

Qualitative Recognition of Ongoing Human Action Sequences

Yasuo Kuniyoshi*

and

Hirochika Inoue†

*Autonomous Systems Section,
Intelligent Systems Division, Electrotechnical Laboratory,
1-1-4 Umezono, Tsukuba Science City, Ibaraki, 305 JAPAN.
E-mail:kuniyosh@etl.go.jp,

†Department of Mechano-Informatics,
Faculty of Engineering, The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113 JAPAN.
E-mail:inoue@jsk.t.u-tokyo.ac.jp,

Abstract

This paper deals with the qualitative visual recognition of continuous human action sequences. Based on an analysis of the structure of physical actions, a framework for temporal segmentation and qualitative classification of physical actions is proposed. In order to achieve correctness as well as efficiency in real time action recognition, a hierarchical spatio-temporal attention control method is developed. As the integration of these ideas, a cognitive architecture for action recognition is proposed. An experimental system is built for recognizing block assembly tasks performed by human workers. The system generated symbolic descriptions of the shown tasks in near real time.

1 Introduction

Social animals, including humans, have high competence for observing, understanding and reacting to other agents' actions effectively. Action understanding is indispensable for intelligent behavior in multi-agent environments. Also, the coincidence of higher intelligence and advanced action understanding ability suggests that action understanding may be deeply related to the fundamentals of natural intelligence.

Intensive research into this issue has been done in AI, especially concerning the plan-based understanding of natural language [Cohen and Perrault, 1979; Allen and Perrault, 1980; Litman and Allen, 1987; Wilensky, 1983]. However, not much is known about its prerequisite; the more basic form of action understanding: *qualitative recognition of physical action sequences based on visual observation*.

Early attempts at this problem dealt with extracting symbolic descriptions from motion pictures of simple two dimensional figures moving around [Tsuji *et al.*, 1977; Thibadeau, 1986]. Previous vision research in "motion understanding" has focused on extracting three dimensional motion parameters from image sequences. But this bottom-up approach has not reached the stage of providing the semantic information we need. Recently, an "assembly plan from observation" system [Ikeuchi

and Suehiro, 1992] was developed which extracted fine motion sequences from transitions among face contact states acquired by a range sensor. But this theory does not account for the continuous nature of human action sequences.

Our research goal is to provide a framework for temporal segmentation and qualitative classification of continuous human action streams carried out in a three dimensional world. It is intended to be suitable for real time processing. We also propose this as a cognitive architecture for such tasks.

The power of active and purposive strategies in vision has been widely recognized [Bajcsy, 1988; Aloimonos and Weiss, 1988; Rimey and Brown, 1992]. They are especially important when the observer is involved in action [Ballard, 1989; Whitehead and Ballard, 1990]. Although we do not go into the details of vision processing in this paper, we show that active selection of input information is necessary in the temporal domain as well as in the spatial domain. Also, using attention control for both visual processing and internal representations, we can establish symbol grounding in a dynamic environment.

We start the body of this paper by deriving fundamental methods of action segmentation and action classification from an analysis of the notion of action in section 2. Based on these considerations, we propose an overall algorithm for action recognition in section 3. We also present an implementation of an action recognizer for block assembly tasks in section 4. Then we demonstrate the validity of our methods by experimental results of near real time recognition of several simple human tasks in section 5.

1.1 The Example Task

We choose the following example task both for explaining the concepts described in this paper and for implementing a working system:

Input: Observation, by stereo vision, of a human worker constructing various structures with blocks.

Output: Symbolic assembly plans. Each plan is a sequence of assembly operators with intermediate state descriptions. These are to be used by a robot for reproducing the performance of the task [Kuniyoshi *et al.*, 1989].

Assumptions: Explicit information on temporal segmentations and a priori knowledge about a specific task are not given. Classification of possible assembly operations is given. The worker uses only one hand and grasps only one block at a time. One structure is assembled in one task. Each task is carried out only once from start to finish without a pause. Spatial segmentation is tractable. Complete occlusion of blocks does not occur. There are no "messed up" situations, such as collapses. Also, blocks are brighter than the background.

2 The Fundamentals of Qualitative Action Recognition

2.1 Extracting the Structure of Physical Actions

One important concept is that of *qualitative classes* of actions. We share some basic perspectives on movements and actions with Israel et al. [1991]. First, we want to generate a qualitative description of human action sequences that can be used for reproducing the observed task in a different environment. In Israel's words, we are interested in *repeatable* action classes, not in accurate measurement of bodily movements. Let us consider the example of picking up a block from a table. In every instance of such an action, the hand goes roughly down and up, but it never takes exactly the same trajectory twice. It may even take a completely different path when there are obstacles in the way. Second, there are nested action classes which form layered contexts. For example, we are usually focusing on a part of the worker's body while ignoring the movements of the rest, which serve as the background condition (Israel's notion of *movement context*). In a block assembly task, when fingers operate on a block in order to align it with another block (Figure 1), an observer is likely to focus on the fingers and the orientation of the manipulated block, provided that the hand is staying roughly in the same position. When the hand suddenly starts off to fetch another block, such fine movements no longer are significant and so the observer must quickly switch its attention and follow the hand.

Another important concept is that of actions as *causal processes*. As Dretske [1988] emphasized, the *behavior* of a system (performer) is not merely a sequence of bodily movements nor physical events, but a *causal process*, a structure wherein internal causes (conditions or events) produce certain external movements or changes. The process begins with the causes and ends with the external events, and comprises time and space sufficiently large to encompass its constituents. Dretske did not make a clear distinction between a performer's bodily *movement* and its *effect* on the environment. But we need more detailed analysis of this, since the only observable portion of an action is the external part of the process; the *movement* causing some *effect*.

Clearly, for action classification, an observer needs information of not only the *movement* but also of the *effect*, which must be causally linked. For example, picking up a block (operation: PICK), putting it down (opera-

tion: PLACE), and more tricky cases (operation: NO-OP) (where an empty hand (or a hand holding a block) approaches the block (or the table) but leaves without grasping (releasing) the block), all involve qualitatively very similar hand movements (approach and then depart from the table), and they cannot be classified using only movement measurements¹.

