.

[14] P. Felzenszwalb and D. Huttenlocher, "Efficient Graph-Based Image Segmentation", *International Journal of Computer Vision (IJCV)*, September 2004.

[15] D.D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization", *In Advances in Neural Information Processing*, 2000.

[16] T. Yu, and Y. Moon, "Unsupervised Abnormal Behavior Detection for Real-time Surveillance Using Observed History", *IAPR Conference on Machine Vision Applications*, 2009.

[17] H. Zhong, J. Shi and M. Visontai, "Detecting Unusual Activity in Video", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[18] O. Boiman and M. Irani, "Detecting irregularities in images and in video", *International Conference on Computer Vision (ICCV)*, 2005.

[19] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bishof, "Anisotropic Huber-L1 Optical Flow", *Proceedings of the British Machine Vision Conference(BMVC)*, 2009.

[20] A. W. Bowman, and A. Azzalini, "Applied Smoothing Techniques for Data Analysis", *New York: Oxford University Press*, 1997.