Common-sense reasoning for human action recognition

Martínez del Rincón, J., Santofimia, M. J., & Nebel, J.-C. (2013). Common-sense reasoning for human action recognition. *Pattern Recognition Letters*, *34*(15), 1849-1860. https://doi.org/10.1016/j.patrec.2012.10.020

Published in:
Pattern Recognition Letters

Document Version:
Peer reviewed version

Common Sense Reasoning for Human Action Recognition

Jesús Martínez del Rincón[#], Maria J. Santofimia*, Jean-Christophe Nebel[#]

[#]Digital Imaging Research Centre, Kingston University, London, KT1 2EE, UK

* Department of Technology and Information Systems, Computer Engineering School, University of Castilla-La Mancha, Ciudad Real, Spain

Abstract

This paper presents a novel method combining computer vision and artificial intelligence techniques for action recognition. The proposed methodology is decomposed into two stages. First, a machine learning based algorithm – bag of words- gives a first estimate of action classification from video sequences. Those results are passed to a common sense reasoning algorithm, which allows analysing, selecting and correcting the initial action estimates. Experiments are performed in realistic conditions, where poor recognition rates by the machine learning technique are significantly improved by the second stage based on reasoning. This demonstrates the value of integrating common sense reasoning into a computer vision pipeline.

Keywords: Common sense reasoning, artificial intelligence, action recognition, bag of words, computer vision

1. Introduction

In the last decade, the automated recognition of human actions from video sequences has become an essential field of research in computer vision. Not only

Corresponding author: Jesus Martinez del Rincon
Email: Jesus.Martinezdelrincon@kingston.ac.uk
Telephone: +44 (0) 20 8417 7159
Postal Address: Penrhyn Road, Kingston upon Thames, Surrey KT1 2EE

does it have applications in video surveillance, but also in indexing of film archives, sports video analysis and human-computer interactions. However, the task of action recognition from a single video remains extremely challenging due to the huge variability in human shape, appearance, posture, the individual style in performing some actions, and external contextual factors, such as camera view, perspective and scene environment.

During the last few years, thanks to the availability of many datasets suitable for training action recognition algorithms, the field has made enormous progress to the point that the automatic annotation of the KTH (Schuldt et al., 2004) and Weizzman (Blank et al., 2005) databases is now considered solved. For more complex data, i.e. IXMAS (Weinland et al., 2006) and UT-Interaction (Ryoo and Aggarwal, 2009), accuracy rates around 80% are now claimed by state-of-the-art approaches (Waltisberg et al., 2010; Weinland et al., 2010; Nebel et al., 2011). Unfortunately, all those action recognition experiments are conducted with videos that are not representative of real life data, which led a recent review to conclude that none of existing techniques would be currently suitable for real visual surveillance applications (Nebel et al, 2011). This is further confirmed by the poor performance, obtained on videos captured in uncontrolled environments, such as Hollywood 1 and 2 datasets (Laptev et al. 2008) and Human Motion DataBase (HMDB51) (Kuehne et al., 2011), where accuracies are 32%, 51% and 20% respectively (Kuehne et al., 2011). In addition, these challenging datasets only display a fraction of the complexity exhibited by the real world, e.g. at most 51 different actions are considered. Consequently, usage of video-based action recognition remains a very distant aspiration for most actual applications.

On the other hand, the human brain seems to have perfected the ability to recognise human actions despite their high variability. This capability relies not only on acquired knowledge, but also on the aptitude of extracting information relevant to a given context and logical reasoning. In contrast, machine learning based action recognition methodologies tend to learn isolated actions from a set of examples. Although only a few and limited attempts to introduce contextual information have been made (Waltisberg et al., 2010; Chen and Nugent, 2009; Akdemir et al. 2008; Vu et al. 2002; Ivano and Bobick, 2000), their performance supports the idea that action recognition can benefit greatly from combining traditional computer vision based algorithms with knowledge based approaches.

In this paper, we propose a novel method relying on common sense reasoning and contextual information which allows analysing, selecting and correcting annotation predictions made by a video-based action recognition framework. The presented approach is decomposed into two stages. First, a classic action recognition algorithm classifies actions independently according to similarity to the training set. Secondly, results are refined using reasoning. More specifically, contextual information is exploited using common sense reasoning.

2. Relevant work

a. Video-based Human Action Recognition

Video-based activity recognition algorithms can be classified into two different classes: machine learning and knowledge based techniques. The first and main category includes action descriptors based on Hidden Markov Models (Vezzani et al., 2010; Kellokumpu et al, 2008; Martinez et al. 2009; Ahmad and Lee, 2008;

Weinland et al., 2007), Conditional Random Field (Zhang and Gong, 2010; Natarajan and Nevatia, 2008; Wang and Suter, 2007), Bag of Words (Laptev et al., 2008; Liu and Shah, 2008; Matikainen et al., 2010; Ta et al., 2010; Liu et al., 2008; Kovashka and Grauman, 2010) and low dimension manifolds (Wang and Suter, 2007b, 2008; Fang et al. 2009; Jia and Yeung, 2008; Blackburn and Ribeiro, 2007; Richard and Kyle, 2009; Turaga et al. 2008; Lewandowski et al. 2010, 2011). Since those approaches do not include any reasoning capability, their efficiency relies on a training set which is supposed to cover the variability of all actions present in the target videos. Given that this condition can only be valid in the most controlled scenarios, it has been proposed to extend these techniques by adding some form of reasoning based on either rules or logic.

The inclusion of reasoning has been sparsely used and mostly for specific applications. It should be noted it is particularly popular in intelligent surveillance for the detection of unusual events (Makris et al. 2008). Since training data do not exist to define those events, rules and reasoning are the only available tools. Usually, activities which do not match those present in the training set are classified as unusual. In the most specific field of action recognition, reasoning rules have proved particularly successful when dealing with interactions between subjects (Waltisberg et al. 2010). Indeed, following initial action recognition on each character individually using a Random Forest framework, analysis of those actions allows inferring the nature of their interaction. As reported by Waltisberg et al. (2010), this scheme outperforms the standard approach which deals with all characters at once and is the current state of the art on the UT-Interaction dataset (Ryoo and Aggarwal, 2009). These results support our hypothesis that additional knowledge and reasoning will lead to better performance.

The second class of video-based activity recognition algorithms exploits a common knowledge-base or ontology of human activities to perform logical reasoning. Since ontology design is empirical in nature and labour intensive - symbolic action definitions are based on manual specification of a set of rules -, current ontologies are only suitable for very specific scenarios. In the field of video surveillance, ontologies have been proposed for analysis of social interaction in nursing homes (Chen et al., 2004), classification of meeting videos (Hakeem and Shah, 2004) and recognition of activities occurring in a bank (Georis et al., 2004). However, there is a need for an explicit commonly agreed representation of activity definitions independently of domain and/or algorithmic choice. Such common knowledge base and its exploitation through rules would facilitate portability, interoperability and sharing of reasoning methodologies applied to activity recognition. Several attempts have been made to design ontologies for visual activity recognition in a more systematic manner (Akdemir et al., 2008, Hobbs et al., 2004, Francois et al, 2005) so that they can cover different scenarios, e.g. both bank and car park monitoring (Akdemir et al., 2008). However, they remain limited to a few domains - up to 6 (Hobbs et al., 2004).

b. Common Sense Reasoning

Within the artificial intelligence (AI) community, the usage of video as information source for reasoning has not been extensively applied (Moore et al., 1999; Duong et al., 2005). This is due to the lack of robustness and consistency of video features in real world scenarios, where the huge variability of the conditions impact considerably on activity recognition. As a consequence, AI researchers have focused on using sensors which are more reliable and consistent, but more intrusive, sensors to gather an actor's behavioural information (Wang et al. 2007c). They include wearable sensors based on inertial measurement units (e.g. accelerometers, gyroscopes, magnetometers) and RFID tags attached to the actors and/or to objects. In such set-up, complex reasoning is possible and successful artificial intelligence approaches have flourished (Wang et al., 2007c; Philipose et al., 2004; Tapia et al., 2004). However, most of these sensors are not suitable in most real life applications due to either their intrusive nature, e.g. subjects may refuse to wear them, or technical factors, such as size, ease of use and battery life.