Based on the above considerations, we propose the following principles of action observation and recognition:

1. Observing the performer's bodily movement is necessary but not sufficient. A recognizer must also look for causally linked effects in the performer's surroundings and relate movement and effects.
2. It is very important to accurately determine the spatio-temporal extent of an action being observed. Otherwise actions will be mis-recognized, because different selections of spatio-temporal extents affect which physical events are selected to be related with the performer's movements.
3. Representations of qualitative movements should be organized so as to reflect the structure of movement contexts. One solution is a hierarchy defined in terms of an inclusion relation between spatio-temporal extents of the movements².

Let us define three concepts for describing actions from an observer's point of view: *movement*, *motion*, and *operation*. A *movement* takes two arguments *base* (the moving/changing thing) and *t* (a time point), and qualitatively describes how *base* is moving (position, orientation, and their time derivatives) at *t*.

We distinguish between *movements* and *motions*. A *motion* of an object (*base*) is not just moving in some way at one time instant, but moving *toward*, or *away from*, or somehow *relative* to, some other object(s) (*targets*) within a given temporal interval *T*. So, a *motion* takes three arguments, *base*, *targets*, *T* and qualitatively describes how *base* is moving with regard to *targets* within *T*.

An *operation*, a synonym for *physical action*, takes three arguments *base*, *targets*, and *T*, and is defined as a causal process which begins with a *motion* and ends with an *effect*. An *effect* describes a qualitative change involving *base* and *targets* within *T*.

2.2 Temporal Segmentation and Action Classification

Newton et al. [1977] carried out a series of psychological experiments which provided a strong evidence that humans perceive an action stream as a sequence of clearly segmented "action units", and that the most of the useful information for action recognition is extracted at the temporal segmentation points. Further, he suggested that the key information useful for action identification

¹Finger movements in grasping and releasing the block are typically very small and therefore almost impossible to detect reliably using vision.

²For physical movements, the inclusion relation between temporal extents coincides with the inclusion relation between spatial volumes swept by the moving object.

is not the individual states themselves but the transitions from one state to another. Unfortunately, he did not explain clearly how the segmentation is done, nor did he suggest any candidate feature types which classify actions.

Based on a logico-linguistic analysis of actions, Thibadeau [1986] argued that segmentation points should be defined in terms of "second order changes" of *interpreted* images. He also claimed that people do action classification by checking whether a specific condition holds at the end point of the action. His overall theory is based on the assumption that complete symbolic interpretation of every image is provided for the whole temporal sequence, and the first and the second order changes between the descriptions are constantly computed. This assumption is infeasible for recognizing human actions carried out in the three dimensional world, especially when using vision modality. We need *direct* methods to detect segmentation points and to classify actions. The only *direct* feature for action segmentation he mentioned is acceleration of a moving object. Unfortunately, we have found that using only acceleration is not a very effective method for segmentation of real actions.

Our approach is to derive a concrete definition of both action segmentation and action classification simultaneously based on the definitions of physical actions and motions.

Let us assume that an observer sees a physically closed world W containing a fixed set of objects O . We define a state $S(t)$ of W as a set of n -ary measurements (for all possible n) of O at time t , for example, position/orientation parameters, distances or contact states between pairs of objects. Let P stand for the set of all possible temporal evolutions of $S(t)$. Then, qualitative recognition of an action requires cutting a portion from P and identifying it as one of the known action concepts.

Following the formulation of the notion of concepts by Hobbs [1985], we define a set of conceptual action classes C as a quotient set of P by a certain equivalence relation R_c : $C = P/R_c$. A set of observable qualitative features F is also defined as a quotient set of P by an equivalence relation R_o : $F = P/R_o$.

We state that F has *indexicality* to C if and only if the observation process, R_o , articulates P in the very same way as the concept formation process R_c does.

An action is a process in which one object affects the state of others, as stated in section 2.1. So, conceptually, an action can be represented using a tuple (o, T, SI, SC) , where SI is a subset of the temporally invariant portion of P , SC is a subset of the temporally changing portion of P , o is a subset of O which support SI and SC , and T is a temporal interval. Then we can divide R_c into three subrelations and relate to R_o , as described below.

Attention: Focused spatial extents o which specify objects, geometrical features or locations, and the temporal extent T . The observer must keep track of o within T . The spatial entities, as they generally participate in motion, will be called: **base**, **targets**, and **special**.

Context: Invariant temporal property of motion, SI :

Qualitative movement and relations to other focused entities (that is, the position, contact state, etc. of the **base** with regard to the **targets**).

Change: Qualitative change of relations (held, affixed) or relative position/orientation (on, coplanar) among focused entities, SC . This introduces an equivalence relation among possible temporal changes in W . We hypothesize that, provided that the vision process correctly captures the spatio-temporal extent which corresponds to the conceptual attention and context, there exists an "indexical feature set" F which can be detected by a very simple visual process. (Examples will be given in section 3.2.) In other words, we hypothesize that action recognition is easy if attention is properly controlled and context is properly maintained.

There is an inherent difficulty in the observation process: The attention, o and T , are variables which must be filled in by visual observation. In general, the only constraint on their values are imposed by SI and SC . However, SI and SC cannot be observed until o is determined because they are defined over o . In order to break this circular problem, we take a following method: Assume that at least one object in o is known, call this subset **base** (it is maintained by tracking). Then choose a portion of SI which is defined using **base**. Using visual search (section 3.2), find other objects in O relative to **base** in terms of SI . Call them **targets** and include them in o . The **base** is reinitialized when the level of hierarchical attention (section 3.2) is changed.

The temporal extent T is defined when the o and SI are determined. In other words, temporal segmentation is done on detection of changes in which object is moving, whether it is moving toward or away from an object, and which such object, how it is moving, whether it is touching another object, etc. We will describe an effective method for segmentation in section 3.

