Towards Learning of Safety Knowledge from Human Demonstrations

Philipp Ertle, Michel Tokic, Richard Cubek, Holger Voos, Dirk Söffker

***Abstract*— Future autonomous service robots are intended to operate in open and complex environments. This in turn implies complications ensuring safe operation. The tenor of few available investigations is the need for dynamically assessing operational risks. Furthermore, a new kind of hazards being implicated by the robot's capability to manipulate the environment occurs: hazardous environmental object interactions.**

One of the open questions in safety research is integrating safety knowledge into robotic systems, enabling these systems behaving safety-conscious in hazardous situations. In this paper a safety procedure is described, in which learning of safety knowledge from human demonstration is considered. Within the procedure, a task is demonstrated to the robot, which observes object-to-object relations and labels situational data as commanded by the human. Based on this data, several supervised learning techniques are evaluated used for finally extracting safety knowledge. Results indicate that Decision Trees allow interesting opportunities.

I. Introduction

To *"[...] overcome the practically impossible problem of preidentifying the full range of kinds of situations robots and other agents will get into during normal interaction with their environments, [...] we should [...] seek to build robots, and artificial agents in general, that are autonomous"* [1]. Of course, the author suggested this, having the complexity and NOT the safety problem in mind. Unfortunately, this statement also holds for safety considerations, and thus, autonomy of technical systems raises new challenges for the system safety process.

In [2] it is reported that *Next Generation Robots* will be highly complex autonomous systems, probably being *"[...] capable of performing such tasks as house cleaning, security, nursing, life-support, and entertainment - all functions to be performed in co-existence with humans in businesses and homes"* [2]. In order to perform such complex tasks, these robots require having abstract skills combining behaviors and tasks. Besides the problem of pre-define or pre-program required behaviors as initially mentioned, the realization of such behaviors for complex everyday environments is non-trivial. Therefore, *Learning from Demonstration* (LfD) has become a major topic in the robotics community in order to provide the required learning and generalization capabilities [3]. For instance, a *pick and place* task is realized via LfD in [4] with the goal to generalize a task over the always differing positions an object needs to be picked-up and placed afterwards.

P. Ertle and D. Söffker, University of Duisburg-Essen, Chair of Dynamics and Control (SRS), 47057 Duisburg, Germany

R. Cubek and M. Tokic, University of Applied Sciences Ravensburg-Weingarten, *ZAFH-Servicerobotik*, 88250 Weingarten, Germany

M. Tokic, University of Ulm, Institute of Neural Information Processing, 89069 Ulm, Germany.

H. Voos, University of Luxembourg, Faculté des Sciences, de la Technologie et de la Communication, L-1359 Luxembourg

This work was also conducted within the collaborative center for applied research *ZAFH-Servicerobotik*. The authors gratefully acknowledge the research grants of the federal state of Baden-Württemberg and the European Union.

A. Problem Statement

When robots are 'trained' with the help of LfD methods, predominant questions are: What about safety? Are safety aspects already comprised in the demonstration?

With regard to pick and place tasks, this can be negated: Surely, new learned skills might be limited by already known safety constraints, but they might comprise new hazards not considered so far. As outlined in former research work [5], various new hazards appear when robots are enabled to manipulate their environment. Gripped objects might hazardously interact with other environmental objects. In this case, a robot learns to manipulate objects (pick and place), it would be very questionable that knowledge about safely handling various objects is adequately considered. Hence, the general problem is to sufficiently integrate safety issues into the learning process as well.

It is assumed that allowing end-consumers to 'teach' their robots has far reaching consequences, not considered in this contribution. The proposed procedural model represents an approach to simplify safety engineering aspects by LfD. The learned safety knowledge is currently intended to be checked and revised afterwards.

II. Related Work

Safety is recently considered most often with regard to physical human robot interaction (pHRI). Most of the related contributions are addressing collision safety [6], [7], [8].

