Chapter 1

Related Work

The general problem to study in this project is activity recognition with a mobile robot. In this chapter, relevant related work is reviewed.

In humans, activity recognition is a cognitive skill that can be considered mainly into perception. The basis to understand it relies first in Psychology, because it provides the concepts and the evidence of how the mind is constituted (1.1). The next step is to look at the possibilities to mimic a cognitive process into a machine, this problem has been studied widely in Artificial Intelligence. In particular, activity recognition has been studied in Computer Vision. Finally, the problem has to landed to a robotic stage, making emphasis on the advantages and disadvantages of a robotic platform.

1.1 General antecedents - Perception in AI

Perception, as a cognitive process, has been studied widely in Psychology. It refers to the process of organizing and interpreting sensory information so that it has a meaning [King, 2014]. Part of the interest is about how sensory information is processed by the brain, and which parts of it are essential. Also relevant is the domain knowledge that the subject has about a particular context. Together, the sensory input and the domain knowledge are used to interpret a scene.

Sensory input is important for perception, however, not all the data is equally important to interpret a particular scene and conclusions can still be made, even with partial data. In [Heider and Simmel, 1944], an animated film was created using only moving polygons to demonstrate how the motion of abstract entities could be interpreted by human observers in meaningful ways. In [Johansson, 1973], locomotion patterns of living organisms using visual marks were studied. By this mean, the emphasis was put in the qualitative motion description of the marks rather than in the qualitative motion description of the moving body.

In Artificial Intelligence, perception has been treated mostly by the computer vision research community. Earlier works can be traced back to the 1960s, as part of the effort to mimic human-like intelligence using visual perception components. The main difference between computer vision and image processing has been the desire to recover the three-dimensional structure of the world from images, and to use this as a stepping stone towards full scene understanding [Winston and Horn, 1975].

One of the earlier works in 3D reconstruction from a single image is found in [Roberts, 1963]. The developed system was able to reconstruct geometrical bodies with flat surfaces by recognizing the borders of the bodies in the scene and later analysing the shades of their visible surfaces. In [Barrow and Popplestone, 1971] object recognition was studied by decomposing an image into regions and describing the spatial relations between them, in a more qualitative, rather than the traditional quantitative, approach.

Since the early 1970s, the *block's world* was used as a test scenario for intelligent systems, particularly regarding knowledge representation, reasoning and planning. In the block's world, an initial state A and a desired state B of the environment are given. The goal is to autonomously generate a plan to transform A into B by the manipulation of the blocks. One important characteristic of the problem is that requires a symbolic representation of the scene. The problem was used as a test case for the robot Shakey [Nilsson, 1984].

1.2 Activity Recognition

Activity recognition is an important research area in the context of automated perception. It has many applications as surveillance, inspection, verification, generation of automated reports, etc. The application will dictate the approach to follow and the kind on sensors that will be required.

First, regarding sensing, two approaches can be followed, environmental and/or pervasive. The first one observes the scene from the distance as it happens with a CCTV camera or a robot. The pervasive approach relies on wearable devices to detect the activity of a person from a first person point of view.

Another possible classification of activity recognition systems focuses on how information is processed. In [Aggarwal and Ryoo, 2011] a taxonomy is proposed as shown in Fig. 1.1.

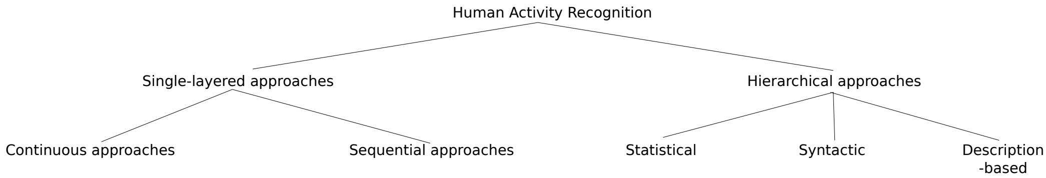

Figure 1.1: The taxonomy of research in activity recognition described in Aggarwal and Ryoo [2011].

1.2.1 Single-layered approaches

They represent activities in terms of raw sensory data[1], because of this, the activity descriptions are trained from datasets.