2.3 Looking for Relevant Effects

An action recognizer must find a qualitative effect (change or invariance) which is causally related with the performer's motion. Selecting the relevant effect is of vital importance to correct recognition of an action. In an everyday world, many things change continuously due to natural phenomena or because multiple agents do different things simultaneously. Relating a change to the wrong agent's motion results in a recognition failure. There are also maintenance actions, actions trying not to change the current state of certain things. If a perfect recognizer were to try to handle both ordinary and maintenance actions, it might first generate a complete description of the world and then extract all causal relations between motions and changes or invariances. In principle this would work, but this would require describing everything in the world at every instant of time and examining every possible relations among them — which would clearly overload any system.

This is the frame problem [Pylyshyn, 1987] and its general solution is yet unknown. But humans (and many other animals) seem to have little difficulty in recognizing other agents' physical actions in real time.

We propose a straightforward but incomplete solution consisting of two methods:

1. Employ heuristically driven attention control both in the temporal and spatial dimensions to actively find and collect only relevant information.
2. Use predictions based on the routineness of ordinary action sequences.

Our heuristics for guiding the observer's attention are based on the nature of causality. Although causality in general has a very complex structure, we restrict our discussion to a very limited case, the block assembly task. Let us define simple causality in block assembly operations: Let M be a motion of a **base** concerning **targets** over a temporal interval $T^M = [t_1^M, t_2^M]$, and let E denote a change or an invariant in the physical state of **targets** over $T^E = [t_1^E, t_2^E]$. Then a causality $M(\text{base}, \text{targets}, T^M) \leadsto E(\text{targets}, T^E)$ holds iff $T^E \subset T^M$ and $\text{Touch}(\text{base}, \text{targets}, T^T), T^T \subset T^E$, where $T^T = [t_1^T, t_2^T]$. The condition part is based on the fact that all the objects in the environment (except for the hand) are passive and no mess (dropping or crashing) is allowed, which means that effects relevant to the current **base** (hand) movement occur only while the **base** (hand) touches the **targets**.

It follows from the above definition that if the **targets** are defined and if the temporal segmentation is correctly done, then E is obtained by differentiating two snapshots of **targets** at the segmentation points t_1^M and t_2^M . This means that temporal segmentations must be done correctly in real time. Moreover, an observer must predict the location of the **targets** before the **base** actually touches it, otherwise it cannot direct attention to the relevant region. The prediction is realized by visual search (section 3.2), which directly searches through the 3D space by guiding visual attention.

In addition, the actual location to take snapshots depends on the context of the action sequence. For example, when a hand approaches a block from above, if the observer knows that the performer did a PICK operation before (and so is holding a block), the observer should anticipate a PLACE operation and watch the location just *above* the target block.

3 The Algorithm for Action Recognition

Based on the analysis given so far, we propose an algorithm for action recognition. First, we describe the main process and then explain briefly about the attention control.

3.1 The Main Process

The algorithm of the main process consists of nested loops executed in parallel. A single loop incorporates the following steps:

1. **Do Pre-segmentation:** Continuously check the qualitative properties of the **movement** of the **base** until one of the anticipated types of changes is detected.

2. **Invoke Attention Control:** Select a spatial attention control mode (pre-attentive movement detection, hand/object tracking, visual search, etc.) which is appropriate for the action context and invoke it.

3. **Analyze the Event:** Wait until one of the anticipated types of events is detected. On receiving a segmentation event, execute the steps below, otherwise, ignore it and go back to the pre-segmentation step. For example, when visual search succeeds, the goal location is saved and a segmentation event is raised.

- (a) **Detect a Change:** Take a subimage of the region including the **target** and store it in the environment model. Compare it with the subimage taken at the previous segmentation point of this recognition level.
- (b) **Classify an Action:** Select one operation type for the current motion based on the qualitative change between the subimage snapshots. Store it into the action history memory (part of the action model).
- (c) **Calculate/Store States:** Based on the recognized operation type, calculate the change in geometrical relations among the objects under attention. Store the results in the environment model and save them in the action history.
- (d) **Do Segmentation:** Mark the end of the current motion. If finer levels of action recognition loops are currently active, force them to do a segmentation and then terminate. Choose the next motion class, as specified by the event type, and instantiate it. Start processing on it.
- (e) **Anticipate Next Actions:** Based on typical action sequence patterns, predict potential next motions, including finer level ones. Update the stack (and thereby initiate checking for pre-segmentation events for those motions).

3.2 Attention Control

In our system, spatial attention control consists of three processes: (1) pre-attentive, (2) fixating or tracking, (3) visual search. Information from the pre-attentive and the tracking modes facilitate pre-segmentation. Regions of interest for change detection are maintained by the fixation mode.

1. Pre-attentive mode is used for continuously monitoring the whole workspace very coarsely. It detects a moving thing very quickly and reports a rough estimation of its 3D position.
2. Tracking mode is used to fixate a moving object. It continuously reports the current 3D position, velocity, and acceleration of the **target**. Fixation watches a specified stationary 3D region. It is used to take a visual snapshot of the area.
3. Visual search mode is invoked when necessary (at pre-segmentation). It places a 3D virtual region on the current spatial attention point (**base**), moves the region gradually in a specified direction and continuously checks for a specified visual feature inside

it, until something is found or the travel distance exceeds the preset limit. It succeeds if anything is found, and reports its position, otherwise it fails. The search direction and the visual feature to be checked is computed based on the movement of the base, and expectations about the next action.

The system also uses temporal attention control. This triggers and stops various modes of spatial attention control based on action (pre-)segmentation. To be specific, tracking is invoked when a moving object is found, subimage snapshots are triggered at segmentation points, and visual search is invoked at pre-segmentation points.

In addition to spatial attention control and temporal attention control, we use hierarchical attention control. Our recognizer maintains superimposed multiple coarse-to-fine spatial attention regions. The levels correspond to the motion hierarchy which defines dependencies among motion classes (movement contexts). The overlapping attentions are processed in parallel.