A. Safety and Learning

Some investigations are available for realizing safe learning algorithms, for instance, by defining stability bounds for controllers [9]. But recently, domain knowledge is assumed to be available prior learning in order to limit the system to safe states, for instance in [9], [10]. In [11], *Apprenticeship Learning* is applied for generating a default policy, which is on the one hand, the basis for further learning, but the fall-back option for adverse circumstances as well.

The learning of safety constraints is described in [12]. A domain ontology is developed where domain concepts define a set of individuals in the world, belonging together, and sharing some properties. Their so-called Constraint Learner procedure checks for demonstrations of specific properties `p` from which new lower or upper bounds can be observed: `if upper_bound(p)<value_observed(p) then upper_bound(p)=value_observed(p)` and the like for lower bounds.

An approach for learning spatial constraints is given in [13]. A decision tree is learned from labeled data. In a theoretic example it is learned how a rectangle is located at a specific (at the right-hand side of the observer) spatial position. Learning of safety constraints is not explicitly mentioned, but as shown in this contribution at hand, the basic idea can easily be extended.

B. Safety Engineering Methods

The general system safety process is integrated in the development process and takes place throughout the complete life cycle of a system [14]. The general steps take place in a cyclic manner: hazard identification, hazard risk assessment, risk control, and risk verification. The risk control is sufficient since all risks are mitigated to an acceptable level [14]. With regard to autonomous systems, it is suggested (in the field of systems of systems) to structure the hazard analysis with regard the capabilities a system provides: *"Each capability will present a variety of possible hazards, stemming from a failure to provide the capability, an incorrect implementation of the capability, or from unexpected side-effects of employing the capability"* [15]. This is in line with the standard ISO/NP 13482 for robots in (non-medical) personal care as reported in [16], where a list of tasks need to be identified, specified with regard to functional and non-functional requirements.

III. Behavior-based Safety Procedure

The majority of robotic architectures is realized with three layers. At the lowest level, skills are orchestrated by the superordinate layers (for instance, see [17]). These skills are combined to more complex behaviors. The behaviors do not need representational knowledge [18], and thus, they can basically be regarded as structurally invariants during the operation phase. Moreover, this is also valid for cognitive-oriented robotic architectures, for instance, described in [19].

The proposed approach is to modify the system safety process [14] so that risks are reduced by learned countermeasures (safety functions). In the first instance, safety functions basically label perceived situations as either risky or not. How far the generated labels are utilized for 'limiting' the autonomy of the system is non-trivial and therefore, not detailed in this contribution (implications of a simple binary view on safety and the mere prohibiting of unsafe decision alternatives are discussed in [20]).

The modified safety process is shown in Fig. 1. At first, hazards being related to a specific behavior are required to be identified and assessed. This is according to the conventional procedure in order to determine the hazards, required to be mitigated. To control the risk, namely the generation of countermeasures according to identified and unacceptable risks, four steps are identified: definition of the requirements for the demonstration, demonstration itself, revising of learned safety functions, and their integration.

The outcome of the demonstration task finally is either a learned or a learned and refined safety function. The safety function labels the perceived (measured) situation, which can

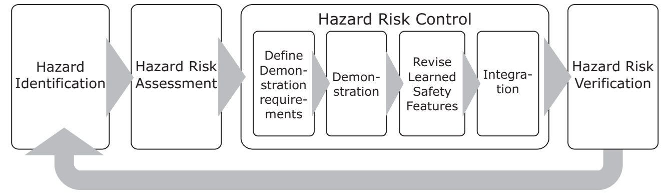

Fig. 1. System safety process for learning safety features from demonstration.

be in turn considered by a subsequent decision or planning process (integration).

A. Demonstration Requirements

The definition of the requirements plays a central role. First, required data must be determined. If relevant data is not provided to the learning approach, the learning problem can not be classified. If relevant aspects can not be measured or derived from prior system knowledge, the demonstration approach is consequently not reasonable. Secondly, demonstrated data has to be labeled. An adequate user interface must be provided. At third, the identified hazards have to be demonstrated without provoking accidents. For the most cases, it is assumed that it is possible to demonstrate an approaching towards a hazard, in terms of the hazard is not endangered to be actuated. The limitations learned in consequence, are still in a safe range, and therefore, they include a specific safety clearance. The indicating of a hazard plays a central role. Therefore, the demonstration sequence must be precisely defined. In order to make the safety learning problem classifiable, sufficient desired and undesired constellations must be demonstrated. Consequently, aspects that are not comprised in the demonstration can not be learned. The hazards must be 'encircled' by the indications of the undesired approaching so that it becomes clear which area (hyper space) is undesired. The problem is illustrated in Fig. 2.