Single-layered approaches are suitable to recognize short-term and simple activities as gestures, movements of the body or simple interactions with objects. This is mainly because the amount of sensory data grows very easily and long-term activities would require to process larger amounts of data. Also, because activities are not always performed in the same way, even by the same individual; the shorter the activity, the more accuracy that will be attained. An finally, because they are dependant on the sensors and on the environmental conditions (e.g. lighting, point of view).

Continuous approaches[2]

The activities are recognized by analysing continuous sensory data and compare it with an activity pattern.

An activity is represented as a block of data along time where the activity was performed, and it is considered as a whole. A volume (or hyper-volume) is built by concatenating the sensor readings in time. The dimension of the data will depend on the sensing capabilities of the system; for example, a video stream would require 3 dimensions (X, Y, T) and a RGBD camera would be able to use 4 dimensions (X, Y, Z, T), etc. The sensory input is compared with the activity patterns to measure similarity. If a threshold is fulfilled, then the activity is labelled.

The advantages of this approach is that it is relatively fast and doesn't require domain knowledge. However, it is very dependant of the sensory

[1]The original survey [Aggarwal and Ryoo, 2011] describes single layered approaches as image-based approaches, but it leaves out the systems with other sensing capabilities (e.g. 3D sensors, sonars, GPS, etc.). However, they can be included too the activities are represented in terms of raw sensory data patterns.

[2]'Space-time approaches' in [Aggarwal and Ryoo, 2011]

input, the continuity of the data, how the activity is performed and of the point of view where the scene is being observed.

There are many examples of this approach. In [Bobick and Davis, 2001] a video stream of aerobics exercises was analysed by attaching to every pixel a vector indicating the presence and recency of motion. Then the stream was compared online with previously described activities to look for matching. In [Ke et al., 2007], volumes were built by attaching similar regions of adjacent frames. Then, the problem was transformed in an object matching problem by comparing the shapes of the volumes (sensory stream and activity patterns).

Sequential approaches

Sequential approaches represent activities as a sequence of states. A state is a vector of features observed in the scene in a specific time. Finally, the sequence is analysed depending on the activity representation. There are two approaches: exemplar-based and model-based.

In exemplar-based approaches, activities, or a class of them, are represented as a sequence of states. Then the sensory input is compared in similarity with the patterns. An example can be found in [Darrell and Pentland, 1993], where states are built from view models. Templates of activities are from sequences of states associated with a physical change (e.g. rotation and scale). The dynamics of articulated objects in scene were recognized using the dynamic time warping algorithm (DTW) to the sequence of states.

In model based approaches, the sequence of states is compared with a set of probabilistic models of activities. The models are built assuming a temporal dependence between the states, so the transitions are modelled probabilistically using hidden Markov models (HMM) or dynamic bayesian networks (DBN).

The first work to use HMM to recognize activities was [Yamato et al., 1992]. They transformed a video stream into a sequence of vectors of image features. Then every vector was transformed to a symbol using vector quantization. Finally, a set of HMMs were created to model the activities, and their parameters were optimized.

1.2.2 Hierarchical approaches

Hierarchical approaches for activity recognition refer to those where complex activities are represented in terms of simpler ones. Multiple layers are defined to represent activities in different levels of complexity. Low level activities can be recognized using single-layered approaches.

Hierarchical approaches are also adequate to represent activities symbolically by using the multi-layered organization to describe semantic relations. By these means, hierarchical approaches are less dependant to training data and they can integrate domain knowledge more easily.

Hierarchical approaches can be categorized, regarding the applied methodology for recognition as statistical, syntactical and description-based.

Statistical

They are based in the hierarchical construction of statistical state-based models, such as HMMs or DBNs.

First the set of activities to work with is defined and organized hierarchically. Complex activities are defined in terms of simpler ones and so on until everything can be synthesized to atomic actions. In this way, many layers are created, from atomic actions to complex activities. In the bottom level, atomic actions are recognized from sensory data using single-layered sequential approaches. As a result, a sequence of feature vectors is transformed into a sequence of atomic actions. This sequence is the input for the next layer, which now will be treated as a new sequence of observations, and the same approach to recognize atomic actions from the first layer will be applied in the second one, and so on.