Among the AI approaches which could be considered for video based human action recognition, commonsense, probabilistic and ontological reasoning, as described in the previous subsection, are of particular interest. Ontological languages such as OWL (Dean et al., 2011a) and RDF (Dean et al., 2011b) use a syntax that imposes severe restrictions in the type of information that can be represented. First, relationships involving more than two entities cannot be considered since they may lead to hold a-priori inconsistent information, which is not allowed in this methodology. Secondly, since reasoning is limited to checking the consistency of the knowledge base, new information cannot be inferred. Both commonsense and probabilistic reasoning are able to address those limitations. However, their nature is very different since they can be classified as techniques based on either qualitative or quantitative reasoning. A weakness of quantitative reasoning comes from the complexity of estimating accurate probabilities for activities of interest: in practice it is unfeasible when dealing with unconstrained and realistic scenarios (Kuipers, 1994). On the other hand, qualitative reasoning has the ability of considering causality and expected behaviour based on logics, i.e. reasoning can provide explanations rationalising or motivating a given action, whereas probabilistic reason can only support decisions according to probability associated to actions.

As a consequence, common sense reasoning (McCarthy, 1968, 1979; Minsky, 1986; Lenat, 1989, 1990) appears particularly suited to video based human action recognition. It provides the capability of understanding the context situation, given the general knowledge that dictates how the world works, which allows correcting mistakes made by the video analysis system. McCarthy proposes an approach to build a system with the capability to solve problems in the form of an “advice taker” (McCarthy, 1968). In order to do so, he reckons that such an attempt should be founded in the knowledge of the logical consequences of anything that could be told, as well as the knowledge that precedes it. In that work, he postulates that “a program has common sense if it automatically deduces from itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows''. Following McCarthy and Minsky’s studies (McCarthy, 1968; Minsky, 1986), it appears a way of enhancing systems with the capability to understand and reason about the context is by introducing commonsense knowledge similar to that humans hold.

In this work, we propose the integration of commonsense reasoning within a video human activity recognition framework in order to improve accuracy. First, a machine learning based action recognition algorithm processes videos to generate data appropriate for logical inferences. Consequently, video data become a suitable information source for reasoning. Secondly, common sense reasoning increases accuracy of the computer vision algorithm by introducing general and context-independent knowledge. This addition should allow usage of video based systems within real life applications.

3. Novel action recognition framework

a. Principles

We propose a novel two-stage framework where initial action predictions made by a machine learning approach are analysed, refined and, possibly, corrected by common sense reasoning.

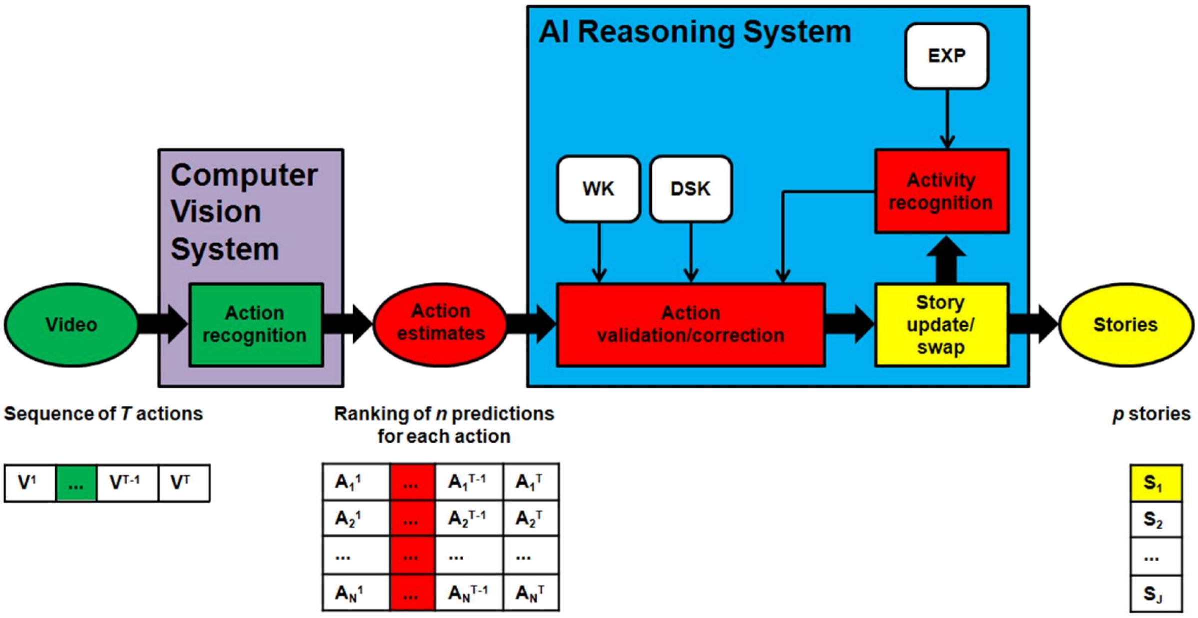


Figure 1: Action recognition framework

Given a video, V, which can be divided into a sequence of T actions and a computer vision system (CVS) trained to recognise N types of actions, each action, V^t, is processed independently and is associated to an action estimation vector, A^t, which ranks the N types of actions according to their similarity to V^t. Eventually, the CVS generates an action estimation matrix, A, of dimensions (T x N), where A_j^t represents the j^{th} most likely type of the t^{th} action occurring in the video. Each action estimate generated by the CVS is passed as input to the AI reasoning system (AIRS) which produces, in an online manner, J stories, S_j. These stories are generated and updated according to every new estimate A^t.

In this paper, we define a 'story' as a coherent list of action types describing a video of interest. Coherence is defined by respect to both world and domain specific knowledge, WK and DSK respectively. Selection of action types relies on common sense reasoning applied to the action estimations A, and possible recognition of activities defined in the expectation knowledge, EXP. Note that a story may contain 'unknown action' labels when, for a given action, none of the estimations allows coherent annotation. Stories are ordered by the AIRS and the most likely one is always first, in the same way that actions have been ordered and prioritised by the CVS.

The AIRS processes every action estimation vector, A^t, according to the J stories S_j existing at $t\text{-}1$. First, the validity of each action estimates A_i^t is verified within the context of each story S_j using knowledge contained in WK and DSK. This is done inside the block Action validation/correction depicted in Figure 1. Secondly, if the sequence of previous actions stored in S_j led to the recognition by EXP of an activity (Figure 1, block Activity Recognition) which expected a specific action type in order to be completed, and if that type is not present in A^t, a correction of A^t is performed, i.e. the expected type is added to the story S_j instead of A^t. Finally, each valid action of A^t updates an existing story (Figure 1, block story update/swap). If a valid action cannot be allocated to a story, a new story is created. Since during the process, the most likely action estimates have priority to be allocated to the first stories, S_1 is the story which is the most likely to describe accurately the video of interest. However, if any other S_j shows a more likely storyline, the position of S_1 as 'main story' may be swapped with S_j (Figure 1, block story update/swap).

We illustrate some of the reasoning performed by AIRS with an example, see Figure 2: an activity ('Getting up') incompatible with the current story (S_1) is rejected according to the world and domain specific knowledge; valid actions (‘Throwing’ & ‘Sitting down’) are assigned to parallel stories (S_2 and S_3); an activity (‘Reading’) is recognised based on expectations, consequently the expected action (‘Sitting down’) is prioritised.

