Appendix A

Transformation to Extended CORE9

The transformation of short video sequences from the UT-interaction dataset [3], the Mind's Eye dataset, and SBU Kinect Interaction dataset [122] to ExtCORE9$_c$, ExtCORE9$_w$, and ExtCORE9$_{cw}$ descriptions is a two step process as described below:

I. Obtain object tracking data from videos.
II. Compute qualitative spatial relations between objects (using Extended CORE9 framework)

I. Obtain object tracking data from video

The tracking data is obtained by manually labelling objects in extracted frames of the videos for UT-Interaction and Mind's Eye dataset; and available skeleton tracks for the SBU Kinect Interaction dataset[1]. The following steps are involved in manual labeling and thereafter obtaining tracking data of the objects and humans in a video.

1. Keyframes of the videos are extracted using the *ffmpeg* tool[2].

2. For each keyframe extracted in step 1

   (a) Position of the object components within each of the keyframes are manually marked and labeled. This is done by drawing *bounding rectangles* for the object components in each keyframe[3].

[1]Please refer to Chapter 3, Section 3.4.2
[2]The *I-frames* obtained with the *ffmpeg* tool www.ffmpeg.org are used as keyframes.
[3]This is possible with the help of an interactive MATLAB program written by the author.

(b) Coordinates of the bounding rectangles of all components of a pair of extended objects in a single keyframe are stored in a matrix.

(c) The coordinate matrix obtained in the previous step is appended to the end of a list. This list keeps track of the extended object coordinates for the sequence of keyframes seen so far.

3. The complete list of extended object coordinates for the keyframes of a video is the tracking data. This list is stored in the form of a text file.

Figure A-1 shows a screenshot taken during the manual tracking of a *Kick* activity video from the Mind's Eye dataset. The program displays the keyframes of the video one after another. For each keyframe the user is allowed to specify the location of the object components by drawing bounding rectangles.

Figure A-2 shows the matrix obtained for the keyframe shown in Figure A-1. The first column refers to the labels[4] of the components. The remaining four columns correspond to the coordinates of bottom right corner and top left corner of the corresponding bounding rectangle. The first six rows correspond to the first object and next six rows correspond to the second object.

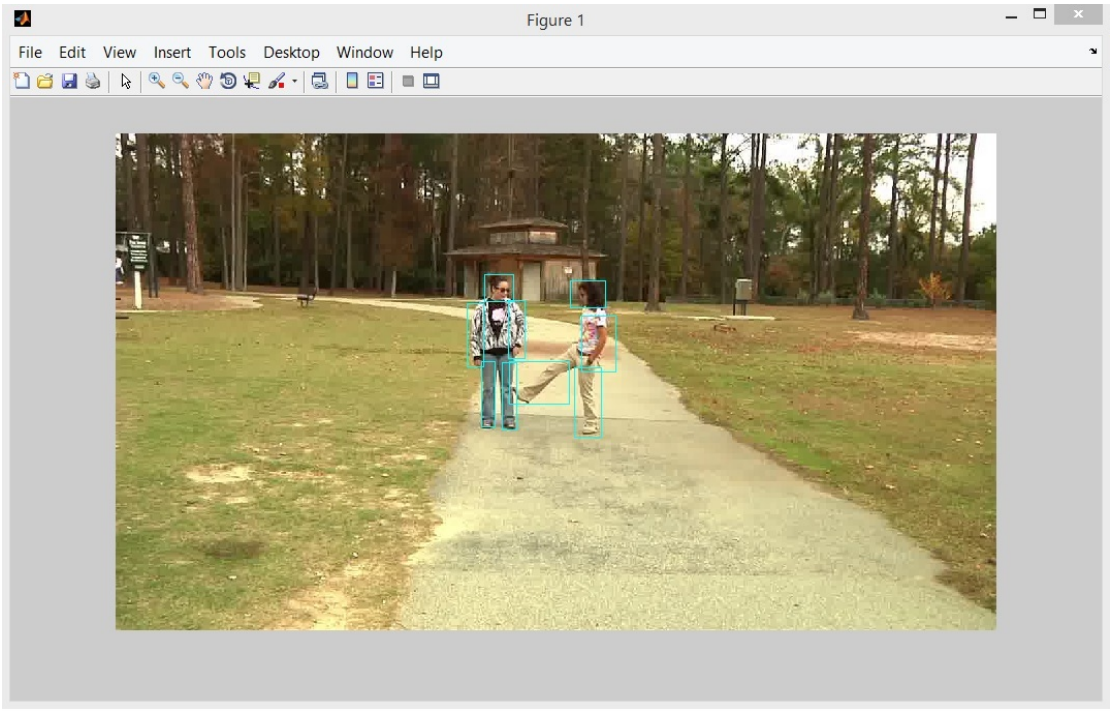

Figure A-1: Screenshot of a keyframe from the Kick activity of the Mind's Eye dataset during the manual tracking.