In [Oliver et al., 2002], the authors present layered hidden Markov models (LHMMs) for online activity recognition using data from video, sound and keyboard data. They divide their system in three layers: the first one is in charge of recognizing features from every source, the second layer trigger short events from the scene, and the last layer is used for longer activities. The hierarchical approach showed an improved performance when compared to single-layered systems. The training data is used more efficiently and it's more easy to add more detail on specific activities.

Some disadvantages of the statistical approaches is their difficulty to model the temporal structure of events (e.g. A occurred 'during'/'before'/'after' B) and also, because of their sequential nature, is hard to handle multiple concurrent tasks.

Syntactic

In the syntactical approach, activities are represented symbolically as a set of production rules generating a string of atomic actions which is later recognized using parsing techniques. Atomic actions are obtained with a single-layered approach, however, in higher layers, recognition is performed symbolically. Context-free grammars (CFGs) and stochastic context-free grammars

(SCFGs) are some of the techniques that have been used to recognize high level activities.

One limitation of this approach is the difficulty to handle concurrent activities, and also to consider unexpected events that are not integrated in the grammar.

An example can be found in [Ivanov and Bobick, 2000]. The authors aim to recognize complex activities in sequences of video. Two layers are defined; in the lower level, atomic actions are recognized using HMMs, and in the upper one uses SCFGs. The approach showed to be able to handle longer time activity constraints and more robust to uncertain detections in the lower level.

Description-based

This approach represent activities as a hierarchy of events, making emphasis in their spatial, temporal and logical structures.

A complex activity is modelled from de occurrence of its sub-events that satisfies certain relations. The temporal relation between sub-events is also considered in the representation, Allen's calculus is frequently used for this [Allen, 1983]. Atomic actions are obtained from sensory data and summarized.

Now to recognize activities the problem becomes a *constraint satisfaction problem*, which is NP-hard. This approach allows a good integration of additional knowledge sources. Particularly, the

There are many possibilities to treat the problem. In [Nevatia et al., 2004, Ryoo and Aggarwal, 2006], CFGs are used to represent activities hierarchically, defining temporal relations between sub-events. In [Sridhar et al., 2010], relevant features from the scene are extracted and their behaviour is represented using qualitative spatio-temporal relations (QSTR), then patterns of activities are learnt using Markov chains.

1.3 Description-based activity recognition and mobile robotics

In this project, the selected approach to follow is a description-based one. This section presents relevant related work in this line, and in the Answer Set Programming (ASP) paradigm for Also, here are presented the precedents of activity recognition with mobile robots.

1.3.1 Description-based activity recognition

As mentioned in section 1.2.2, description-based approaches represent activities hierarchically by decomposing complex activities in sub-events. The representation should also make emphasis in the spatial, temporal and logical structures. The recognition is performed by obtaining features from scene (spatial, temporal, logical) and creating a scene description as a *list* of facts, then the problem becomes a constraint satisfaction problem, to find the best activity match for these set of observations.

Representation

An activity is represented as a set of *facts* that needs to be fulfilled with the observations. This is important, because the facts can be used as logical predicates. These facts act as constraints between the activity patterns and help to discriminate between them, some of them may be more relevant than the others.

The execution of an activity depends on the subject, and even a particular subject doesn't execute the same activity in the same way. This is the reason why activities are usually defined in qualitative terms, i.e. in a symbolic more human-like manner. Quantitative descriptions of activities are still interesting, however, they are restricted to specific domains as rehabilitation or sports.

Regarding the temporal dimension, time can be represented as an instant t or as an interval (t_1, t_2). For the instants, simple temporal logic can be used to represent these kind of statements. Intervals have been typically treated with two aproaches [Fisher, 2008]: Interval temporal logics [Moszkowski, 1983] and Allen's interval algebra [Allen, 1983].

Allen's interval algebra was introduced as a calculus for temporal reasoning. It defines 13 possible relations between intervals, and provides a composition table that can be used for reasoning about temporal descriptions of events.