B. Demonstration

During the demonstration it must be ensured that the demonstration specifications are maintained. Further on, it is important to be as precise as possible when indicating undesired/risky situations. Otherwise, too many ambiguities may let the learning problem become unclassifiable. Several repetitions of a demonstration might be necessary.

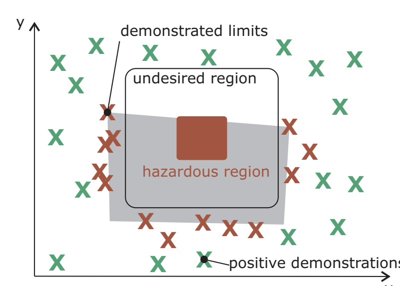

Fig. 2. Demonstrations must be carefully planned in order to ensure that undesired regions are sufficiently represented.

C. Revise Safety Features

Since there are many factors that contribute to good or bad learning results, an intensive revising of the learning success is assumed to be mandatory, at least for severe hazards. In this case, the LfD approach can be seen as a design aid for generating safety functions. The learned safety functions have to be additionally checked. Further on, they have to be available in a readable form what is either given (learning of decision trees) or what could be realized by additional transformation steps (for instance, extraction of rules from neural networks [21] or from reinforcement learning [22]). Nevertheless, an LfD approach remains helpful and may reduce faults. As shown in the following experiments, the considering of spatial relations between objects in 3D-space requires coordinate transformations which often are not intuitive.

For hazard risks that are basically acceptable but undesired, the LfD approach might be applied in a fully autonomous manner or it may be additionally active during operating time.

D. Integration

The outcome of the learned safety function depends on the applied learning algorithm. In general, the outcome might consist of binary, any continuous or probabilistic prediction values. In case the severity S of an accident A is numerically expressed, a risk value R can be computed in terms of,

$$R_A = P(A) \cdot S(A) \ ,$$

where $P(A)$ is the probability of the accident A. The expression of hazards in risk measures is important [20] and can be considered in a subsequent decision process [5].

Finally, the verification and the assurance of the successful mitigation of identified hazard risks has to take place. For integrating the LfD approach into the safety engineering process, traditional verification techniques such as testing can be applied. When the LfD approach is applied during operating time, automated verification might be required. This implicates further problems that are not discussed in this contribution.

IV. Experimental Setup

In order to test the described approach, two different scenarios were chosen. It is assumed that for each scenario a specific robotic behavior would be required. Each scenario comprises simple hazards to be learned. The goal is to learn a safety function for each behavior, which can be applied for detecting unsafe/undesired situations.

The approach is applied considering real world conditions with noisy sensor data. Furthermore, those risks are focused which appear when environmental and robot manipulated objects interact with each other. These objects are represented by markers, whose position and orientation are detected by the *ARToolKit*[1]. The camera observing the scene was mounted at the demonstrator's head in order to simulate the consistently different positions a robot will be relatively located in real world application.
The data is labeled as risky if a key is pressed (e.g. representing an emergency button or voice command module).

The demonstration videos are first captured and stored. Afterwards, the videos are presented to a C++ program[2] utilizing the ARToolKit library in order to store the recognized position and pose data vectors. Finally, the stored data is analyzed, see Sec. V.

A. The Ironing Task

The first scenario is the ironing scenario. In this task, the hazard of fire is potentially comprised when an *iron* remains too long at the same position. Therefore, it is demonstrated that the standstill of the *iron* on the *ironing board* is not desired. The placing of the *iron* at its backside (in upright position) or in the provided deposit is uncritical. The scene is shown in Fig. 8.

