Presented to the 1995 International Conference on Intelligent Robots and Systems, Pittsburgh, PA, August 5-9, 1995

Using Sensor Fusion and Contextual Information to Perform Event Detection during a Phase-Based Manipulation Task

Marc R. Tremblay and Mark R. Cutkosky

Center for Design Research
Stanford University
560 Panama St.
Stanford, CA 94305

Abstract

In this paper we present an approach to event detection during a dextrous manipulation task. The approach utilizes a combination of tactile sensors as well as contextual information. The manipulation task is decomposed into distinct phases, each of which is associated with a limited number of feasible events such as making or breaking contact, slipping, etc. A set of context-based and sensor-based features is associated with each possible event for each type of manipulation phase. The goal is to detect events as reliably and as rapidly as possible. At any time during a task, each possible event is assigned a confidence value between 0 and 1. This indicates how confident the detection scheme is that a given event could be occurring at that instant. A high-level controller can then make use of this information to determine when to switch to a different manipulation phase.

1. Introduction

Dextrous manipulation with a robotic hand is a complex process interspersed with events, control discontinuities and phase changes. For example, as the fingers close upon an object they are driven using position control, but when manipulating an object they are driven to maintain control of internal forces. In this example, the sensation of contact is an event that signals the transition from one control regime to the next.

Recently, there have been several efforts in the robotics literature to develop control systems for dextrous manipulation that consist of a series of phases, such as independent finger motion, coordinated motion, object impedance control, etc. [Cut93]. Each phase contains its own control law, controller gains, trajectories, and so forth. The transition from one phase to another is triggered by events which can be either expected or unexpected. A high-level controller then selects an appropriate subsequent phase based on the detected event. Using phase-based control, a complex manipulation task can be broken into distinct phases, making the manipulation process more tractable. However, robust event detection is clearly essential in a phase-based control approach.

When the signals from several types of sensors must be integrated, we are faced with the problem of data fusion. In robotics there has been much research done on data fusion but most of it has concentrated on vision rather than tactile sensing. A variety of approaches have been presented in the literature, including Bayesian methods [Bec92], Dempster-Shafer Theory [Gar81], Artificial Neural Networks [Mar89] and Team Decision Theory [Dur88] to name a few.

While much work has been done on developing tactile sensors that detect specific events, little has been done on developing strategies to detect a variety of events and/or properties using multiple sensors. One exception is the work done by Eberman and Salisbury [Ebe94] who examined the signatures obtained from fingertip force/torque sensors during manipulation. Their research looked at labeling some simple events without context using a combination of signal processing methods and sequential hypothesis testing based on statistical analysis of the properties of the signals. Incorporating contextual information is also an area that has not been researched extensively. Brock [Bro93] has addressed the inclusion of contextual information into a control framework. Most of his work focused on “high level” context such as knowledge of the environment and the relationships among objects but the approach can also be extended to encompass “low level” context, such as the awareness of one's own behavior.

To summarize, there has been a considerable work on developing tactile sensors, less work on fusing multiple tactile sensors and little done on developing sensing strategies for detecting multiple events during a task. Also, little has been done to incorporate context into sensing strategies. Our goal is to develop a general approach for event detection that incorporates multiple sensors and an awareness of the robot's actions (context).

2. Theoretical Framework

In our multi-sensor scheme, any given sensor can provide several types of information. For example, tip position, velocity and acceleration can be computed from joint sensor data and short-time energy can be obtained from skin acceleration sensor data. We refer to each of these types of information as *sensor-based* features. When contextual information about the robot's behavior is included as a feature (desired velocity for example) we refer to these features as *context-based* features. These features can be combined in a way that lets the controller determine how confident it is at any given instant that one of a list of possible events could be occurring. Each event is assigned a value between 0 and 1 that determines how confident the controller is that the event might have occurred. This is achieved with the use of *confidence distribution functions*. Another area where context is included is at the phase-level, where the robot only considers the events that are possible during a given phase and ignores the rest. This phase-level implementation of context is important because it reduces the list of possible events and minimizes the chance of detecting the wrong event. This section will introduce the feature space approach to event detection. The concept of a phase's *feature space* and its accompanying *event subspaces* will be discussed. As mentioned earlier, context is implemented at two levels in our approach: at the phase level and at the feature level.