Let us see two examples of indexical features defined in section 2.2.

First, classification of PICK, PLACE, and NO-OP is done by subtracting silhouette sub-images around the location of the targets (the block to be picked up, or the location in which a block is to be put) taken at segmentation points. Assuming that the manipulated block has a brighter color than its background, the three qualitative difference values, decreased, increased, and no-change, directly indicate the actions PICK, PLACE, NO-OP, respectively.

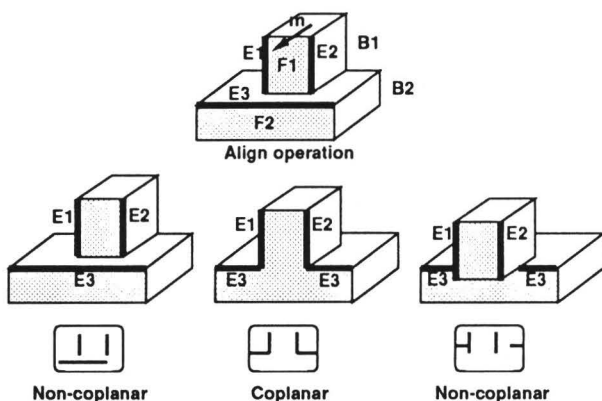


Figure 1: ALIGN action (top) and indexical features (in round boxes) of coplanar/non-coplanar relations.

Second example is ALIGN action shown in Figure 1, where block B1 is manipulated so that the two faces F1 of B1 and F2 of B2 coincide in a common plane. In the context in which the bottom face of B1 touches the top face of B2, and B1 is moving in direction $\vec{m}$ so that F1 approaches the common plane, we can detect the coplanar state by merely checking for specific types of specific edge junctions, as shown in Figure 1.

4 Implementation of the Action Recognizer

Here we briefly describe an implementation of the action recognizer. Details are presented elsewhere [Kuniyoshi *et al.*, 1992]. It is written in an object-oriented Lisp [Matsui and Inaba, 1990], and works by message passing. The overall structure of the recognizer is given in Figure 2, illustrating the state of the system while recognizing a PLACE and its sub-action, APPROACH, expecting their end points. The recognizer consists of visual feature detectors, an action model, an environment model, and an attention stack. The knowledge comprising two models are implemented as collections of nodes. Only the part of the action model and the environment model which is pointed to by the attention stack is activated at any one time.

4.1 Visual Feature Detectors

The visual routines used in spatial attention control are implemented as separate classes called visual feature detectors. These are: movement-detector, hand-tracker, object-finder, change-detector (silhouette differentiation), coplanar-detector (edge junction classifier). Each detector is connected to an appropriate environment model node, which has an associated method to control the detector to extract a specific visual feature. Each detector has a specific vision algorithm and maintains a stereo pair of visual windows. The visual feature detectors are executed as parallel background processes using multi-processor vision hardware [Inoue and Mizoguchi, 1985].

4.2 The Action Model

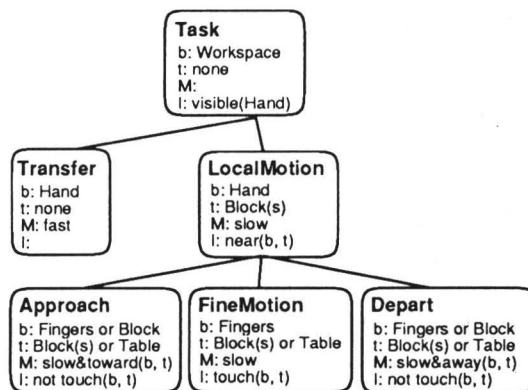


Figure 3: Class hierarchy of assembly motions. Items "b:" and "t:" specify the environment model classes permitted to fill the slots "base" and "targets". "M:" specifies the qualitative movement. "l:" denote invariances throughout the temporal interval of the motion.

The motion hierarchy for the assembly task example is defined as illustrated in Figure 3. Class "Task" covers the whole spatio-temporal extent of an action-sequence. There are two types of basic assembly motions: (1) "Transfer", a large movement of the hand in free space, and (2) "LocalMotion", a hand movement

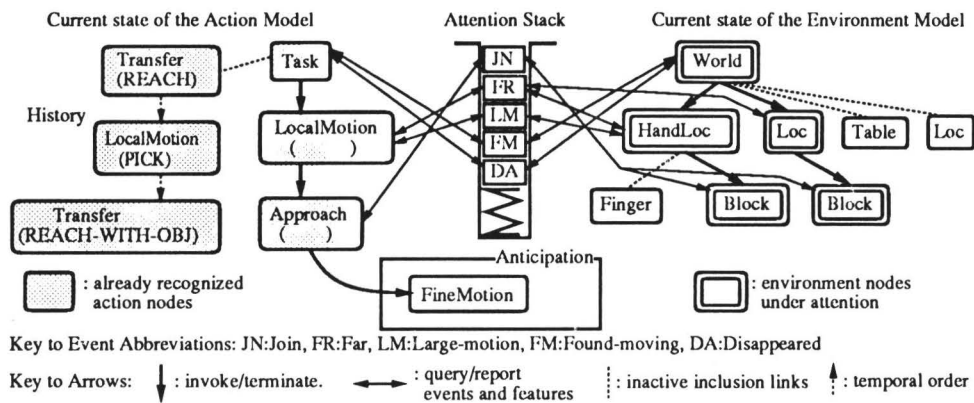


Figure 2: Structure of the action recognizer.

near a **target** object. "LocalMotion" is further subdivided into three types of assembly motions: (1) "Approach", in which the fingers (and the held object, if any) move toward the **target** objects, (2) "Depart", similarly, and (3) "FineMotion", in which the held object moves in contact with the **target** objects.