[4]Labels 1 through 5 refer to *head, left hand, right hand, left leg,* and *right leg.* Label 6 refers to inanimate objects. Label 0 refers to missing values or occluded components.

```
1,537,205,42,39
2,511,247,24,92
3,571,243,24,83
4,532,331,20,95
5,562,333,21,98
0,0,0,0,0
1,662,214,51,39
3,678,265,50,81
4,573,331,87,62
5,668,340,38,101
0,0,0,0,0
0,0,0,0,0
```

Figure A-2: Coordinate matrix of the keyframe shown in Figure A-1

The main thrust being on representation, manual labelling and tracking for video processing is done. It is worth noting that manually obtained tracking data for video has also been used for experimentation in related works [41]. This does not impact the result of such experimentation. Further, it relies purely on obtaining minimal bounding boxes for the components of the objects in each video frame, which can also be done relatively accurately using object tracking methods [133].

II. Compute qualitative spatial relations between objects:

The coordinates of the extended objects obtained in the previous step are used to compute the qualitative spatial relations. Qualitative spatial relations, viz. topological (RCC5 [70]), qualitative direction (CDC [71]), and qualitative distance relations are computed as described in the Extended CORE9 framework. This is done in an *automated* manner using Algorithm 1 described in Chapter 3. The algorithm is implemented in MATLAB. The algorithm computes all qualitative spatial relations between the extended objects in a single keyframe. The algorithm is run for all the keyframes to obtain a complete qualitative description of the video. At this stage, the qualitative description of a video is a *bag of words*, where each *word* is a three-tuple of topological, qualitative distance and qualitative direction relations. The words are converted into strings of the format $top - dir - dis$. The values that top, dir, and dis can have are discussed in Sections 3.3 of Chapter 3 in the thesis. Here, top is the topological relation, dir is the direction relation and dis is the distance relation between two components or whole objects based on Extended CORE9. The output of this step is obtained in the form of three text files -

- The first file consists of only component relations - this is the ExtCORE9$_c$ bag-of-words description of the video.
- The second file consists of only whole relations - this is the ExtCORE9$_w$ bag-of-words description of the video.
- The third file consists of both component and whole relations - this is the ExtCORE9$_{cw}$ bag-of-words description of the video.

Appendix B

Transformation to TAG

The transformation of videos into various TAGs has the following basic steps:

I. Obtain object tracking data from videos
II. Compute qualitative spatial relations between objects and construct TAG

I. Obtain object tracking data from videos:

The tracking data is obtained as discussed in Appendix A. For experiments using TAG, the tracking data already available from the Extended CORE9 experiments are used.

II. Compute qualitative spatial relations between objects and construct TAG :

The qualitative spatial relations are computed in an automated manner using the Extended CORE9 framework as discussed in Appendix A. However, instead of storing the relations as a *bag of words*, they are maintained as edge-labels of a Temporal Activity Graph (TAG).

1. For each keyframe:
   (a) Using the tracking data, compute qualitative spatial relations using Extended CORE9 framework
   (b) Construct a subgraph of TAG for the current keyframe.
      i. The subgraph has vertices corresponding to each component of the two annotated extended objects.
      ii. Spatial edges exist between all components of the first extended object (say a) to all components of second extended object (say b).

iii. Corresponding qualitative spatial relation between the components is the edge label.

(c) Append subgraph of TAG for the current keyframe to the TAG constructed so far using temporal edges.

- Unlabelled temporal edges are constructed between vertices corresponding to same components in the last keyframe of the existing TAG to the current subgraph of the TAG[1].

In Step 1(b), the subgraph of TAG constructed for the current keyframe is termed as *instantaneous TAG subgraph* (insubTAGs) (please refer to Chapter 5, Section 5.2 of the thesis). The qualitative spatial relations computed using Extended CORE9 framework are used as edge labels. The Extended CORE9 framework computes topological, direction and distance relations between all pairs of components. Depending on which relations are used for labelling the spatial edges, four variants of TAGs are constructed as follows:

- **TAG (only topological):** In this variant, only topological relations are used as edge labels. That is, edge labels are of the format *top*, where *top* is the topological relation (RCC5).
- **TAG (only directional):** In this variant, only direction relations are used as edge labels. That is, edge labels are of the format *dir*, where *dir* is the direction relation (CDC).
- **TAG (only distance):** In this variant, only distance relations are used as edge labels. That is, edge labels are of the format *dis*, where *dis* is the distance relation (Qualitative Distance).
- **TAG (topological + directional + distance):** In this variant, topological, directional and distance relations are used as edge labels. That is, edge labels are of the format $top - dir - dis$, where *top* is the topological relation (RCC5), *dir* is the direction relation (CDC) and *dis* is the distance relation (Qualitative Distance).