Qualitative spatial representations are the focus of study of Qualitative Spatial Representations (QSR). These are a set of calculi which allow a machine to represent and reason about spatial entities [Cohn and Renz, 2007], e.g. lines, dots, regions, etc. They are usually combined with a temporal representation (e.g. Allen's interval algebra) to represent behaviour dynamics.

In [Sridhar, 2010], activities are learnt, in an unsupervised fashion, and recognized from video sequences by reasoning under qualitative spatio-temporal representations (QSTR). Objects positions and their trajectories are extracted from scenes and represented in QSTR. Activities are learnt using a Markov Chain Monte Carlo (MCMC) procedure to find the maximum a posteriori probability (MAP) of candidate interpretations. This work shows an example of unsupervised learning of activities, using simulated and real examples. The qualitative approach (QSTR) is robust to changes in the execution of actions and to sensory errors. Finally, the categorization of activities showed to be reliable in learning functional object categories which provides semantic information from the scene. On the other hand, some limitations of this work are a fixed point of view and a posterior analysis. The analysis is performed only in short video sequences, and in short activities. The search space grows very easily as the scene becomes more complex.

In [Young and Hawes, 2013, 2014], QSTR are applied to the analysis of multi-agent behaviour in the RoboCup simulation league, particularly in estimating future behaviours. Positions, trajectories and orientation of the agents in scene are represented using region connection calculus (RCC), qualitative trajectory calculus (QTC) and the Star calculus respectively. Other agent's behaviours are learnt by using a HMM, which is fed with the current observations and a window of previous ones. As not all the data is relevant, this is first filtered. This work presents a study of activity prediction. The results show that qualitative representations are more easy to treat general cases of activities and require less training data. Some drawbacks are that the system posses global information from the environment, and the domain is restricted.

Reasoning

1.3.2 Mobile robotics

A robot is an ideal system to perform activity recognition. This is, indeed, a desirable skill for an autonomous robot that will share an environment with people. The robot needs to be *aware* of the surrounding humans. As described in section 1.1, even though AI, robotics and computer vision are relatively recent research fields, the interest in perception and particularly in automatic human analysis have been there since the very beginning.

Activity recognition with a mobile robot offers many potential advantages compared with other systems, however, there are some limitations and challenges to be considered too. The range of action for a robotic system in this context will depend on many factors as sensing and processing capacities, knowledge accessibility, etc. Positively, the mobility of a robot can potentially help to improve its perception capacity by participating in the scene, taking data from different points of view and interacting with the environment, this has been stated as *active perception.* [Bajcsy, 1988].

On the other hand, some challenges arise by using a mobile robot:

sensing Sensing data is usually corrupted because of hardware limitations, presence of statistical noise, discretization by the digitalization process, moving sensors, unpredictable environment conditions, etc. Sensing is also restricted by the location of robot (there is no omni-presence) and as a consequence the robot will only gather information from the visible parts of the environment, loosing the rest of it.

storage and processing Sensory data can grow very easily and its storage and processing becomes a challenge. Knowledge bases may grow very easily too, the massive amount of possible instances, relations and categories forces to restrict the scope of knowledge bases to specific domains. The algorithms' complexity is usually high for the required techniques (e.g. pattern recognition, logic programming, etc.).

time Time constraints are relevant in robotic systems as, for many interesting applications, the data cannot be post-processed and real-time response is required.

In this section some relevant systems that implement activity recognition are presented.

Activities

The problem

Objects

1.4 Answer Set Programming

Answer set programming (ASP) is form of declarative programming oriented towards difficult, primary NP-hard, search problems. As an outgrowth of research on the use of non-monotonic reasoning in knowledge representation, it is particularly useful in knowledge intensive applications [Lifschitz, 2008].

It has its roots in deductive databases, logic programming (with negation), logic-based knowledge representation and (non-monotonic) reasoning, and constraint solving (satisfiability testing).

The basic idea in ASP is to express a problem in a logical format so that the models of its representation provide solutions to the original problem. The resulting models are referred as *answer sets* [Gebser, 2013].