2.1 Feature Space and Event Subspaces

For each phase in a manipulation task, if one identifies the possible events and the features required to detect them, one can construct a feature space for the phase. Let us define a feature F as a set of discrete real numbers f corresponding to all possible values for that feature and let Φ be the current phase of a manipulation task. Now let us define an n-dimensional Euclidean feature space corresponding to the cartesian product of the family of sets, or observed features, and denoted by

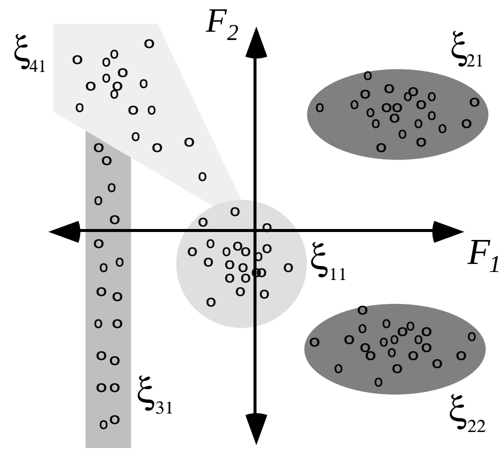
$$\boldsymbol{F}^n_\Phi = F_1 \times F_2 \times \ldots \times F_n \tag{1}$$

At any given moment during the phase, there will be an n-tuple $(f_1, f_2, ..., f_n)$ which corresponds to the current feature values and defines a position in the n-dimensional feature space.

Inside this feature space, there will be m regions ξ which correspond to the p possible events which can occur during the phase. Let

$$E^p_\Phi = \{ e^{q_1}_{1(\Phi)}, e^{q_2}_{2(\Phi)}, \ldots, e^{q_p}_{p(\Phi)} \} \tag{2}$$

be the set of q_i-dimensional event subspaces associated with the phase Φ. Note that $q_i \le n$ which means that the



$e^2_1 = \xi_{11}$ $\quad$ $e^1_3 = \xi_{31}$

$e^2_2 = \xi_{21} \cup \xi_{22}$ $\quad$ $e^2_4 = \xi_{41}$

Figure 1: Example of a 2-D feature space.

dimension of the event subspaces might be of lesser or equal dimension than the feature space. In other words, the detection of each event is not necessarily dependant on all n features. Also note that $p \le m$ because an event may occupy more than one distinct region in the feature space. Thus,

$$e^{q_i}_{i(\Phi)} = \{ \bigcup_{j=1}^{k} \xi_{ij} \mid \xi_{ij} \subseteq e^{q_i}_{i(\Phi)} \text{ for all } i \in \aleph_p \} \tag{3}$$

where k corresponds to the number of regions ξ associated with each event e. The sum of all the k_i will be equal to m. Therefore, if we define an n-tuple in the n-dimensional feature space of a given phase as

$$\boldsymbol{f}^n_\Phi = (f_1, f_2, \ldots, f_n) \tag{4}$$

It then follows that the condition

$$\boldsymbol{f}^n_\Phi \in E^p_\Phi \tag{5}$$

must be satisfied for an event to have *possibly* occurred. Note that this does not mean that an event has *definitively* occurred but rather that the n-tuple will be assigned a confidence value in the interval [0,1] based on its proximity to the subspace's centroid or some other criteria. It is then up to the controller to determine if an event has occurred or not. This will be discussed in further detail later in this section.

To illustrate this terminology, consider a simple two-dimensional feature space for a given phase, as illustrated in Figure 1. In this example, there are four possible events whose typical feature distributions are indicated by the shaded areas. At any instant during the phase, the current values of the 2 features (i.e. the 2-tuple $\boldsymbol{f}_{\Phi}^{2}$) define a position in the feature space. If this position falls within the boundaries of one of the event subspaces, it is possible that an event might have occurred and a confidence value is assigned. Note that event #3, or e_3^1, is only dependant on feature F_1 to be detected, therefore it has a one-dimensional subspace which explains the superscript of 1. Obviously, in order to eliminate ambiguity it is desirable, if possible, to define a feature space with mutually exclusive event subspaces such that

$$\bigcap_{i=1}^{p} e_i^{q_i} = \varnothing \tag{6}$$

2.2 Incorporating Contextual Information

Making use of context while performing event detection greatly reduces the chances of making erroneous decisions during a manipulation task. Generally, most attempts to include context include it as what we term "high-level context". This high-level context is mostly concerned with knowledge of the environment and how it affects the robot's behavior. In our approach, we wish to implement context on a lower level. Our definition of context is *information that makes the robot aware of its own behavior.* To achieve this, we implement context in two areas: at the *phase level* and at the *feature level*.