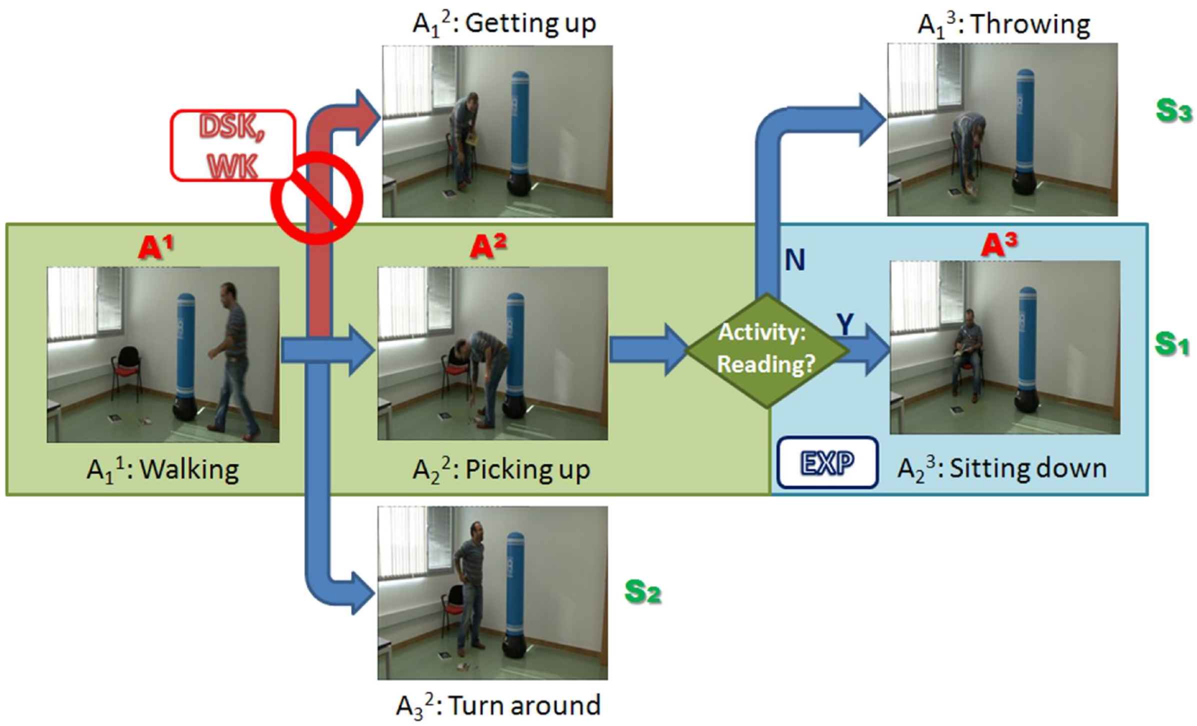


Figure 2: Example of reasoning performed by AIRS. Blue and red arrows represent, respectively, valid and invalid actions. Green box depicts the sequence of action which led to the recognition of an activity (reading) based on expectations. Blue box shows the expected action (sitting down).

b. Common sense reasoning algorithm

The AIRS assigns and evaluates correspondences between action estimations in vector A^t and the stories S existing at t-1. The validity of each action estimate A_i^t is verified sequentially within the context of the main story S_1 using knowledge contained in WK and DSK. Once action allocation, if any, has been completed for the main story, the same process is followed for all the other stories S_j using the remaining action estimates. This double sequentiality in the assignment of actions to stories deals with the fact that both stories and actions are ordered, where the first actions/stories are always the most likely.

The *n* first action estimates are all considered as possible alternatives. Therefore, new stories are created if they do not fit any of the existing ones. The rationale behind this is that, although the first estimate provided by the CVS is not always correct, the CVS is quite robust since the correct action is likely to be present among the first *n* estimates (see 'Experimental results' section). During the allocation process of a given time step, some stories may not be allocated to any action, if none of the available action estimates is valid in their context according to WK and DSK.

A second level of reasoning is introduced by exploiting the concept of activity recognition. This is modelled in our system through the expectation knowledge, EXP. For each story S_j, if the sequence of previous actions leads to the recognition of an activity by EXP, the next action assigned to the story S_j must match the expected one, *eA*. In case where the expected action type is not present in A^t, A^t is corrected by including *eA* in the estimate vector so that *eA* can be assigned to story S_j. This mechanism provides a higher level of reasoning, going further than the validation mechanism provided by the DSK and WK, which allows correcting estimate failures of the CVS. However, in order to avoid over-reasoning errors, corrections are introduced only when, in addition to validation, a unique activity is recognised, i.e. when there is no doubt regarding the type of the expected action.

Through the previously described process, the AIRS gives priority to the most likely action estimates in their allocations to the first stories. As a consequence, the AIRS output is an ordered set of stories, where S_1 is the story which is the most likely to describe accurately the video of interest.

However, the accuracy of the CVS may depend of the nature of the action and vary over time during video processing, which may lead to the correct estimates to be lower in the action estimation vectors. Consequently, after a while S_1 may not contain the most likely story. The AIRS addresses this issue using a story swapping mechanism. When the AIRS is able to allocate systematically actions to a given story S_j and activities kept being recognised according to the expectations, this story is accepted as the main story and swapped with S_1. Empirical experimentations have shown that the story swapping mechanism should be triggered when a story displays two consecutive activity recognitions, *TH=2*.

This reasoning algorithm is presented through the following pseudo code. First, the main variables are defined. Then, the core of the algorithm is detailed. Finally, the main functions are described. Note that functions are colour-coded to allow better readability of the algorithm.

```
////////////////////////////////////////////////////////////////////////////
// INPUT
////////////////////////////////////////////////////////////////////////////
// Expert systems
Expert DSK,WK,ExP;
//An action is a primitive
Action eA;          // expected action
Action At[N];       // alternative actions predicted for time t,
                    // At are ranked according to CVS's prediction confidence
Int N;              // number of alternative actions at time t
//A story is a list of actions
Story S[J];         // existing stories
Int J=1;            // number of existing stories, one starts with 1 story
S[1]=null;          // the initial story is empty

//Each story is associated to a list of possible activities containing
future actions for the next time t
Typedef Action[] Activity;
Activity PossibleActiv[][J]=[ ALL ][J]; // set of activities, initially all
                                      // activities are possible
Int expect_fulfill[J]=zeros(1,J); // story counter for swapping mechanism
```

```
/////////////////////////////////////////////////////////////////////////////
// MAIN
/////////////////////////////////////////////////////////////////////////////
for t=1:Inf              // for each time step
   N=length(A^t);                         // number of alternative actions
   Bool assigned_action[N]=zeros(1,N);    // no action is assigned
   J=length(S);                           // number of existing stories
   Bool updated_story[J]=zeros(1,J);      // no story has been updated
   for i=1:N             // for each alternative action
      // integration of action i into an existing story
      for j=1:J          // for each existing story
         if (updated_story(j)==0)         // if story j is available
            // activity recognition process
            eA=f_activity_recognition(PossibleActiv(j));//expected activity
            if (eA!=null)                 // if activity recognised   //
            story updating process
               [PossibleActiv(j),S(j)]= f_story_update
                                     (eA,PossibleActiv(j),S(j),ExP);
               updated_story(j)=1;        // story j is updated
               // action allocation process
               assigned_action=f_action_allocation(assigned_action,eA,A^t);
               // story swapping process
               [S,expect_fulfill]=f_storySwapping(S,expect_fulfill,j);
            else                          // no activity is recognised
               if (assign_action(i)==0)   // if action i is available
                  // action validation process
                  if f_action_validation(A^t(i),DSK,WK,S(j))//if A^t(i)valid
                     // story updating process
                     [PossibleActiv(j),S(j)]=f_story_update
                                     (A^t(i),PossibleActiv(j),S(j),ExP);
                     updated_story(j)=1;        // story j is updated
                     // action allocation process
                     assign_action(i)=1;        // action i is allocated
                  end
               end
            end
         end
      end
      // integration of non-assigned action i into a new story
      if (assign_action(i)==0) // if action i is available
         // action validation process
         if f_action_validation(A^t(i),DSK,WK,S(j)) // if action i is valid
            // story creation process
            [PossibleActiv,S,expect_fulfill]=f_story_creation
                                     (S,A^t(i),ExP,expect_fulfill);
            J=length(S);                        // update number of stories
            updated_story(J)=1;                 // story J is updated
            // action allocation process
            assign_action(i)=1;                 // action i is allocated
         end
      end
   end
end
```

Expectations are checked at each given time *t*, for each current story (function `f_activity_recognition`). If the number of current expected activities is only one, the nature of the ongoing activity is known. Therefore, the function is able to return the expected type of the next action, *eA*.

```
function [Action a]=f_activity_recognition(Activity pred)
      if (size(pred)==1)
            return pred(1);
      else
            return null;
      end
```