Each motion class is implemented as a separate node. Each class node specifies segmentation events, and action classification mappings, and has empty slots to be filled during recognition. The slots named "base", "targets", and "special" are for pointers to appropriate environment model nodes. For example in a PLACE operation, "base" points to a node representing the hand, "special" points to a node for the block held, and the blocks on which that block is to be put are represented by nodes pointed from the "targets". A motion node has various other slots. The "initial-state" and "final-state" are for snapshot descriptions of relations among base, targets, and special. The "operation" is to be filled with a result of action classification. The potential operation types for each motion class are listed in Table 1; this also shows which indexical feature values corresponds to which operations. The mappings are explicitly represented in each motion class node; this information is used by the main recognition process. An instantiated motion class is tagged with an operation type when action classification succeeds. An example of an instantiated node is shown in Figure 6.

4.3 The Environment Model

An environment model is a collection of nodes representing the observer's view of the current task state. It represents information about focused objects; this is used for the spatial attention control. The relative positions and dimensions of the **base** and the **target** blocks are used to determine the corresponding parameters of attention regions. Whether or not the hand is holding a block is used to help set the snapshot regions appropriately.

The environment model consists of distributed nodes which belong to one of following classes organized in a coarse to fine hierarchy:

World: A node containing a very coarse global map of

the whole environment. The entries in the map are the nodes at one lower level: HandLoc, Locs, and Table. A movement-detector is associated with this class.

HandLoc: A special node that represents the convex closure including the hand and the held block, if any. It describes the position and the area of the closure. It is identified by the silhouette shape and gross motion. Its 3D position is continuously updated by a hand-tracker. The state of the hand, eg holding a block or not, is also described. This information is not acquired by vision but remembered as a result of recognizing a previous action in the current sequence. Contains a visual search procedure.

Loc: A node that represents a convex closure including one block or an aggregation of blocks. It describes position and area. The primal visual feature representing this level is "silhouette". A change-detector is associated with this class.

Table: A special node representing the top of the work-table.

Block: A precise geometrical model (B-rep) of a block. Describes shape, position, and orientation. Instantiated by matching a model to the line image. An object-finder and a coplanar-detector are associated with this class.

Each class node has slots for current position, a geometric model of the described region, and links to other nodes for other objects to which it bears relations such as supporting and holding. It also specifies associated visual feature detectors and their control procedures, and methods for interpreting visual features to generate qualitative events (that is, to recognize event types).

4.4 Hierarchical Parallel Automata for Motion Segmentation

Motion segmentation corresponds to transitions in hierarchical parallel automata, as illustrated in Figure 4. As stated in the discussion of the main recognition process, the three levels of automata work in parallel, but when a segmentation occurs at one level, the lower level

assembly motion class	indexical feature(s)	feature value	assembly operation
Task	none		BUILD
Transfer	(SegEvent, Hold)	(:Near,:False)	REACH (Move an empty hand to a block.)
		(:Near,:True)	REACH-WITH-OBJ (Move a grasped block elsewhere.)
		other	WITHDRAW (Retract the hand after an operation.)
LocalMotion	SilDiff	:decreased	PICK (Pick up a block from the table.)
		:increased	PLACE-ON-TABLE (Put a block down on the table.) PLACE-ON-BLOCK (Put a block on (an)other block(s).)
		:no-change	NO-OP (Did neither PICK nor PLACE.)
Approach	Hold	False	APPROACH
		True	APPROACH-WITH-OBJ
Depart	Hold	False	DEPART
		True	DEPART-WITH-OBJ
FineMotion	CoplDet	:Positive	ALIGN (Move a block in contact with another to align their side faces.)
		other	NO-OP

Table 1: Mapping between assembly motion classes and assembly operations. Each motion is classified by the indexical feature values to one of the corresponding operations. Hold = True iff the Hand is holding a block. SilDiff = temporal difference of silhouette based subimage snapshots. CoplDet = temporal difference of truth values (True = 1, False = 0) meaning whether coplanar indexing edge junctions were found. In the action model used for the experiments, "Task" is always classified as a BUILD operation.

Event Type	Meaning	Detector	Env. model node responsible
:Loc-exist	identify and locate objects	object-finder	World
:Found-moving	a moving target (Hand) is found inside base (World)	movement-detector	World
:Disappeared	target (Hand) went outside of base (World)	hand-tracker	World
:Large-motion	base (Hand) is moving fast	hand-tracker	HandLoc
:Small-motion	base (Hand) is moving slowly	hand-tracker	HandLoc
:Near	base (Hand) is near some targets (Blocks)	visual-search	HandLoc, Loc
:Far	base (Hand) is far from targets (specified blocks)	hand-tracker	HandLoc, Loc
:Join	base touches targets (specified blocks)	hand/object-tracker	HandLoc, Loc
:Split	base separates from targets (specified blocks)	hand/object-tracker	HandLoc, Loc

Table 2: Qualitative visual events. Each event is detected by a corresponding environment model node using the specified detector. :Large-motion and :Small-motion are pre-segmentation events. Others are segmentation events.

automata are forced to terminate. This is done to maintain correct motion contexts. The transition events are described in Table 2. An event is detected by a corresponding environment model node. It invokes an appropriate visual feature detector, controls the attention region and interprets the result based on the current state of the node.

4.5 The Attention "Stack"

The "attention stack" is our mechanism for implementing the hierarchical parallel automata. It relates the action model and the environment model dynamically in the context of action recognition. It is by manipulating the attention stack that the main recognition process controls the whole recognizer. The stack elements correspond to the hierarchy of assembly motion types.

Each element of the stack can hold several "attention nodes". These nodes are generated dynamically to represent currently expected events, links to corresponding action/environment model nodes, and to temporarily store detected events and feature values. The attention

stack has two methods: (1) "Detect" queries the associated environment model node for the expected event and records the result, (2) "Analyze" sends the recorded event type and the feature value to the associated action model node, which provides the information needed to specify attention control and the anticipated segmentation events.

5 Experimental Results

The experimental system succeeded in generating symbolic descriptions of human tasks in near real-time.