Phase-Level Context:

At any given instant during a task, only a subset of all the possible events for that task can reasonably occur. Therefore, our goal is to determine the sets of possible and likely events for each phase of typical manipulation tasks and thus reduce the set of events that are looked for. This use of context at the phase level reduces the set of feasible events for each phase and thereby both increases the robustness of the approach and decreases the computation time required for event detection.

Let us define the set of all t possible events during a task τ as

$$\boldsymbol{E}_{\tau}^{t} = \{e_1, e_2, \ldots, e_t\} \tag{7}$$

As stated earlier, each phase Φ will have its own subset of p possible events:

$$E_{\Phi}^{p} = \{e_1, e_2, \ldots, e_p\} \tag{8}$$

Therefore, in order to make good use of context at the phase level, it is our goal to make $p<<t$ for each phase of the task. For example, if the fingers are moving without holding an object, there is obviously no point in checking for object slippage or object contact with the environment. This simple and intuitive way of implementing contextual information is especially well suited to a phase-based strategic control law.

Feature-Level Context:

As stated earlier, features can be either sensor-based or context-based and our second implementation of context is at the feature level. When constructing the feature space for a phase, it is possible to make the robot aware of its actions by incorporating features that are not sensor-based but rather behavior-based. For example, take the case where a robot suddenly accelerates. If the controller has a simple contact detection scheme that waits for the force sensor signal to exceed a certain threshold, then it would be fooled by this sudden acceleration because the fingertip has non-negligible mass and this inertia will cause an increase in the force reading. In this case it would be possible to include desired acceleration as one of the features for that phase and thus not misinterpret the sudden acceleration as a contact.

Consider a multi-sensor robotic hand system comprising a set of tactile sensors. There are k sensor-based features that can be extracted from said sensors during a given phase:

$$\hat{\Im}_i^k = \{\hat{F}_1, \hat{F}_2, \ldots, \hat{F}_k\} \tag{9}$$

Now let us consider the behavioral information available to the robot during a given phase based on its knowledge of its own actions; for example, desired velocity, direction of travel, etc. This information can also be represented as a set of features

$$\tilde{\Im}^{\upsilon} = \{\tilde{F}_1, \tilde{F}_2, \ldots, \tilde{F}_{\upsilon}\} \tag{10}$$

where υ corresponds to the total number of context-based features used for that phase. Now if we combine the information obtained from the sensor-based features in Equation 11 with that which is obtained from the context based features in Equation 12, we obtain a more complete definition of the feature space, based on Equation 1, as follows:

$$\boldsymbol{F}_{\Phi}^{n} = \hat{F}_1 \times \hat{F}_2 \times \ldots \times \hat{F}_k \times \tilde{F}_{k+1} \times \ldots \tilde{F}_n \tag{11}$$

where there are k sensor-based features represented by $\hat{F}$ and υ (or n-k) context-based features represented by $\tilde{F}$.

2.3 Assigning Confidence

Once all the required features have been computed, the controller has to decide whether an event has occurred or not. For this purpose we desire a efficient way of assigning a *confidence* value to each event at any time during a phase. By observing these continuously varying confidences, the controller can then make decisions relating to event occurrence. We make use of *confidence*

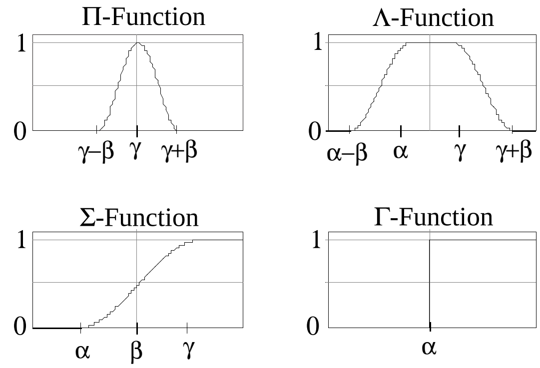


Figure 2: Four types of CDFs used.