If any of the *n* observed actions of A^t matches *eA*, this action is set as allocated to avoid inclusion in any other story (function `f_action_allocation`).

```
function [bool b]=f_action_allocation(bool b, Action a, Action[] v)
      for i=1:size(v)
            if(v(i)==a)
                  b=1;
            end
      end
      return b;
```

When an action has been judged suitable to be added to a story, the current story is updated (function `f_story_update`). This also involves updating the list of possible ongoing activities, i.e. knowledge about possible actions for time t+1: `PossibleActiv(j)`. This is achieved by, first, retrieving all expected activities in the knowledge of action *a* at time *t*, *p2*, (function `retrieve_expected_activities`) and, then, by finding the intersection between this list and the one predicted for time *t*, *p*, (function `intersection`). If no intersection exists, i.e. either CVS has failed or reasoning has been erroneous, since it is not possible to distinguish the source of the failure, expected activities are reset to *p2* to avoid propagating errors.

```
function [Activity p,Story s]=f_story_update
                                    (Action a, Activity p, Story s, ExP e)
      Activity p2=null;
      s=[s a];                      // add action a to current story s
      p2=retrieve_expected_activities(e,a);
      p=intersection(p,p2);         // new list of expected activities
      if (size(p)==0)
            p=p2;
      end;
      return [p,s];
```

If the activity recognition algorithm was able to detect unequivocally the nature of an ongoing activity within a story, S_j, confidence in that story is increased. This is stored in the variable `expect_fulfill`. The valued of that variable is evaluated during the story swapping mechanism (function `f_storySwapping`). If it shows that the story S_j has consecutively recognised activities (in our case twice `TH=2`), the story S_j is swapped with S_1 and becomes the main story, i.e. the most likely one.

```
function [Story s[], int[] f]=f_storySwapping(Story s[], int[] f, int indx)
      Story s_tmp;
      f(indx)++;
      if f(indx)>=TH
      // s(index) is moved as top story and all the others are shifted down
            s = [s(indx) s(1: indx-1) s(indx-1:end)};
            f = zeros(1,J);
      end
      return [s,f];
```

If the activity recognition mechanism does not detect any ongoing activity or several activities are possible, action allocation only relies on action validity. This is evaluated according to the action global coherence with the world WK and the domain specific knowledge DSK within the context of a story (function `f_action_validation`).

```
function bool=f_action_validation(Action a,DSK d,WK w,Story s)
      return validate(a,d,s,w);
```

If an action is judged as valid, the action is assigned to the story and expected activities are updated (function `f_story_update`). After the assignment, boolean vectors, `assigned_action` and `updated_story`, are updated to make sure that each action is assigned at most to one story and that each story is not updated more than once for a given time t.

Finally, if an action is valid but has not been assigned to any current story, a new story is created (function `f_story_creation`).

```
function [Activity p, Stories s, int[] f]=f_story_creation(Stories s,
Action a, EXP e, Activity p, int[] f)
      Activity Activnew=[All];
      Story Snew=[];
      [Activnew, Snew]=f_story_update(a,Activnew,Snew,e);
      J=J+1;
      s(J)=Snew;
      p(J)= Activnew;
      expect_fulfill(J)= 0;
      return [p,s];
```

4. Implementation

a. Computer vision based action recognition

Although computer vision based action recognition has been a very active field of research, only a few approaches have been evaluated on view independent scenarios. Accurate recognition has been achieved using multi-view data with either 3D exemplar-based HMMs (Weinland et al., 2007) or 4D action feature models (Yan et al. 2008). But, in both cases performance dropped significantly in a monocular setup. This was addressed successfully by representing videos using self-similarity based descriptors (Junejo et al., 2008). However, this technique assumes a rough localisation of the individual of interest which is unrealistic in many applications. Similarly, the good performance of a SOM based approach using motion history images is tempered by the requirement of segmenting characters individually (Orrite et al. 2008). More recently a few approaches have produced accurate action recognition from simple extracted features: two of them rely on a classifier trained on bags of words (Kaaniche and Bremond, 2010; Liu et al. 2008) whereas the other one is based on a nonlinear dimensionality reduction method designed for time series (Lewandoski et al. 2010).

Among those approaches, the Bag of Words (BoW) framework is particularly attractive since, not only it is one of the most accurate methods for action recognition, but its computational cost is low. Moreover, BoW can be applied directly on video data without the need of any type of segmentation. The versatility of that framework has been demonstrated on a large variety of datasets including film-based ones (Laptev and Perez, 2007). Consequently, in this study, we decided to base the computer vision system of our action recognition framework on a BW methodology.

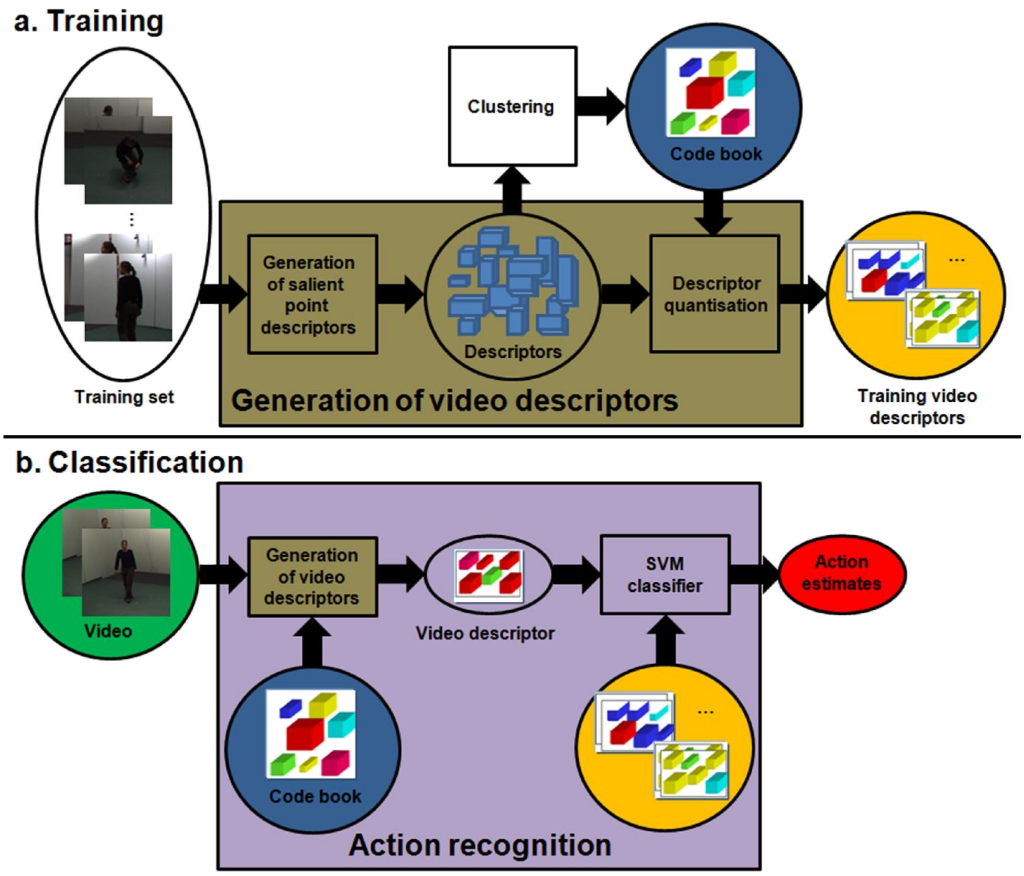


Figure 3: BoW framework: a) Training and b) classification pipelines

BoW is a learning method which was used initially for text classification (Joachims, 1998). It relies on, first, extracting salient features from a training dataset of labelled data. Then, these features are quantised to generate a code book which provides the vocabulary in which data can be described and analysed. Here, we based our implementation on that proposed by (Csurka et al., 2004).

The BoW training stage aims at, first, producing a codebook of feature descriptors and, secondly, generating a descriptor for each action video available in the training set, see Figure 3 a). The training pipeline starts by detecting salient feature points in each video using a spatio-temporal detector (Harris 3D) and describing each individual point by a histogram of optical flow (STIP) (Laptev, 2005). Once feature points are extracted from all training videos, the k-means algorithm is employed to cluster the salient point descriptors into k groups, where their centres are chosen as group representatives. These points define the codebook which is then used to describe each video of the training set. Finally, those video descriptors are used to train SVM classifiers – one per action of interest - with a linear kernel.