Figure B-1 shows a subgraph of the TAG that encodes the qualitative spatial relations between the components in the keyframe shown in Figure A-1. The figure shows TAG (topological + directional + distance). The data structure used for the TAG is a *list of reduced adjacency matrices*. As discussed in Chapter 5 of

[1]A keyframe is interpreted as a time-point herein.

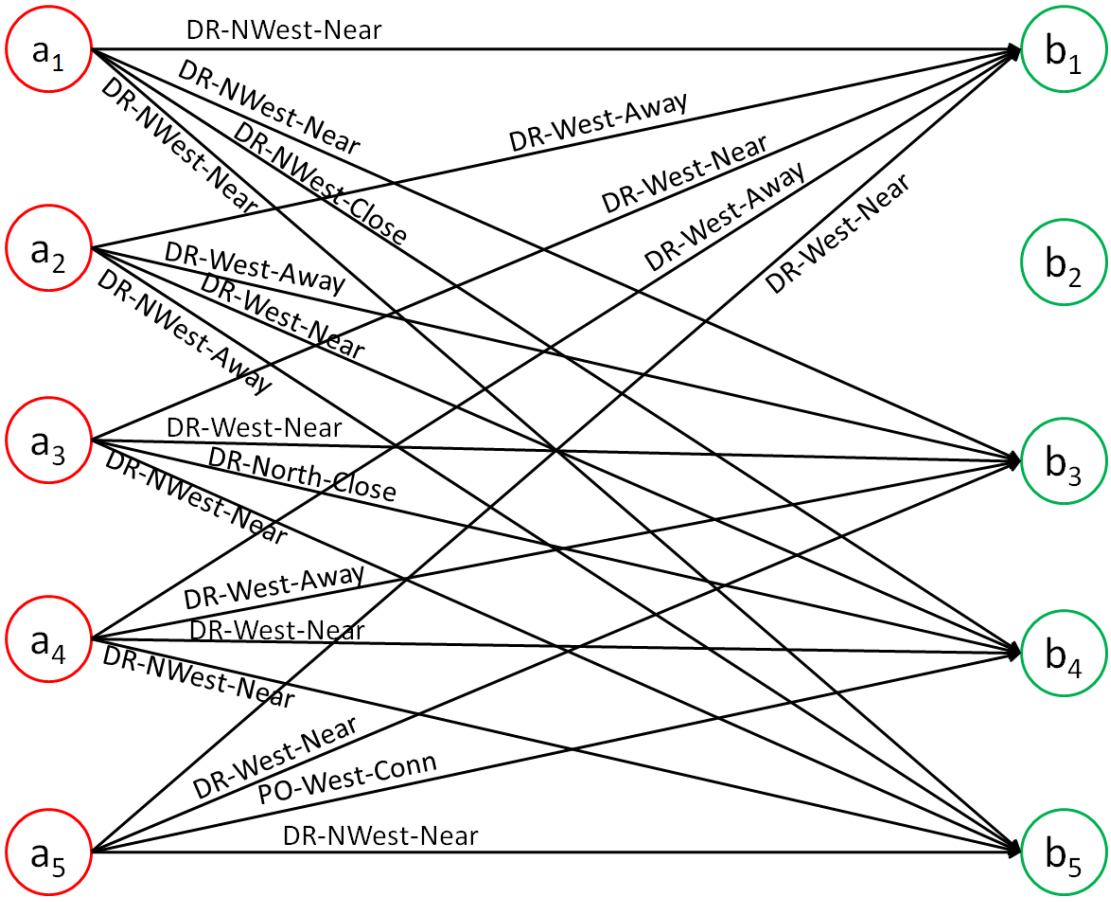

Figure B-1: Subgraph of a TAG (topological + directional + distance) constructed for the keyframe of the Kick activity shown in Figure A-1

the thesis (Section 5.3.1), a *reduced adjacency matrix* is used for maintaining the *insubTAGs* representing the spatial relations at a single time-point.