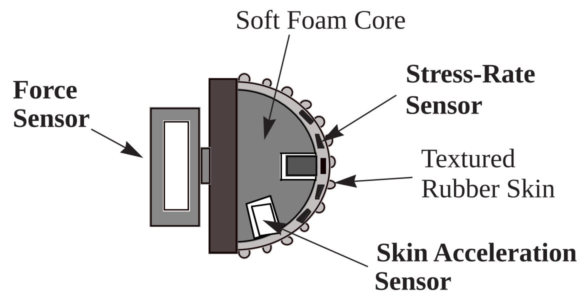


Figure 3: Fingertip

distribution functions to assign a confidence value to a given feature measurement. One of the advantages of using a distribution function to describe a feature is that it is essentially a dimensionless scalar quantity. This provides data abstraction and allows the techniques developed here to be applied to all types of features regardless of what their units or value ranges are.

Recall from section 2.1 that we define a feature F as the set of discrete real numbers f corresponding to all possible values for that feature. Now let us define a confidence distribution function $\psi(f)$ for that feature as

$$\psi(f) : F \to [0, 1] \tag{12}$$

That is, the c.d.f. is a real number

$$0 \le \psi \le 1 \tag{13}$$

where 0 means that from that feature's point of view, the event could not possibly have occurred. A value of 1 on the other hand means that from that feature's point of view, all its requirements have been met and it is satisfied that the event *could* have occurred. In other words, a value of 1 does not mean that an event has occurred, but rather that this particular requirement (perhaps one of many for the event) has been met. Any value between 0 and 1 indicates the level of confidence that the feature value has for a particular event. For our approach, we have selected four different types of confidence distribution functions which are used based on feature characteristics. These functions are similar to the membership functions used in fuzzy set theory. They include a sigmoidal function Σ, a bell-shaped function Π, a flat-bell function Λ and a binary function Γ. They are illustrated in Figure 2.

Once confidence values have been obtained for each feature, they need to be combined in order to get an overall confidence value for a specific event. Whereas the sensor-based features are typically uncertain and noisy, the context-based features represent current knowledge and serve mainly to rule out certain events. Therefore, the overall confidence for an event consists of the weighted sum of the confidences of the sensor-based features (some might have more relevance than others) multiplied by confidence values for each of the context based features. This means that a single context-based feature (firm knowledge) has the ability to nullify or significantly influence the overall confidence value while a sensor-based feature's influence is limited to the weight accorded to it. Let the overall confidence Ψ that an event ε is occurring at sampling period k be defined as

$$^{\varepsilon}\Psi_k = \left(\prod_{i=1}^{m} {}^{\varepsilon}_{i}\psi(\tilde{f}_i)\right) \times \left(\sum_{j=1}^{n} \omega_j \; {}^{\varepsilon}_{j}\psi(\hat{f}_j)\right) \tag{14}$$

where m corresponds to the number of context-based features, n corresponds to the number of sensor-based features, ω_j is the weight assigned to a sensor-based confidence values and ψ is the cdf of a given feature. Note that the same feature will have different c.d.f.'s depending on which event is being considered.

3. Experiments

The robot used for these experiments consists of a 2-DOF planar manipulator with a tactile sensor equipped fingertip mounted on the end (Figure 3). This set-up has been described in detail in [Tre93] minus the stress-rate sensor which is described in [Son94]. The fingertip is made of a soft hemicylindrical foam core which is covered by a thin layer of textured rubber skin. The system incorporates four types of sensors: position sensors, force sensors, skin acceleration sensors and stress-rate sensors.

We conducted a series of tests of the event detection scheme described in the preceding section on the first stage of a manipulation task in which the fingers approach and make contact with an object. This phase was chosen because it is present in almost all manipulation tasks and it is a phase for which it would be easy to fool a primitive event detection scheme. In this phase, we consider four types of events that cause sensor excitation and thus have the ability to trigger a reaction, desired or not, by the robot. One of these events is associated with finger

contact	hard object	high speed (8 cm/s)
		medium speed (5cm/s)
		low speed (2 cm/s)
	soft object	high speed (8 cm/s)
		medium speed (5 cm/s)
		low speed (2 cm/s)
	object moves	medium speed (5 cm/s)
link collision		high speed (8 cm/s)
		low speed (2 cm/s)
disturbance	large	medium speed (5 cm/s)
	small	medium speed (5 cm/s)
sudden acceleration		from standstill
		while moving (5 cm/s)

Table 1: Data Runs

acceleration, the remaining three involve interactions with the environment (fingertip contacts, link collisions and unknown disturbances). In this example the desired terminal event is fingertip contact with the object, which triggers the onset of the next phase. Link collisions would also trigger a new phase (typically a retreat to a safe place).