In order to recognise the action performed in a video, Figure 3 b), salient feature points are first detected. Then, their descriptors are quantified using the codebook in order to generate a video descriptor. Finally, the video descriptor is fed into each SVM classifier, which allows quantifying the fit between the video and each trained action type. Therefore, an action estimation vector *A* can be generated where action types are ranked according to their fit.

b. Knowledge-Base System for Common Sense Reasoning

Automating common sense reasoning requires an expressive-enough language, a knowledge base and a set of mechanisms capable of processing this knowledge to check consistency and infer new information. A few knowledge-based approaches offer such features, i.e. Scone (Chen and Fahlman, 2008; Fahlman, 2006), Cyc (Lenat et al. 1989, 1990), WordNet (Fellbaum, 1998) or ConceptNet (Eagle et al., 2003). Among them, the open-source Scone project is of particular interest since, instead of placing its focus on collecting commonsense knowledge, it provides efficient and advanced means for accomplishing search and inference operations.

The main difference between this and other approaches lies in the way in which search and inference are implemented. Scone adopts a marker-passing algorithm (Fahlman, 2006), which is not a general theorem-prover, but is much faster and supports most of the search and inference operations required in commonsense reasoning: inheritance of properties, roles, and relations in a multiple-inheritance type hierarchy; default reasoning with exceptions; detecting type violations; search based on set intersection; and maintaining multiple, immediately overlapping world-views in the same knowledge base. In addition, Scone provides a multiple-context mechanism which emulates humans' ability to store and retrieve pieces of knowledge, along with matching and adjusting existing knowledge to similar situations.

In our framework, the algorithm described in section 3b was implemented using Scone in order to encode formal definitions and their applications for WK, DSK and EXP. It is important to note that, although we took advantage of the proposed multi-context mechanism (Chen and Fahlman, 2008), we exploited it for a usage it was not originally intended for, extending its application for a wider purpose. In particular, we propose the usage of multi-context for the management of alternative stories describing coherent explanations of the video of interest.

The three sources of knowledge exploited in our implementation, i.e. WK, DSK and EXP, are described below:

1. World knowledge, WK, comprises all relevant commonsense knowledge that describes "how the world works". This information is independent of the application domain, in the sense that it only considers general knowledge rather than specific or expert knowledge about a specific field. As an example, we provide below the description of the implications of performing the action of 'scratching the head'.

```
(new-event-type {scratch} '({event})
:roles
((:type {scratcher} {animated thing})
(:type {scratched thing} {thing})))
```

```
(new-event-type {scratch head}
'({scratch} {action})
:roles
((:rename {scratched thing} {scratched head})
(:rename {scratcher} {scratcher hand}))
:throughout
((new-is-a {scratcher hand} {hand}))
:before
((new-statement {scratcher hand} {approaches} {scratched head})
(new-not-statement {scratcher hand} {is in direct contact to}
{scratched head}))
:after
((new-statement {scratcher hand} {is in direct contact to}
{scratched head})))
```

2. Domain specific knowledge, DSK, describes a given application domain in terms of the entities that are relevant for that specific context, as well as, the relationships established among those. The description of an element "punching ball" as part of the layout of a specific room is an example of domain specific information.

```
(new-type {bouncing element} {thing})
(new-type {punching ball} {thing})
(new-is-a {punching ball} {bouncing element})
(new-indv-role {punching ball location} {punching ball} {location})
(new-statement {punching ball} {is in} {test room})
(new-statement {punching ball} {rests upon} {test room floor})
```

3. Expectations, EXP, consist in sequences of actions that are expected to happen one after the other. It encapsulates logical concepts such as causality, motivation and rationality, which are expected in human action recognition. For example, in a waiting room context, if a person picks up a magazine, that person is expected to sit down and read the magazine. Expectations are part of the domain specific knowledge since described behavioural patterns are context specific.

```
(new-indv {picking up a book for reading it} {expectations})
(the-x-of-y-is-z {has expectation} {picking up a book for reading it} {walk
towards})
(the-x-of-y-is-z {has expectation} {picking up a book for reading it} {pick
up})
(the-x-of-y-is-z {has expectation} {picking up a book for reading it} {turn
around})
(the-x-of-y-is-z {has expectation} {picking up a book for reading it} {sit
down})
```

```
(the-x-of-y-is-z {has expectation} {picking up a book for reading it} {get up})
```

5. Experimental results

i. Dataset and Experimental Setup

In order to perform action recognition experiments which are relevant to real life applications, videos under study should display realistic scenarios. In addition, a suitable training set must be available, i.e. it must be able to cover a variety of camera views so that recognition is view-independent and the set should include a sufficiently large amount of instances of the actions of interest. These instances must be not only annotated but perfectly segmented and organised to simplify the training.

The only suitable training sets which fulfil these requirements are IXMAS (Weinland et al., 2006) and Hollywood (Laptev et al. 2008), as stated in the introduction. Whereas the Hollywood dataset is oriented towards event detection which includes significant actions but largely independent from each other (drive car, eat, kiss, run...), IXMAS is focused on standard indoor actions which allows providing quite an exhaustive description of possible actions in this limited scenario. Therefore, IXMAS actions may be combined to describe simple activities, i.e. sit down-get up, pick up-throw, punch-kick and walk-turn around, and eventually provide complete representations of sets of actions performed by individual, i.e. recognition of whole stories.

Thus, for training, the publicly available multi-view IXMAS dataset is chosen (Weinland et al., 2006). It is comprised of 13 actions, performed by 12 different actors. Each activity instance was recorded simultaneously by 5 different cameras.

Since no suitable standard videos are available in order to describe the complexity of a real life application with a significant number of complex activities, we create a new dataset, called the Waiting Room dataset “WaRo11” (Santofimia et al., 2012), that we make available to the scientific community. In addition, using very different datasets for training and testing allows us to show the generality of our framework, its capabilities for real-world applications and its performance under a challenging situation.

Since the “WaRo11” dataset has been designed for being representative of the variability existing in a real life scenario, but also for integrating most of the actions trained for the CVS, a specific setup was configured to simulate a waiting room. In this setup, actions happen without giving any instructions to the subjects. They are performed as natural part of their behaviour and motivation as human beings. This is facilitated thanks to the presence of several elements interrelated to each other, which may introduce causality and sequentiality as it is found in a real situation. For instance, the presence of a book and a chair could motivate a subject to first pick up the book and then sit down to carry out the action reading. Alternatively, a subject may pick up the book, realises its topic of no interest and decides to throw it away.

This waiting room setup was implemented in a single room and filmed by a single fixed camera. A book was positioned on the floor, a chair was left in a corner and a punching ball was placed in another corner. Eleven sequences were recorded with eleven different actors of both genders comprising a wide range of ages (19-57) and morphological differences. No instruction was given to the actors further than “go to the room and wait for 5 minutes and feel free to enjoy the facilities while you wait”. The resulting variability in the actions performed is depicted in Table 1.

Sequence	Age	Sex	Number of actions
Actor 1	34	M	31
Actor 2	33	M	25
Actor 3	35	M	10
Actor 4	57	F	12
Actor 5	19	M	9
Actor 6	19	M	18
Actor 7	20	F	15
Actor 8	19	M	9
Actor 9	22	F	5
Actor 10	19	M	12
Actor 11	20	F	9
Total			**155**

Actions	Instances
check watch	4
cross arms	0
scratch head	2
sit down	13
get up	12
turn around	18
walk	53
wave hand	9
punch	26
kick	10
point	3
pick up	13
throw	0

Table 1: a) Number of actions performed by each actor. b) Number of instances of the trained actions found in the WaRo11 dataset.