A rule is expressed in ASP as:

$$L_0 \; or \; \ldots \; or \, L_k \longleftarrow \; L_{k+1}, \ldots, L_m, \; not \, L_{m+1}, \ldots, \; not \, L_n,$$

each L_i is a literal in the sense of classical logic. The above rule means that if $L_{k+1}, \ldots, L_m$ are true and if $L_{m+1}, \ldots, L_n$ can be assumed to be false, then at least one $L_0, \ldots, L_k$ must be true [Gelfond and Lifschitz, 1988]. The symbol *not* is called *negation as failure*.

Monotonicity refers to the property of a logic programming system that, when more rules are added, it won't produce a reduction in the set of conclusions of the system. Non-monotonicity allows to a conclusion reduction when more rules are added [Poole, 2010]. This concept is important in systems with incoming knowledge, in dynamic and non deterministic scenarios. Also, allows the assumption of truth states, or belief states and a posterior revision of them when more rules are known. It is clear, that this is a desired property, in a logic system, to handle uncertain and incomplete information.

Negation as a failure symbol *not* L_i it is often read as "it is not believed that L_i is true". However, this does not imply that L_i is believed to be false, *not* L_i is a statement about belief [Gelfond, 2014].

1.4.1 ASP as a declarative problem solving technique

In declarative programming, in stead of coding the method to solve a problem, the idea is to describe the problem and leave the computer to find the solution.

1.4.2 ASP as a knowledge representation language

ASP is well suited for modelling problems in the area of Knowledge Representation and Reasoning involving incomplete, inconsistent and changing information [Schaub, 2013]. Some of its properties, in this context, are [Maximova-Todorova, 2003]:

Restricted monotonicity ASP can behave monotonically which addition of literals about certain predicates.

Language independence The results of a program are not dependant on the ASP solver.

Sort-ignorable The sorts can be ignored through language tolerance.

Knowledge extension Knowledge can be extended by *filtering*, i.e. updating the belief state [Amir and Russell, 2003].

ASP has been particularly applied to reasoning satisfiability problems. However, other problems can be treated too by ASP, as: model enumeration, intersection or unioning, as well as multi-criteria and -objective optimization. Formally, ASP allows for solving all search problems in NP and NP^{NP} in a uniform way.

ASP implementations

ASP implementations work n two steps:

1. A [grounder] builds an intermittent representation of the problem files by generating all possible values of the variables.

2. A **solver** that reads the grounded file and generates the answer sets (solutions).

Since the inception of the concept in the 1980s [Gelfond and Lifschitz, 1988] many implementations of ASP have been created.

The majority of them uses the syntax of the language *Lparse*, also called *AnsProlog**. However, *DVL*, one of the most prominent ASP solvers uses a different syntax.

Here is a list of some of the more relevant toolkits available:

potassco Created and maintained by the University of Potsdam, potasscobeen developed and [Gebser et al., 2011].

1.4.3 ASP and robotics

Regarding

1.4.4 ASP Solvers

1.5 Semantic Mapping

Bibliography

J. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, Nov. 1983. ISSN 0001-0782. doi: 10.1145/182.358434. URL `http://doi.acm.org/10.1145/182.358434`.

E. Amir and S. J. Russell. Logical filtering. In G. Gottlob and T. Walsh, editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 75–82, San Francisco, 2003. Morgan Kaufmann.

R. Bajcsy. Active perception. *IEEE Journal on Computer Vision*, 76(8): 996–1006, Aug. 1988.

H. Barrow and R. Popplestone. Relational descriptions in picture processing. In *Machine Intelligence 6*, page 377, 1971.

A. Bobick and J. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, Mar 2001. ISSN 0162-8828. doi: 10.1109/34.910878.

A. G. Cohn and J. Renz. Qualitative Spatial Representation and Reasoning. In F. van Harmelen, V. Lifschitz, and B. Porter, editors, *Handbook of Knowledge Representation*, pages 551–596. Elsevier, Oxford, 2007.

T. Darrell and A. Pentland. Space-time gestures. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, pages 335–340, Jun 1993. doi: 10.1109/CVPR.1993.341109.

M. Fisher. Temporal representation and reasoning. In F. van Harmelen, V. Lifschitz, and B. Porter, editors, *Handbook of Knowledge Representation*, pages 513–550. Elsevier, Amsterdam, 2008.