To help us assess the robustness of the approach, multiple tests were conducted over a range of finger velocities and with a variety of events as shown in Table 1.

3.1 Context-Based Features

Desired Acceleration: This feature is included to help reject any sensor excitation caused by a sudden acceleration of the manipulator. This feature has a bell c.d.f centered around zero.

Force-Velocity Dot Product: This is a useful feature for determining contact, since one of the conditions for contact is

$$\vec{F} \cdot \vec{v} < 0 \tag{15}$$

or else the increase in the force signal is probably due to a link collision or a disturbance. This feature has a binary distribution function.

Desired Tip Velocity: For contacts or collisions to occur, the finger has to be moving (assuming all objects are stationary):

$$\left|\vec{v}_d\right| \neq 0 \tag{16}$$

3.2 Sensor-Based Features

Tip Position Error: Used because it will increase for contacts and collisions but remain largely unaffected for disturbances.

Filtered Force Signal: When contact occurs, the measured force will increase gradually. For link collisions and disturbances, the fingertip will behave as a standard second order system and the force readings will oscillate about zero because the fingertip is not in contact with anything.

STE of SAS: The short-time energy of the skin acceleration sensor signal is defined as follows:

$$\zeta_n = \sum_{m=0}^{N-1} x^2(n-m) \cdot h(m) \tag{17}$$

where x is the sensor reading and h represents a sampling window (filter) of size N. This is a useful feature because the various events have very different STE ranges.

Skin Stress-Rate: The stress-rate sensor is quite insensitive to non-contact events and is therefore well suited for distinguishing between contacts and other events such as collisions and disturbances.

3.3 Results

As stated earlier, the overall confidence value for each monitored event (contact, disturbance and link collision) was computed for a variety of runs. In all of our test cases, the scheme quickly identified the correct event. Figure 4 shows the results for 4 typical runs where the overall confidence for each event is displayed as a function of time. Confidence results for contacts with hard objects at high speeds and soft objects at low speeds are displayed, as well as results for a vibration disturbance and a link collision. For all plots, the actual event occurs at t = 20ms and one can see that in every case the correct event is detected within 20ms and sometimes in less than 10ms.

4. Conclusions

Our goal when approaching this problem was to come up with a general approach to event detection in dextrous manipulation that incorporates multiple sensors and an awareness of the robot's commanded actions (a simple form of context). Our initial results indicate that these goals can be achieved. We have developed a scheme that rapidly and reliably detects typical events present in a manipulation phase. This scheme assigns a confidence value (0-1) to each possible event in a phase and the controller can use this information to determine if any of these events has occurred. To determine the confidence for each event, a feature space is constructed based on all the data features used to detect the events. Each feature is assigned a confidence distribution function. By knowing the value of each feature at any given instant, and its corresponding confidence value, it is possible to compute