Each of the recorded sequence was manually groundtruthed: first, the video of interest was segmented into a set of independent actions, then each action was labelled. Note that the segmentation of a video into independent actions is outside the scope of this study. Therefore, when testing our algorithms, we processed manually segmented actions. Readers interested in automatic action segmentation should refer to (Rui and Anandan, 2002; Black et al., 1997; Ali and Aggarwal, 2001; Shimosaka, 2007; Shi, 2011).

ii. Results

a) Performance of the computer vision system

First the CVS was applied to IXMAS sequences using the leave-one-out strategy followed by (Weinland et al., 2007; Yan et al., 2008; Junejo et al., 2008; Richard and Kyle, 2009). In each run, we select one actor for testing and all remaining subjects for training. Secondly, using the whole of the IXMAS dataset for training, the CVS was applied to WaRo11. Accuracy performances for both experiments are provided in Table 2.

Table 2. Average recognition rate for all the actions on the datasets obtained by the computer vision system based on BoW

	IXMAS	WaRo11
CVS: BoW	63.9%	29.4%

The BoW based technique displays results comparable to those of the state of the art on the IXMAS dataset (Nebel et al. 2011). However, when applied to a more realistic environment, performances decrease considerably. This shows the limitations of the CVS methodology under real circumstances, when the testing conditions differs significantly from the training. On the other hand, when performance is analysed in terms of average cumulative recognition curve (ACR) - Figure 4, blue -, i.e. percentage that an action is accurately recognised within a set of estimates,- one can see that considering the first few ranks may improve significantly accuracy. For example, accuracy would jump from 29 to 66% if the best solution could be detected within the 6 first estimates. This confirms that additional information is contained within the action estimation vector generated by BoW, and, therefore, there is scope to exploit it to improve the initial annotation. This is exactly what our reasoning system intends to do.

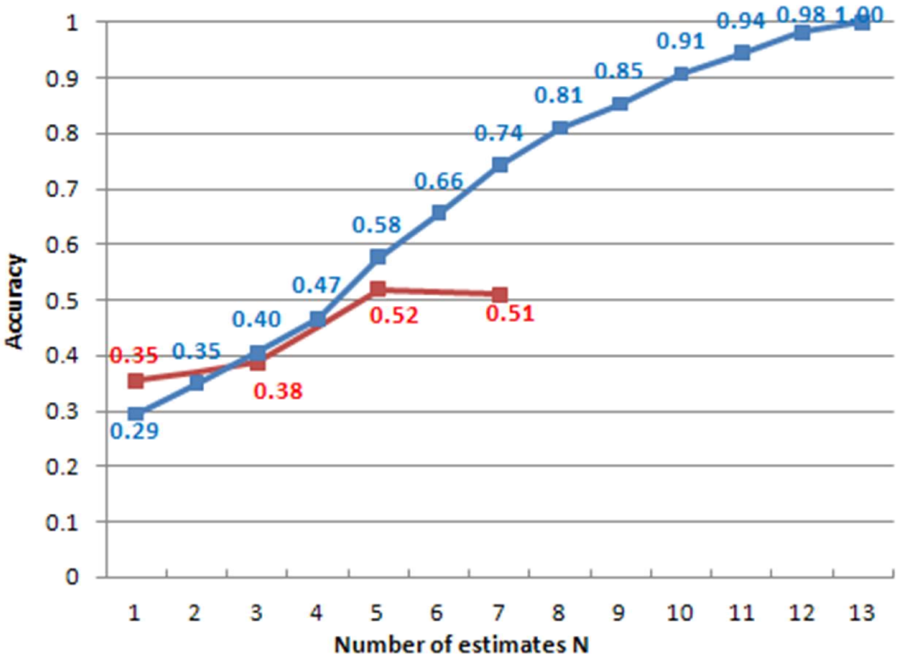


Figure 4: Blue: Average Cumulative Recognition curve for a number of estimations from 1 to 13. Red: Recognition rate obtained by our approach depending on the number of considered action estimates.

b) Performance of the whole framework

The proposed framework integrating AIRS has been tested using the 11 sequences of WaRo11. Experiments were conducted considering the N={1,3,5,7} most likely actions estimates – as calculated by CVS - for AIRS analysis. Performance results are evaluated against the CVS only system in Table 3, where average and recognition rates per sequence are provided. In addition, they are compared with the CVS cumulative recognition rate, Figure 4, red.

Table 3. Recognition rates obtained using either CVS or the combination of CVS and AIRS on WaRO11 dataset.

Actor	1	2	3	4	5	6	7	8	9	10	11	Average per action
CVS	35.5%	16.0%	30.0%	58.3%	44.4%	22.2%	**40.0%**	15.4%	40.0%	16.7%	**33.3%**	29.4%
CVS+AIRS (n=1)	38.7%	24.0%	30.0%	58.3%	44.4%	22.2%	33.3%	**30.8%**	**60.0%**	**25.0%**	**33.3%**	35.5%
CVS+AIRS (n=3)	41.9%	28.0%	40.0%	66.7%	44.4%	38.9%	20.0%	**30.8%**	**60.0%**	**25.0%**	**33.3%**	38.7%
CVS+AIRS (n=5)	**64.5%**	**52.0%**	50.0%	**75.0%**	**55.6%**	**66.7%**	**40.0%**	**30.8%**	**60.0%**	**25.0%**	**33.3%**	**51.9%**
CVS+AIRS (n=7)	61.3%	40.0%	**60.0%**	**75.0%**	**55.6%**	**66.7%**	33.3%	**30.8%**	40.0%	**25.0%**	**33.3%**	51.0%

These results show a considerable increase of performance due to the inclusion of the reasoning system, i.e. accuracy raises from 29% to 52%, in the best case. Our framework outperforms significantly the CVS system, even for the case where only 1 action prediction is considered by the AIRS. Moreover, it can be noticed that accuracy is only rarely deteriorated by reasoning: the system does not seem to suffer from either reasoning errors or over reasoning. Only in sequences 7 and 11 performance are either deteriorated or unaffected by the inclusion of the AIRS. Detailed analysis of these two sequences permits to identify, first, absence of continuity or causality between their composing actions and, secondly, a high percentage of unconstrained actions, i.e. actions that are not linked to any other and that can be performed at any instant (‘cross arms’, ‘check watch’, ‘scratch head’).

These two factors explain why no effective reasoning can be performed to improve recognition.

A more detailed analysis of the AIRS can be obtained by comparing the performance of our approach when varying the number of predictions considered in the action estimate vector. When only considering the most likely action estimate (N=1), the reasoning system is already able to improve on the CVS. This demonstrates the value of one of the AIRS reasoning mechanisms, i.e. activity recognition based on expectations. In this context, the AIRS is comparable to the state-of-art techniques in video-based systems based on simple ontologies and rules.

When several action estimates are available, the AIRS's second mechanism, i.e. common sense action validation and the coherent assignation of actions to stories, can be exploited, which leads to deeper reasoning. Performance of the total system – i.e. 38% and 52% for N=3 and 5 estimates, respectively - compared with those displayed by the ACR – 40% and 57%- shows that the complete reasoning system is quite efficient at selecting an action among the N best estimates (see Figure 4, red). Finally, when more estimates are considered, it seems that the added noise prevents the reasoning system to further improve accuracy, i.e. 51% for N=7.

Figure 5 provides confusion matrices with (CVS+AIRS for the best case, i.e. N=5) and without reasoning (CVS only) to visualise improvement on the recognition rate per action. For many actions, such as 'sitting down', 'getting up', 'turn around', 'check watch' or 'kick', the system is able to move from a recognition rate of almost 0% to a situation where the action is recognised correctly in a majority of instances. This is particularly remarkable in the case of 'sitting down' where the CVS was trained using sequences of individuals sitting on the floor, whereas in WaRO11, they sit on a chair.

Such achievement could not have been reached without usage of world and contextual information. As discussed earlier, recognition rate of an unconstrained action such as ‘scratch head’ does not benefit from reasoning.