The relevant data for the safety function to be learned is measured. These consist of the relative positions x, y, z and angles α, β, γ from the coordinate system of *object* 0 (*ironing board*) and *object* 1 (*iron*), respectively. Additionally, the absolute values of the relative velocities v are computed. Observed data is transformed into vector form with a predefined order. Thus, the situation input vector s of a typical ironing scene consists of the following 14 continuous inputs, as

$$s = [x_0^1, y_0^1, z_0^1, v_0^1, \alpha_0^1, \beta_0^1, \gamma_0^1, x_1^0, y_1^0, z_1^0, v_1^0, \alpha_1^0, \beta_1^0, \gamma_1^0] \ .$$

Data points are labeled as commanded by the user. Initially, *normal* operation is assumed, which is the default label for all data vectors. In case hazardous situations are demonstrated, data is labeled to *risk* as long as the emergency button is pressed.

B. The Stack-It-Safely Task

Inspired by the *Safe-To-Stack* approach in [23], a scenario is designed for learning the safe stacking of two objects. The hazard that was identified for this scenario is the toggling and falling down of the narrow *object* 1 that is stacked upwards on *object* 0. The flat stacking is assumed to be uncritical. Surely, the modeling of spatial relations as *on, upwards* and so forth may simplify the problem description, but finally, these relations also relay on similar relative coordinates as they were utilized in the demonstration. A flag whether an object is gripped or released is not considered in the scenario, the relative velocities of the objects becoming zero indicate a similar state as the release of a gripped object. The vector remains the same as described in Sec. IV-A. The scene is shown in Fig. 9.

[1] `http://www.hitl.washington.edu/artoolkit/` [online, accessed 2012-03-12]

[2] all data, videos, source code and KNIME workflows are available under `http://www.robot-safety.com/LearningSafety/`

	Parameter	Ironing	Stacking
RBFN	clustering seed	1	1
	maxIts	unl.	unl.
	minStdDev	0.1	0.1
	numClusters	20	1
	ridge	1.00E-08	1.00E-08
MLP	data	normalized	normalized
	Max.Num.Iterations	1000	1000
	Hidden layer	2	2
	Neurons per hid. layer	10	10
J48 DT	conf. Factor	1	1
	minNumObj	100	62
	numFold	3	3
	seed	1	1
	subtreeRiasing	true	true
	pruning	true	true
	use laplace	false	false

Fig. 3. Parameters used for learning algorithms.

V. Experiment Evaluation

The data-mining tool KNIME[3] in combination with Weka[4] extensions are utilized for extracting functional representations of the demonstrated safety knowledge. Three supervised learning techniques are evaluated: *Radial Basis Function Network* (RBFN), J48 *Decision Tree* (DT), and *Multilayer Perceptron* (MLP). Each of this classifiers is manually hand-tuned, according to Fig. 3, for obtaining reasonable results.

The performance of these classifiers is measured with the recall and F1-measure (positive means indication of risk, true/false positive/negatives being tp, fp, tn, fn),

$$\begin{aligned} \text{Recall} &= tp/(tp+fn) \ , \\ \text{F1} &= 2 \cdot tp/(tp+fn+tp+fp) \ . \end{aligned}$$

The recall reflects the sensitivity with regard to correct classification of risks. The F1-measure represents the overall accuracy.

VI. Results

In the sequel the results of the described tasks are given. In the subsequent section, the results are discussed.

A. The Ironing Task

In total three demonstrations have been presented with a total of 3633 training-data vectors, where the duration of each demonstration varies between 114 to 142 seconds. These comprise 2909 vectors (80%) belonging to the *normal* class, and 724 vectors (20%) to the *risk* class. A fourth demonstration is used as test data consisting of 1100 vectors, 900 vectors (81%) belonging to the *normal* class, and 200 vectors (19%) to the *risk* class.

The diagrams in Fig. 4 show the classifications of the test data according to the trained classifiers. The classification recall and F1 performance of the overall task is shown in Fig. 7 (upper). The parameters are computed for classification of the test data after learning of 1, 2, and 3 demonstrations. In the last demonstration, 4*, the test data is additionally learned. The performance of the complete demonstration scenario is computed with regard to the same test data.