Figure B-2 shows the reduced adjacency matrix corresponding to the insubTAG shown in Figure B-1. In this reduced adjacency matrix representation, the position corresponding to (a_i, b_j) stores the qualitative spatial relation between component i of the first object with component j of the second object.

	b_1	b_2	b_3	b_4	b_5	b_6
a_1	DR-NWest-Near	X-X-X	DR-NWest-Near	DR-NWest-Close	DR-NWest-Near	X-X-X
a_2	DR-West-Away	X-X-X	DR-West-Away	DR-West-Near	DR-NWest-Away	X-X-X
a_3	DR-West-Near	X-X-X	DR-West-Near	DR-North-Close	DR-NWest-Near	X-X-X
a_4	DR-West-Away	X-X-X	DR-West-Away	DR-West-Near	DR-NWest-Near	X-X-X
a_5	DR-West-Near	X-X-X	DR-West-Near	PO-West-Conn	DR-NWest-Near	X-X-X
a_6	X-X-X	X-X-X	X-X-X	X-X-X	X-X-X	X-X-X

Figure B-2: Reduced Adjacency Matrix representation for the keyframe shown in Figure A-1. Each entry is of the form $top - dir - dis$, where top is the topological relation, dir is the qualitative direction relation, and dis is the qualitative distance relation computed using Extended CORE9. Entries of the form $X - X - X$ correspond to unavailable data for missing or occluded components.

In Step 1(c), to construct the complete TAG, temporal edges connect vertices representing the same component at consecutive time-points. Temporal edges do not have labels and are assumed to exist between all vertices representing the same component for all consecutive time-points. For example, it is implicitly understood that there is a temporal edge between vertex representing component 1 at time point 1, to vertex representing component 1 at time point 2 and so on. Thus, a list of reduced adjacency matrices can be used to maintain the complete TAG.

Appendix C

Experimentation Details

The system is implemented in a way that it learns in the training phase from known videos and recognizes the contents of unknown videos. There are three distinct set of experiments investigating Extended CORE9, TAG Kernel and TAG Grammar.

For the first set of experiments reported in Chapter 3, involving Extended CORE9, four different classifiers are trained on a Extended CORE9 bag-of-words representation of the activities. The system is tested on unseen videos represented as Extended CORE9 bag-of-words. For the second set of experiments reported in Chapter 4, a SVM using the TAG kernel defined in the thesis is trained on activities represented as TAGs; the system is tested on TAG represented unseen videos. For the final set of experiments, TAG Grammars are induced from activities represented as TAGs. Videos are transformed into TAGs as detailed in Appendix B. A particular TAG is a representation of some video activity. On the other hand, the TAG grammar is a *model* of an activity class. The TAG grammar is learned from the TAG representations of the video activity instances; TAG grammar does not have any role in the automatic transformation of videos into TAGs. The TAG grammar is used for recognition of activities.

After the videos are converted into TAGs as discussed in Appendix B, the TAG grammar is learned in an automated manner. The TAG grammar learning algorithm is discussed in Chapter 5 of the thesis (Algorithms 2 and 3)[1] Figure 5-2 in Chapter 5 shows a block diagram of the complete sequence of steps involved in the learning and recognition of activities using TAG Grammar. This is the set of experiments reported in Chapter 5. As shown in the block diagram, training and test sets are different. In the Learning phase the videos in the training set are converted to TAG. The TAG grammars are induced from these TAG representations using Algorithm 2 detailed in Chapter 5 (Please refer to Section 5.3).

[1]These algorithms are implemented by the author in MATLAB.

Results reported in the thesis are based on a 10-fold cross-validation. This involves the following standard steps:

1. The dataset is transformed into the appropriate representation, i.e., Extended CORE9 *bag of words* or TAG.
2. The dataset is partitioned into 10 equal sub sets.
3. For each unique set
   (a) Consider the unique set as test set.
   (b) Consider the remaining nine sets as Cross-validation training set,
   (c) In case of Extended CORE9 *bag of words* and TAG kernel based approach, train the classifier on the training set. In case, of the TAG grammar based approach, learn TAG grammars for different activity classes using the training set.
   (d) Use the trained classifiers or the learned TAG grammars over the test set and evaluate precision, recall, f1-scores, and classification accuracy.
   (e) Retain precision, recall, f1-scores, and classification accuracy for the current evaluation
4. Compute precision, recall, f1-scores, and classification accuracies by averaging the individual values derived in all the 10 cases of cross validation.

The aforementioned evaluation process is completely automated and is repeated for all three datasets - Mind's Eye Dataset, UT Interaction Dataset, and SBU Kinect Interaction Dataset.