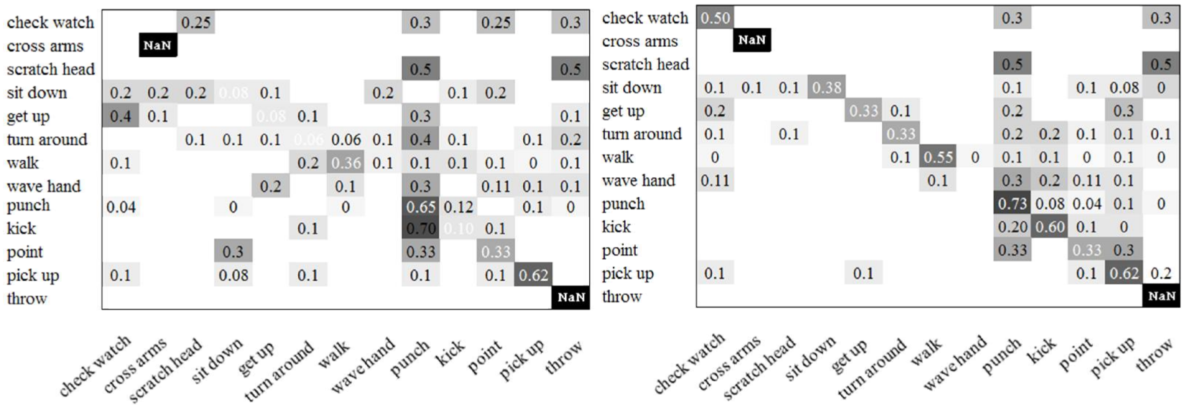
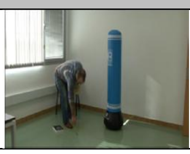
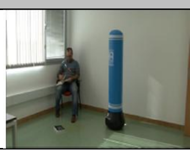
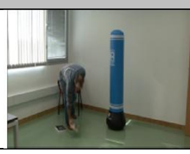


Figure 5. Confusion matrices obtained with CVS (left) and CVS+AIRS (right)

Table 4: Outputs of CVS (N=5) and AIRS for the first 10 actions of WaRo11 seq. 1

					
Frames	220-271	271-310	310-344	344-373	373-394
Ground truth	**Walk**	**Pick up**	**Turn around**	**Sit down**	**Get up**
CVS 1	**Walk**	**Pick up**	**Kick**	**Sit down**	**Check watch**
CVS 2	**Kick**	**Point**	**Point**	**Throw**	**Throw**
CVS 3	**Point**	**Throw**	**Turn around**	**Check watch**	**Kick**
CVS 4	**Wave hand**	**Scratch head**	**Pick up**	**Pick up**	**Point**
CVS 5	**Sit down**	**Sit down**	**Cross arms**	**Cross arms**	**Pick up**
AIRS main story	**Walk**	**Pick up**	**Turn around**	**Sit down**	**Get up**
					
Frames	394-432	432-1243	1243-1276	1276-1326	1326-1533
Ground truth	**Pick up**	**Sit down**	**Get up**	**Pick up**	**Punch**
CVS 1	**Pick up**	**Cross arms**	**Punch**	**Pick up**	**Punch**
CVS 2	**Get up**	**Point**	**Point**	**Throw**	**Kick**

CVS 3	Throw	Check watch	Kick	Get up	Throw
CVS 4	Scratch head	Scratch head	Pick up	Point	Point
CVS 5	Turn around	Sit down	Throw	Check watch	Check watch
AIRS main story	Turn around	Sit down	Get up	Pick up	Punch

Table 4 illustrates the importance of reasoning to improve performance by showing outputs of CVS (N=5) and AIRS for the first 10 actions of sequence 1. When CVS failed to identify the correct actions as its first estimate, AIRS was able to choose the correct annotations among the other 4 estimates, i.e. ‘turn around’ and ‘sit down’ actions. Moreover, when none of the CVS outputs was suitable, AIRS managed to correct those estimates by inferring a new action consistent with common sense reasoning – ‘get up’ actions. An error of reasoning occurred in the 6th action, where the AIRS contradicted the correct CVS estimation. This error is explained by the unexpected presence of a second object on the floor, i.e. a pen, which was not known by the DSK. Consequently, the rule imposing that a second object could be picked only after releasing the first one proved invalid.

6. Conclusions

We present a novel approach for action recognition based on the combination of statistical and knowledge based reasoning. The inclusion of artificial intelligence strategies, based on common sense, allows outperforming significantly the state of the art technique in computer vision when dealing with realistic datasets. Our main contributions are the creation of the first integrated framework combining computer-vision-based and artificial-intelligence-based action recognition techniques which is fully context and scenario independent, and the implementation of a common sense reasoning schema which outperforms machine learning methodologies.

Results are highly encouraging and confirm the validity of our hypothesis: the computer vision community should not focus exclusively on classical statistical reasoning, but should integrate ideas and methodologies from artificial intelligence in order to overcome the limitations of current applications under real-life conditions.

Acknowledgement

This research has been partly supported by the Spanish Ministry of Economy and Competitiveness under the project DREAMS TEC2011-28666-C04-03.

References

Ahmad, M. and Lee, S.-W., 2008. Human action recognition using shape and clg-motion flow from multi-view image sequences. Pattern Recognition, 41(7): pp. 2237–2252.

Akdemir, U., Turaga, P., Chellappa, R., 2008. An ontology based approach for activity recognition from video. Proceeding of the 16th ACM international conference on Multimedia, pp.709-712.

Ali, A., Aggarwal, J. K., 2001. Segmentation and Recognition of Continuous Human Activity. IEEE Workshop on Detection and Recognition of Events in Video.

Black, M., Yacoob, Y., Jepson, A., Fleet, D., 1997. Learning parameterized models of image motion. IEEE Conf. on Comput. Vis. and Patt. Recog.

Blackburn, J., Ribeiro, E., 2007. Human motion recognition using isomap and dynamic time warping. Lecture Notes in Computer Science, 4814: pp. 285–298.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R., 2005. Actions as space-time shapes. ICCV.

Chen, D., Yang, J., Wactlar, H.D., 2004. Towards automatic analysis of social interaction patterns in a nursing home environment from video.Proc. 6th ACM SIGMM Int. Workshop Multimedia Inf. Retrieval, pp. 283–290.

Chen, W., Fahlman, S.E., 2008. Modeling Mental Contexts and Their Interactions", AAAI 2008 Fall Symposium on Biologically Inspired Cognitive Architectures.

Chen, L. Nugent, C.D., 2009. Ontology-based activity recognition in intelligent pervasive environments. IJWIS 5(4), pp. 410-430.

Csurka, G., Bray, C., Dance, C., Fan, L., 2004. Visual categorization with bags of keypoints. Workshop on Statistical Learning in Computer Vision, pp. 1–22.

Dean, M., Schreiber, G., (ed) van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, L., 2011a. OWL Web Ontology Language Reference http://www.w3.org/TR/2003/WD-owl-ref-20030331/ (last accessed March 2011).

Dean, M., Schreiber, G. (eds), Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L. Patel-Schneider, P.F., Stein, L.A., Olin, F.W., 2011b.

OWL Web Ontology Language http://www.w3.org/TR/owl-ref/ (last accessed March 2011).

Duong, T.V, Bui, H.H., Phung, D.Q, Venkatesh, S., 2005. Activity recognition and abnormality detection with the switching hidden semi-markov model. CVPR, pp. 838-845.

Eagle, N. Singh, P., Pentland, A., 2003. Common sense conversations: understanding casual conversation using a common sense database. Proceedings of the Artificial Intelligence, Information Access, and Mobile Computing Workshop.

Fahlman, S.E., 2006. Marker-Passing Inference in the Scone Knowledge-Base System. First International Conference on Knowledge Science, Engineering and Management (KSEM'06).

Fang, C., Chen, J., Tseng, C., Lien, J., 2009. Human action recognition using spatio-temporal classification. Proceedings of the 9th Asian Conference on Computer Vision, pp. 98–109.

Francois, A.R.J., Nevatia, R., Hobbs, J., Bolles, R.C., 2005. VERL: An Ontology Framework for Representing and Annotating Video Events. IEEE MultiMedia 12(4): pp.76-86.

Fellbaum, C., 1998. WordNet: An Electronic Lexical Database. MIT Press

Georis, B., Maziere, M., Bremond, F., Thonnat, M., 2004. A video interpretation platform applied to bank agency monitoring. Proc. 2nd Workshop Intell. Distributed Surveillance System, pp.46–50.