M. Gebser. *Answer set solving in practice*. Morgan & Claypool Publishers, San Rafael, 2013. ISBN 1608459713.

M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and M. Schneider. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107–124, 2011.

M. Gelfond. *Knowledge representation, reasoning, and the design of intelligent agents : the answer-set programming approach*. Cambridge University Press, New York, NY, 2014. ISBN 1107029562.

M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *5th Conference on Logic Programming*, pages 1070–1080. Seattle, 1988.

F. Heider and M. Simmel. An experimental study of apparent behavior. *The American Journal of Psychology*, pages 243–259, 1944.

Y. Ivanov and A. Bobick. Recognition of visual activities and interactions by stochastic parsing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):852–872, Aug 2000. ISSN 0162-8828. doi: 10.1109/34.868686.

G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, 1973. ISSN 0031-5117. doi: 10.3758/BF03212378. URL `http://dx.doi.org/10.3758/BF03212378`.

Y. Ke, R. Sukthankar, and M. Hebert. Spatio-temporal shape and flow correlation for action recognition. In *CVPR*. IEEE Computer Society, 2007. URL `http://dblp.uni-trier.de/db/conf/cvpr/cvpr2007.htmlKeSH07`.

L. King. *The science of psychology : an appreciative view*. McGraw-Hill Education, New York, NY, 2014. ISBN 0078035406.

V. Lifschitz. What is answer set programming? In D. Fox and C. P. Gomes, editors, *AAAI*, pages 1594–1597. AAAI Press, 2008. ISBN 978-1-57735-368-3. URL `http://www.aaai.org/Library/AAAI/2008/aaai08-270.php`.

Y. Maximova-Todorova. Representing commonsense knowledge using Answer Set Programming. Master's thesis, Universidad de las Américas Puebla, Puebla, Mexico, 2003.

B. C. Moszkowski. *Reasoning About Digital Circuits*. PhD thesis, Stanford, CA, USA, 1983. AAI8329756.

R. Nevatia, J. Hobbs, and B. Bolles. An ontology for video event representation. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, pages 119–119, June 2004. doi: 10.1109/CVPR.2004.27.

N. Nilsson. Shakey the robot. Tech Note 323, AI Center, SRI International, 1984.

N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 3–8, 2002. doi: 10.1109/ICMI.2002.1166960.

D. Poole. *Artificial intelligence foundations of computational agents*. Cambridge University Press, New York, 2010. ISBN 0521519004.

L. G. Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, Massachusetts Institute of Technology, June 1963.

M. Ryoo and J. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1709–1718, 2006. doi: 10.1109/CVPR.2006.242.

T. Schaub. Answer set programming: Boolean constraint solving for knowledge representation and reasoning. In C. Schulte, editor, *CP*, volume 8124 of *Lecture Notes in Computer Science*, pages 3–4. Springer, 2013. ISBN 978-3-642-40626-3. URL `http://dx.doi.org/10.1007/978-3-642-40627-0`.

M. Sridhar. *Unsupervised Learning of Event Classes from Video*. PhD thesis, University of Leeds, 2010.

M. Sridhar, A. G. Cohn, and D. C. Hogg. Unsupervised learning of event classes from video. In *Proc. AAAI*, pages 1631–1638. AAAI Press, 2010.

P. H. Winston and B. Horn. *The psychology of computer vision*. McGraw-Hill computer science series. McGraw-Hill, New York, 1975. ISBN 0-07-071048-1. URL `http://opac.inria.fr/record=b1083572`. Includes index.

J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 379–385, Jun 1992. doi: 10.1109/CVPR.1992.223161.

J. Young and N. Hawes. Predicting situated behaviour using sequences of abstract spatial relations. In *Proceedings of the AAAI 2013 Fall Symposium How Should Intelligence be Abstracted in AI Research: MDPs, Symbolic Representations, Artificial Neural Networks, or _____?*, 2013.

J. Young and N. Hawes. Effects of training data variation and temporal representation in a qsr-based action prediction system. In *AAAI Spring Symposium 2014 on Qualitative Representations for Robots*, Stanford University in Palo Alto, California, US, March 2014.