[3] `http://www.knime.org` [online, accessed 2012-03-12]

[4] `http://www.cs.waikato.ac.nz/ml/weka/` [online, accessed 2012-03-12]

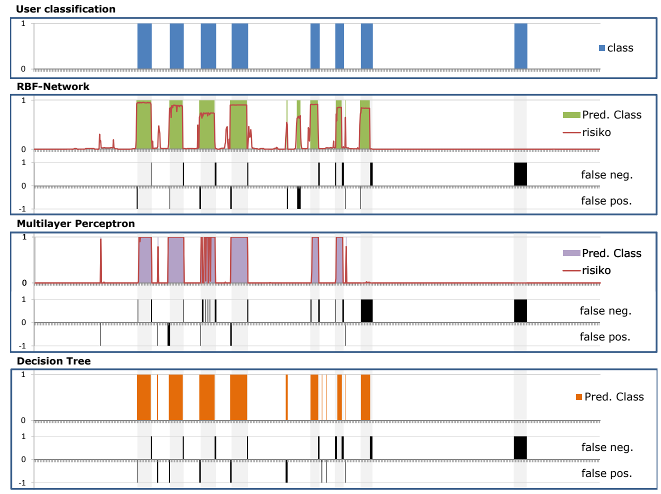

Fig. 4. Time plot section illustrating the classification of the experiment test data for the ironing task. The section shows all classification failures of the experiment. The first diagram indicates when the user has demonstrated risks (1 corresponds to risk/positive). The following double diagrams each illustrate the results of the respective learning method. The first of the double diagrams shows when risks are predicted, the second illustrates when fn (1), correct prediction (0), and fp (-1) occur. The curve in the RBFN and MLP diagrams represent the predicted probability for risk, respectively.

$v_0^1 \leq$ 2.4362
| $x_0^1 \leq$ -93.2441
| | $\gamma_1^0 \leq$ 21.063
| | | $z_1^0 \leq$ -39.851
| | | | $\alpha_1^0 \leq$ 54.86 : normal (128.0 / 46.0)
| | | | $\alpha_1^0 >$ 54.86 : risk (182.0 / 41.0)
| | | $z_1^0 >$ -39.851 : risk (453.0 / 28.0)
| | $\gamma_1^0 >$ 21.063 : normal (211.0)
| $x_0^1 >$ -93.2441 : normal (397.0)
$v_0^1 >$ 2.4362 : normal (2262.0 / 112.0)

Fig. 5. Learned decision tree for the ironing task (v in [cm/s]; x, y, z in [mm]; α, β, γ in [°]).

The decision tree generated, reflects well comprehensible the effect of the respective measurements: Sufficient velocity is desired, via relative x-position the deposit of the *ironing board* is detected, via the γ-angle it is detected whether the *iron* is turned upright, and if the *iron* is above the *ironing board* there is no risk as well. The angle α remains as a fragment. It is noteable that the relative x-position is considered from the perspective of the *ironing board* (object 0) and γ-angle from perspective of the *iron* (object 1). Otherwise neither the position at the iron storage nor the depositing of the *iron* at its backside could be easily distinguished.

B. The Stack-It-Safely Task

In total three demonstrations have been made as well, with a total of 3184 training-data vectors, where the duration of each demonstration varies between 114 to 145 seconds. These comprise 2452 vectors (77%) belonging to the *normal* class, and 732 vectors (23%) to the *risk* class. A fourth demonstration is used as test data consisting of 772 vectors, 666 vectors (86%) are belonging to the *normal* class, and 106 vectors (14%) to the *risk* class. The diagrams in Fig. 6 show the classifications of the test data according to the trained classifiers. The classification recall and F1 performance of