Hakeem, A., Shah, M., 2004. Ontology and Taxonomy Collaborated Framework for Meeting Classification. Proc. Int. Conf. Pattern Recognition, pp.219–222.

Hobbs, J., Nevatia, R., Bolles, B., 2004. An Ontology for Video Event Representation. IEEE Workshop on Event Detection and Recognition.

Ivano, Y., Bobick, A., 2000. Recognition of Visual Activities and Interactions by Stochastic Parsing. IEEE Trans Pattern Analysis and Machine Intelligence .22(8): pp.852–872.

Jia, K., Yeung, D., 2008. Human action recognition using local spatio-temporal discriminant embedding. International Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. ECML.

Junejo, I.N., Dexter, E., Laptev, I., Pérez, P., 2008. Cross-view action recognition from temporal self-similarities. ECCV 2008, Part II. LNCS, vol. 5303, pp. 293–306.

Kaaniche, M.B., Bremond, F., 2010. Gesture Recognition by Learning Local Motion Signatures. CVPR.

Kellokumpu, V., Zhao, G., Pietikäinen, M., 2008. Human activity recognition using a dynamic texture based method. Proceedings of the 19th British Machine Vision Conference, pp. 885–894.

Kuipers, B., 1994. Qualitative Reasoning: Modelling and Simulation with Incomplete Knowledge. Cambridge, Mass.: MIT Press.

Kovashka, A., Grauman, K., 2010. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 2046–2053.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T. 2011. HMDB: A Large Video Database for Human Motion Recognition. ICCV.

Laptev, I., 2005. On Space-Time Interest Points. International Journal of Computer Vision. 64(2/3): pp. 107–123.

Laptev, I., Perez, P., 2007. Retrieving Actions in Movies. ICCV.

Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B., 2008. Learning realistic human actions from movies. Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Lenat, D., Guha, R.V, 1989. Building Large Knowledge-Based Systems: Representation and Inference in the Cyc project. Addison-Wesley Longman Publishing Co., Inc.

Lenat, D., Guha, R.V., Pittman, K., Pratt, D., Shepherd, M., 1990. Cyc: Toward programs with common sense. Commun, ACM, 33(8): pp.30-49.

Lewandowski, J., Makris, D., Nebel, J.C., 2010. View and style-independent action manifolds for human activity recognition. Proc. ECCV 6316.

Lewandowski, J., Makris, D., Nebel, J.C., 2011. Probabilistic Feature Extraction from Time Series using Spatio-Temporal Constraints. Pacific-Asia Conference on Knowledge Discovery and Data Mining.

Liu, J., Ali, S., Shah, M., 2008. Recognizing human actions using multiple features. Proceedings of the International Conference on Computer Vision and Pattern Recognition.

Liu, J., Shah, M., 2008b. Learning human actions via information maximization. Proceedings of the International Conference on Computer Vision and Pattern Recognition.

McCarthy, J., 1968. Programs with Common Sense. Semantic Information Processing, Vol. 1, pp. 403–418.

McCarthy, J., 1979. Ascribing mental qualities to machines. Philosophical Perspectives in Artificial Intelligence, pp. 167-195.

Makris, D., Ellis, T., Black, J., 2008 Intelligent Visual Surveillance: Towards Cognitive Vision Systems. The Open Cybernetics and Systemics Journal, 2, pp. 219-229.

Martinez F., Orrite, C., Herrero, E., Ragheb, H., Velastin, S., 2009. Recognizing human actions using silhouette-based HMM. Proceedings of the 6th International Conference on Advanced Video and Signal Based Surveillance, pp 43–48.

Matikainen, P., Hebert, M., Sukthankar, R., 2010. Representing pairwise spatial and temporal relations for action recognition. Proceedings of the 11th European Conference on Computer Vision.

Minsky M., 1986. The society of mind. Simon & Schuster, Inc.

Moore, D.J., Essa, I.A., Hayes, M.H., 1999. Exploiting human actions and object context for recognition tasks. ICCV, pp 80-86.

Natarajan, P., Nevatia, R., 2008. View and scale invariant action recognition using multiview shape-flow models. Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Nebel, J.C., Lewandowski , M., Thevenon, J., Martinez, F., Velastin, S., 2011. Are Current Monocular Computer Vision Systems for Human Action Recognition Suitable for Visual Surveillance Applications? International Symposium on Visual Computing.

Orrite, C., Martinez, F., Herrero, E., Ragheb, H., Velastin, S.A., 2008. Independent viewpoint silhouette-based human action modeling and recognition. MLVMA.

Philipose, M., Fishkin, K.P., Perkowitz, M., Patterson, D.J., Kautz, H., Hahnel, D., 2004. Inferring activities from interactions with objects. IEEE Pervasive Computing Magazine, 3(4): pp. 50-57.

Richard, S., Kyle, P., 2009. Viewpoint manifolds for action recognition. EURASIP Journal on Image and Video Processing.

Rui, Y. Anandan, P., 2002. Segmenting visual actions based on spatiotemporal motion patterns. CVPR.

Ryoo, M.S., Aggarwal, J.K., 2009. Spatio-Temporal Relationship Match: Video Structure Comparison for Recognition of Complex Human Activities. ICCV.

Schuldt, C., Laptev, I., Caputo., B., 2004. Recognizing human actions: A local SVM approach. ICPR.

Shi, Q., Wang, L. Cheng, L., Smola, A., 2011. Discriminative Human Action Segmentation and Recognition using Semi-Markov Model, International Journal of Computer Vision, 93(1): pp. 22-32.

Shimosaka, M., Mori, T., Sato, T., 2007. Robust Action Recognition and Segmentation with Multi-Task Conditional Random Fields. IEEE International Conference on Robotics and Automation, pp. 3780 - 3786.

Ta, A., Wolf, C., Lavoué, G., Baskurt, A., Jolion, J.-M., 2010 Pairwise features for human action recognition. Proceedings of the 20th International Conference on Pattern Recognition.

Tapia, E.M, Intille, S., Larson, K., 2004. Activity recognition in the home using simple and ubiquitous sensors. Pervasive, pp. 158-175.

Turaga, P., Veeraraghavan, A., Chellappa, R., 2008. Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. International Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Vezzani, R., Baltieri, D., and Cucchiara, R., 2010. HMM based action recognition with projection histogram features. Proceedings of the 20th International Conference on Pattern Recognition: Contest on Semantic Description of Human Activities.

Vu, V.T., Bremond F., Thonnat, M. 2002. Temporal Constraints for Video Interpretation. 15th European Conference on Artificial Intelligence.

Waltisberg, D., Yao, A., Gall, J., Van Gool, L., 2010. Variations of a hough-voting action recognition system. ICPR 2010. LNCS, vol. 6388, pp. 306–312.

Wang, L., Suter, D., 2007. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Wang, L., Suter, D., 2007b. Learning and matching of dynamic shape manifolds for human action recognition. IEEE Transactions on Image Processing, 16(6): pp. 1646–1661.

Wang, S., Pentney, W., Popescu, A.M., Choudhury, T., Philipose, M., 2007c. Common Sense Based Joint Training of Human Activity Recognizers. Proc. International Joint Conference on Artificial Intelligence.

Wang, L. and Suter, D., 2008. Visual learning and recognition of sequential data manifolds with applications to human movement analysis. Computer Vision and Image Understanding, 110(2): pp. 153–172.

Santofimia, M.J., Martinez-del-Rincon, J., Nebel, J.C., 2012. WaRo11 Dataset (under development)

Weinland, D., Boyer, E., and Ronfard, R., 2007. Action recognition from arbitrary views using 3d exemplars. Proceedings of the 11th International Conference on Computer Vision, 5(7):8.

Weinland, D., Ronfard, R., Boyer, E., 2006. Free viewpoint action recognition using motion history volumes. Computer Vision and Image Understanding 104(2-3), pp. 249–257.

Weinland, D., Özuysal, M., Fua, P., 2010. Making Action Recognition Robust to Occlusions and Viewpoint Changes. ECCV.

Yan, P., Khan, S., Shah, M., 2008. Learning 4D action feature models for arbitrary view action recognition. CVPR.

Zhang, J., Gong, S., 2010. Action categorization with modified hidden conditional random field. Pattern Recognition, 43(1): pp. 197–203.