Figure 5 shows a monitor display during the recognition of an table construction task. The worker's hand is tracked by multiple visual windows and an indexical feature is extracted from the **target** region. The result of recognition is displayed at the bottom of each image. The elapsed time for the whole task was 3 min. 50 sec., excluding the computation time for initial state recognition. The worker moved his hand at low speed (typically, 10 cm/s) so that the system could follow. The system

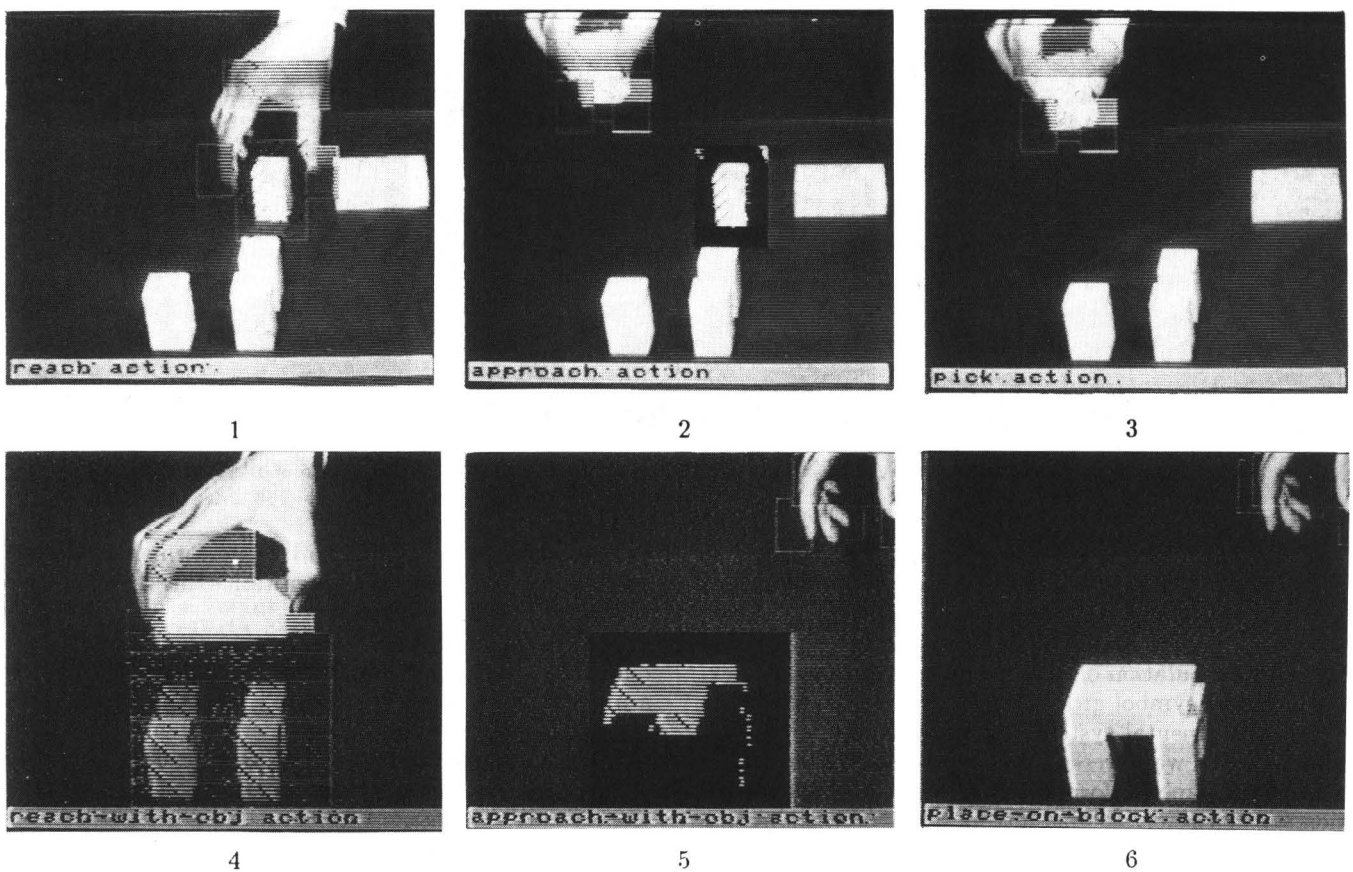


Figure 5: Two action recognitions involved in recognizing the task "Build Table": (1)(4) Target region defined. (2)(5) Differentiation. (3) Classified "PICK". (6) Classified "place-on-block".

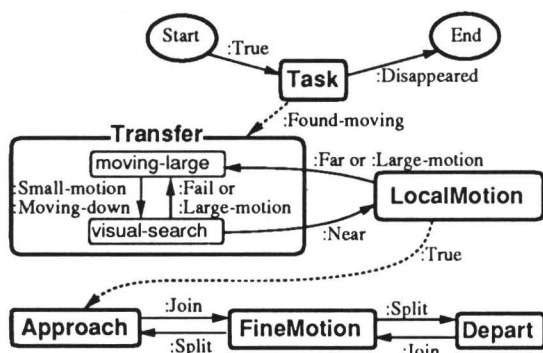


Figure 4: A hierarchical parallel automata model for action segmentation. Round boxes denote states and arrows show transitions. Dotted arrows show triggering of lower level transitions.

did all the processing for action segmentation and action classification *on-line*, and reported the recognized action type in time, before the worker completes the next action. Twenty-one basic level operations, were correctly segmented and classified. Contents of the primary slots of the recognized final PLACE operation is shown in Figure 6. A summary of the whole recognized sequence is

given in Figure 7.

Figure 8 shows the coplanar detector at work. In this experiment (run separately using this detector in isolation), the detector was run continuously to repeatedly check for coplanar relations. The held block was moved continuously and the detector reported coplanar/non-coplanar discrimination at a speed of 1 Hz.