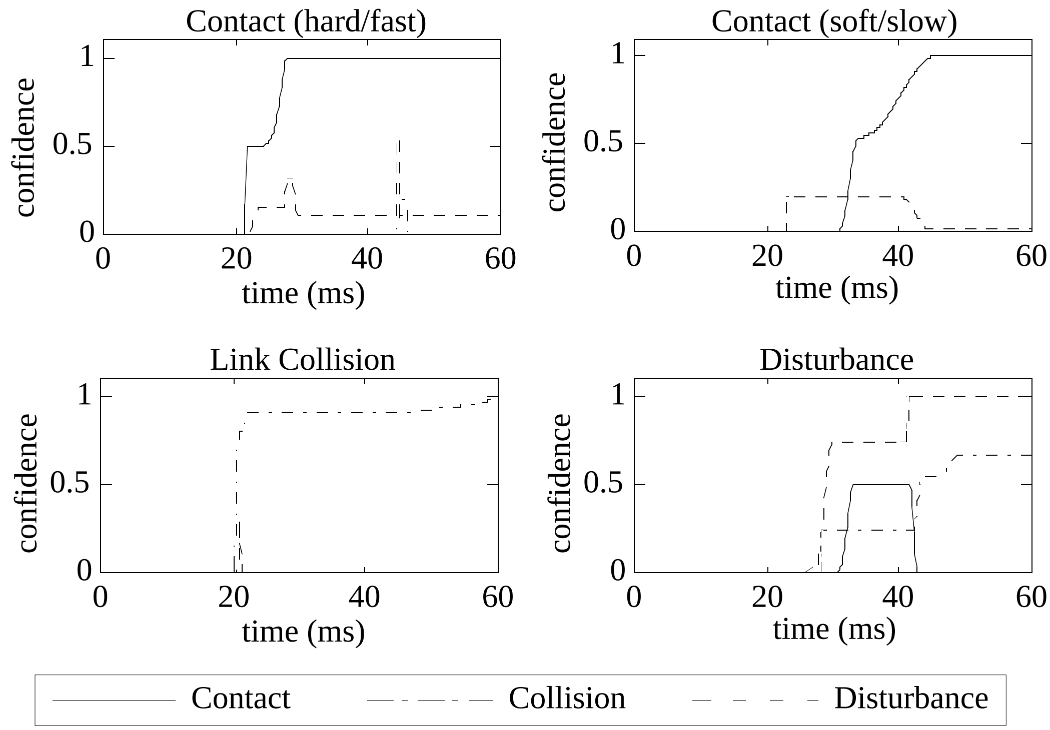


Figure 4: Sample Experimental Results

an overall confidence for the observed event. Once a robotic controller is informed of these confidence values, it can then determine when it should transition from one manipulation phase to another.

One advantage of this method is that, unlike other approaches like Bayesian methods or Neural Networks, it does not require extensive training. Instead, one is required to determine the limits for each feature for each type of event instead of training over the whole range of the feature.

A number of extensions to this approach are evident. It should be extended to other manipulation phases and we need to formalize our feature selection process. Also, event independence in the feature space needs to be studied further.

References

[Bec92] M. Beckerman, “A Bayes-maximum entropy method for multi-sensor data fusion,” 1992 IEEE Int. Conf. on Robotics and Automation, pp.1668-1674, Nice, France, May 12-14, 1992.

[Bro93] D.L. Brock, "A sensor based strategy for automatic robotic grasping", PhD Thesis, MIT, January 1993

[Cut93] M.R. Cutkosky and J.M. Hyde, "Manipulation control with dynamic tactile sensing", 6th Int. Symposium on Robotics Research, Hidden Valley, PA, Oct 2-5, 1993

[Dur88] H.F. Durrant-Whyte, “Integration, coordination and control of multi-sensor robot systems,” Kluwar Academic Publishers, Boston, 1988

[Ebe94] B. Eberman and J.K. Salisbury, "Application of change detection to dynamic contact sensing," IJRR, Vol. 13, No. 5, pp.369-394, October 1994

[Gar81] T.D. Garvey, J.D. Lowrance and M.A. Fischler, “ An inferance technique for integrating knowledge from disparate sources,” Proceedings of the 7th International Joint Conference on Artificial Intelligence, pp.319-325, Vancouver, British Columbia, 1981

[How89] R.D. Howe and M.R. Cutkosky, "Sensing skin acceleration for texture and slip perception," 1989 IEEE Int. Conf. on Robotics and Automation, pp.145-150, Scottsdale, AZ, May 14-19, 1989.

[Mar89] A.J. Maren, R.M. Pap and C.T. Harston, “A hierarchical data structure representation for fusion of multisensor information,” SPIE, Vol.1100, Sensor Fusion II, pp.162-178, Orlando, FL, 1989

[Son94] J.S. Son, E.A. Monteverde and R.D. Howe, "A tactile sensor for localizing transient events in manipulation," 1994 IEEE Int. Conf. on Robotics and Automation, pp.471-476, San Diego, CA, May 8-13, 1994.

[Tre93] M.R. Tremblay and M.R. Cutkosky, "Estimating friction using incipient slip sensing during a manipulation task," 1993 IEEE Int. Conf. on Robotics and Automation, Vol.1, pp.429-434, Atlanta, GA, May 2-6, 1993.