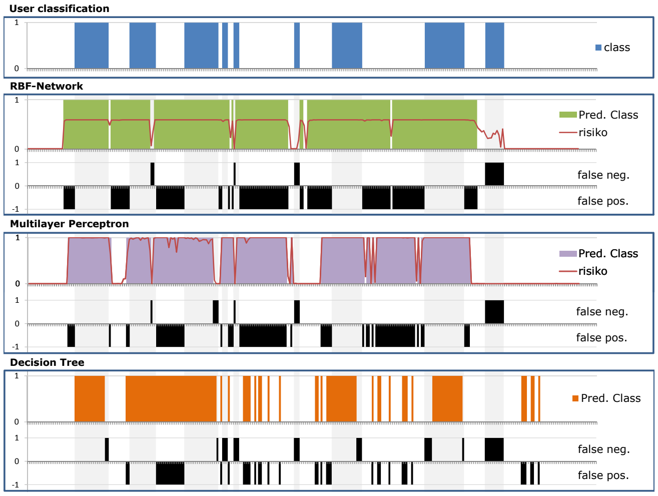

Fig. 6. Time plot section illustrating the classification of the experiment test data for the stack-it-safely task. For explanation, see Fig. 4.

#Demos	F1			Recall		
	RBFN	MLP	DT	RBFN	MLP	DT
	Inoning Task					
1	**0.76**	0.29	0.73	0.74	0.80	**0.88**
2	0.72	0.70	**0.75**	0.84	0.78	**0.86**
3	**0.82**	0.77	0.80	**0.88**	0.79	0.85
4*	0.80	0.70	**0.88**	0.74	0.74	**0.90**
	Stack-it-Safely					
1	**0.60**	0.19	0.54	0.47	0.19	**0.64**
2	0.58	0.54	**0.75**	0.45	**0.76**	0.75
3	0.58	**0.73**	0.66	0.44	**0.62**	0.61
4*	0.61	**0.68**	**0.67**	0.45	0.53	**0.63**

Fig. 7. The classification performance in terms of the F1 and recall measure. The performance is measured against the test data after 1, 2, and 3 demonstrations. *The test data are additionally integrated into the learning process in a fourth step, the performance is measured according to the same test data. Best F1/recall measure is highlighted, respectively.

the overall task is shown in Fig. 7 (lower). The parameters are computed for classification of the test data after learning of 1, 2, and 3 demonstrations. In the last demonstration, 4^*, the test data is additionally learned as well.

VII. Discussion

Basically, only a selection of learning algorithms was considered. The performance of further classifiers would be from general interest.

The learning results in the ironing task show an interesting recall and F1 measure, already, after the first demonstration. The decision tree reflects very well the relevant aspects of the problem. The RBFN-algorithm performs not bad at all, but the MLP seems to have potential with further demonstrations, see Fig. 7 (upper). Many of the false predictions remain during the transitions from risk to normal situations and vice versa, see Fig. 4. The DT-algorithm seems to have the best potential. The ironing task seems to be a learning problem that is well classifiable.

The stack-it-safely task appears to comprise many blurred classification boundaries. The data indicates that further demonstrations are possibly required. The MLP seems again to have potential in that respect, see Fig. 7 (lower). The RBFN- and MLP-algorithm strongly overgeneralize the risk situations according to the current demonstration data, see Fig. 6. Here, the DT-algorithm appears to have the best potential, similar as in the ironing task.

In general, the tuning of the learning parameters has a big effect on the results, which is basically nothing new in the machine learning community. The quality and clearness of the demonstration plays also an important role. Especially, false classifications during the transitions might be filtered (false positives) in order to reduce ambiguities during the transitions. In addition to that, the utilization of already learned safety knowledge might increase the clearness of the demonstration because only missed alerts are retrained, and ambiguous redundant alarms might be avoided. Further on, the idea of *Conceptual Spaces* [24] in combination with *Subspace Clustering* is considered as promising approach to be considered in future research.

VIII. Conclusions

The safety problem with regard to autonomous capabilities of future robots is briefly outlined. An approach extending the system safety process for the design of safe robotic behaviors is described. Therefore, the idea of Learning from Demonstration is integrated in the hazard risk control process. In two real-world tasks with noisy sensor data, three learning algorithms were evaluated. The experiments show that the *Decision Tree*-algorithm appears to provide good performance to learn safety functions in object manipulation scenarios besides its advantage to be readable, and at need also adjustable by safety engineers. The results indicate that the proposed safety procedure gains great potential for either supporting the safety engineering process or for improving the safety for post-design learned behaviors as well.