Other assembly tasks recognized successfully by the system include; (1) pick and place with an embedded ALIGN operation (1min. 30sec.), (2) tower building (1min. 30sec., 13 operations), (3) arch construction (2min., 16ops.) (4) building an inverted arch balanced on a center pillar (2min. 40sec., 22ops). The last task contains a maintenance action, carrying a block with other blocks on it.

6 Conclusions

We have demonstrated the qualitative recognition of ongoing human action sequences. Our framework includes temporal segmentation, performed by attention switching; this is initially triggered by qualitative changes in the movement of the tracked object and is confirmed by directed visual search; and action classification, efficiently done on the basis of context dependent indexical features which directly point to qualitative action concepts.

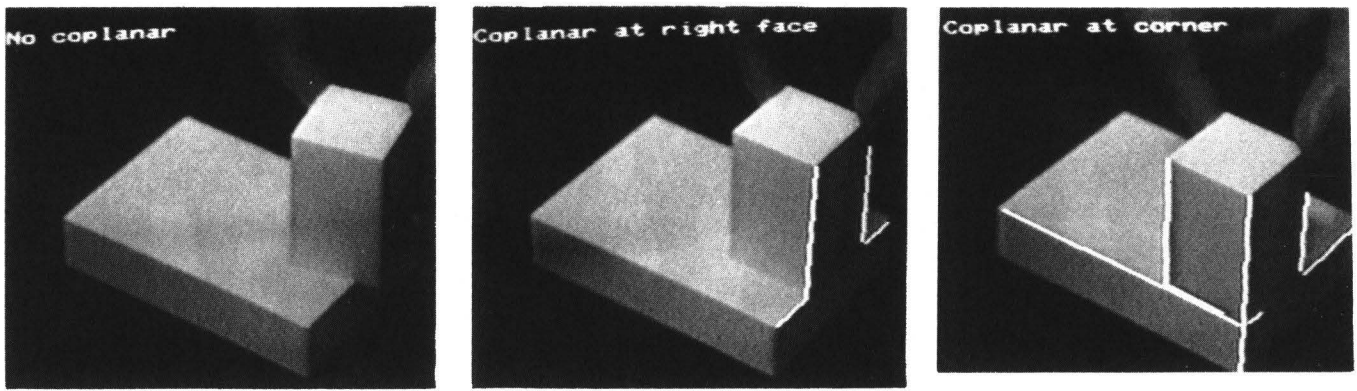


Figure 8: Recognizing an ALIGN action. Indexical features are displayed as white lines.

The relation between action recognition and the frame problem was shown. A hierarchical spatio-temporal attention control was proposed as an incomplete but practical solution. It works both for vision processing and for the manipulation of internal representations, achieving dynamic symbol grounding to the real world.

An experimental system which recognizes simple assembly tasks was presented; the system demonstrated a near real time performance. Without any cue for action segmentation such as hitting a key or withdrawing a hand from the workspace at the end of each action, the system correctly recognized each action before the worker completed the next action.

However, our implementation relies on the simplicity of the example task and substantial improvements will have to be made for extending the system for realistic domains. First, classes of recognizable actions needs to be extended. Second, the vision modules should be improved by adopting, for example, optical flow patterns for movement detection and scene segmentation, powerful tracking methods for 3D object movement, multiple features for object detection and identification, active stereo vision with a zero disparity filter for object segmentation, etc.

A straightforward application of the presented framework will be programming robots by showing them examples of the performance of tasks, which we have demonstrated elsewhere [Kuniyoshi *et al.*, 1992]. Another promising direction is intelligent behavior in multi-agent environments, where an agent recognizes other agents' actions, makes intelligent decisions and reacts to those actions, all in real time.

We expect an important modification of the presented framework: Unifying action recognition and action performance. Although the presented framework assumes that the observer uses the result of action recognition for reproducing the observed task performance, the two processes (recognize and perform) are separated, using a symbolic description of action sequences as the sole interface. In the modified framework, an agent observes its own actions as well as other agents' actions, *understanding* actions in terms of what it can do (not limited to mimicking), and sharing skills between action recognition and action performance. Such agent will be able

to learn how to recognize and to perform new sorts of actions by itself, improve recognition skills through practicing actions and vice versa.

Many other problems are open to the researchers of qualitative action understanding, including (1) building a more general and precise theory of action recognition, (2) reasoning about the performer's intentions, (3) exploiting background knowledge such as functional descriptions of physical objects.

Acknowledgements

Masayuki Inaba provided numerous important suggestions and comments, especially on the implementation of the presented theory and its robotic application. Yuji Fujita, Kazuro Hamada, Makoto Fujii, Taketoshi Mori, Satoshi Iwashita, and Tomohiro Shibata contributed to the experimental system. Nigel Ward provided many helpful comments on the draft of this paper.

References

- [Allen and Perrault, 1980] J. F. Allen and C. R. Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15:143-178, 1980.
- [Aloimonos and Weiss, 1988] J. Aloimonos and I. Weiss. Active vision. *Int. J. of Computer Vision*, pages 333-356, 1988.
- [Bajcsy, 1988] R. Bajcsy. Active perception. *IEEE Proceedings*, 76(8):996-1006, August 1988.
- [Ballard, 1989] D. H. Ballard. Reference frames for animate vision. In *Proc. IJCAI 11th*, pages 1635-1641, 1989.
- [Cohen and Perrault, 1979] P. R. Cohen and C. R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3:177-212, 1979.
- [Dretske, 1988] F. Dretske. *Explaining Behavior*. MIT Press, 1988.
- [Hobbs, 1985] J. R. Hobbs. Granularity. In *Proc. IJCAI 9th*, pages 432-435, 1985.
- [Ikeuchi and Suehiro, 1992] K. Ikeuchi and T. Suehiro. Towards an assembly plan from observation. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2171-2177, 1992.
- [Inoue and Mizoguchi, 1985] H. Inoue and H. Mizoguchi. A flexible multi window vision system for robots. In *Proc. Second Int. Symp. on Robotics Research*, pages 95-102, 1985.
- [Israel *et al.*, 1991] D. Israel, J. Perry, and S. Tutiya. Actions and movements. In *Proc. IJCAI 12th*, pages 1060-1065, 1991.

```

motion: localmotion305 operation: (place-on-block)
|base: hand183
|targets: (brick190 brick186 brick187 brick188)
|special: plate189
|initial-event: :near |final-event: :found-moving
|initial-state:
hand183 pos: #f(-41.9 129.9 137.4) with relation:
(holding plate189)
plate189 pos: #f(0.0 0.0 0.0) with relation:
(held-by hand183)
brick190 pos: #f(-24.9 178.9 25.0) with relation:
(on table182 :at #f(-24.9 178.9 25.0) :wrt table182)
brick186 pos: #f(12.5 126.9 25.0) with relation:
(on table182 :at #f(12.5 126.9 25.0) :wrt table182)
brick187 pos: #f(-73.1 137.3 25.0) with relation:
(on table182 :at #f(-73.1 137.3 25.0) :wrt table182)
brick188 pos: #f(-42.9 85.7 25.0) with relation:
(on table182 :at #f(-42.9 85.7 25.0) :wrt table182)
|final-state:
hand183 pos: #f(167.5 153.2 125.2)
plate189 pos: #f(-25.1 151.0 59.8) with relation:
(on-multiple (brick190 brick186 brick187 brick188)
:at #f(-25.1 151.0 59.8))
brick190 pos: #f(-24.9 178.9 25.0) with relation:
(co-support plate189 :at #f(1.1 27.9 -34.8)
:wrt plate189)
(on table182 :at #f(-24.9 178.9 25.0) :wrt table182)
brick186 pos: #f(12.5 126.9 25.0) with relation:
(co-support plate189 :at #f(37.6 -24.1 -34.8)
:wrt plate189)
(on table182 :at #f(12.5 126.9 25.0) :wrt table182)
brick187 pos: #f(-73.1 137.3 25.0) with relation:
(co-support plate189 :at #f(-48.0 -13.7 -34.8)
:wrt plate189)
(on table182 :at #f(-73.1 137.3 25.0) :wrt table182)
brick188 pos: #f(-42.9 85.7 25.0) with relation:
(co-support plate189 :at #f(-17.9 -65.3 -34.8)
:wrt plate189)
(on table182 :at #f(-42.9 85.7 25.0) :wrt table182)

```

Figure 6: Recognized final "PLACE" operation in task "Build Table". Placing the top plate. Shape names are abbreviated as; "brick" for "short-brick-standing" and "plate" for "rectangular-plate".

- [Kuniyoshi *et al.*, 1989] Y. Kuniyoshi, M. Inaba, and H. Inoue. Teaching by showing: Generating robot programs by visual observation of human performance. In *Proc. 20th Int. Symp. on Industrial Robots*, pages 119-126, 1989.
- [Kuniyoshi *et al.*, 1992] Y. Kuniyoshi, M. Inaba, and H. Inoue. Seeing, understanding and doing human task. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2-9, 1992.
- [Litman and Allen, 1987] D. J. Litman and J. F. Allen. A plan recognition model for subdialogues in conversations. *Cognitive Science*, 11:163-200, 1987.
- [Matsui and Inaba, 1990] T. Matsui and M. Inaba. Euslisp: An object-based implementation of Lisp. *J. of Information Processing*, 13(3), 1990.
- [Newton and others, 1977] D. Newton *et al.* The objective basis of behavior units. *J. of Personality and Social Psychology*, 35(12):847-862, 1977.
- [Pylyshyn, 1987] Z. Pylyshyn, editor. *The Robot's Dilemma*. Ablex, 1987.
- [Rimey and Brown, 1992] R. D. Rimey and C. M. Brown. Where to look next using a Bayes Net: Incorporating geometric relations. In *Proc. Second ECCV*, 1992.

```

(reach (b: hand183))
(pick (b: hand183)
(t: short-brick-standing190))
(reach-with-obj (b: hand183)
(s: short-brick-standing190))
(place-on-table (b: hand183)
(t: table182)
(s: short-brick-standing190))
(reach (b: hand183))
(pick (b: hand183)
(t: short-brick-standing186))
(reach-with-obj (b: hand183)
(s: short-brick-standing186))
(place-on-table (b: hand183)
(t: table182)
(s: short-brick-standing186))
(reach (b: hand183))
(pick (b: hand183)
(t: short-brick-standing187))
(reach-with-obj (b: hand183)
(s: short-brick-standing187))
(place-on-table (b: hand183)
(t: table182)
(s: short-brick-standing187))
(reach (b: hand183))
(pick (b: hand183)
(t: short-brick-standing188))
(reach-with-obj (b: hand183)
(s: short-brick-standing188))
(place-on-table (b: hand183)
(t: table182)
(s: short-brick-standing188))
(reach (b: hand183))
(pick (b: hand183)
(t: rectangular-plate189))
(reach-with-obj (b: hand183)
(s: rectangular-plate189))
(place-on-block (b: hand183)
(t: short-brick-standing190)
(t: short-brick-standing186)
(t: short-brick-standing187)
(t: short-brick-standing188)
(s: rectangular-plate189))

```

Figure 7: Recognized operation sequence for "Build Table". Items "b:", "t:" and "s:" denote slot names base, targets, and special.

- [Thibadeau, 1986] R. Thibadeau. Artificial perception of actions. *Cognitive Science*, 10(2):117-149, 1986.
- [Tsuji *et al.*, 1977] S. Tsuji, A. Morizono, and S. Kuroda. Understanding a simple cartoon film by a computer vision system. In *Proc. IJCAI 5th*, pages 609-610, 1977.
- [Whitehead and Ballard, 1990] S. D. Whitehead and D. H. Ballard. Learning to perceive and act. Technical Report 331, Computer Science Dept., Univ. of Rochester, Rochester, NY 14627, USA, June 1990.
- [Wilensky, 1983] R. Wilensky. *Planning and Understanding*. Addison-Wesley, 1983.