References

[1] T. Smithers, "Autonomy in robots and other agents," *Brain and Cognition*, vol. 34, pp. 88–106, 1997.

[2] Y. Weng, C. Chen, and C. Sun, "Toward the Human–Robot Co-Existence society: On safety intelligence for next generation robots," *International Journal of Social Robotics*, vol. 1, no. 4, pp. 267–282, 2009.

[3] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot Programming by Demonstration," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ch. 59, pp. 1371–1394.

[4] M. Schneider and W. Ertel, "Robot learning by demonstration with local gaussian process regression," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 255–260.

[5] P. Ertle, H. Voos, and D. Söffker, "On risk formalization of on-line risk assessment for safe decision making in robotics," in *7th IARP Workshop on Technical Challenges for Dependable Robots in Human Environments*, Toulouse, France, 2010, pp. 15–22.

[6] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for safe robots: Measurements, analysis and new insights," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, Nov. 2009.

[7] A. Pervez and J. Ryu, "Safe physical human robot interaction-past, present and future," *Journal of Mechanical Science and Technology*, vol. 22, no. 3, pp. 469–483, Mar. 2008.

[8] A. Bicchi, M. A. Peshkin, and J. E. Colgate, "Safety for physical Human-Robot interaction," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin / Heidelberg, 2008, pp. 1335–1348.

[9] T. J. Perkins and A. G. Barto, "Lyapunov design for safe reinforcement learning control," AAAI Press, Menlo Park, CA, Technical Report SS-02-07, 2002.

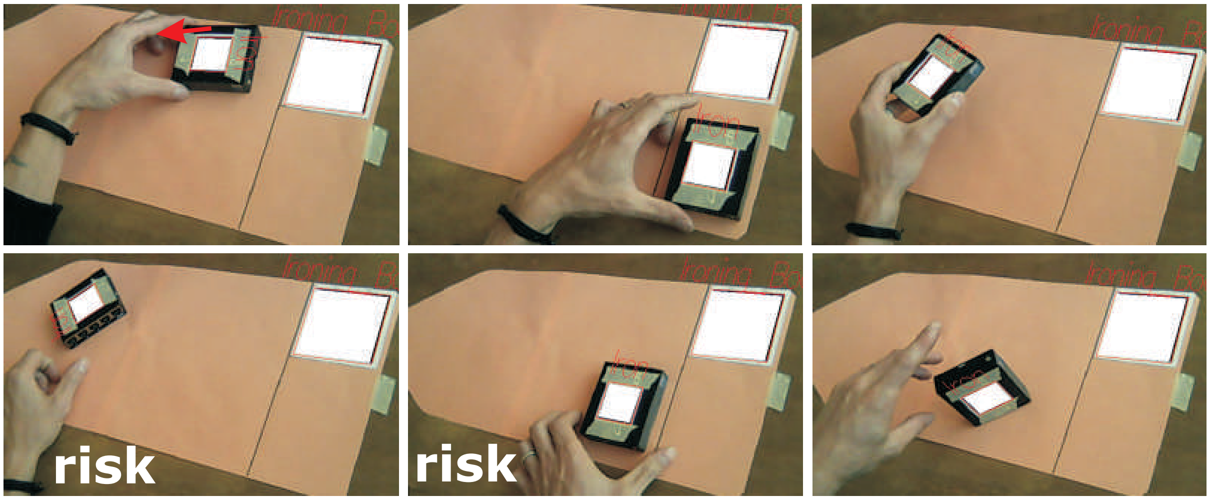

Fig. 8. The ironing task: Pictures extracted from the demonstration video captured from changing positions. The object positions and poses are recognized with ARToolKit. The labeled pictures show *risk* situations (stand-still of the iron). The others show *normal* situations (iron positioned at the iron storage, iron turned at the back side or iron moved over the surface).

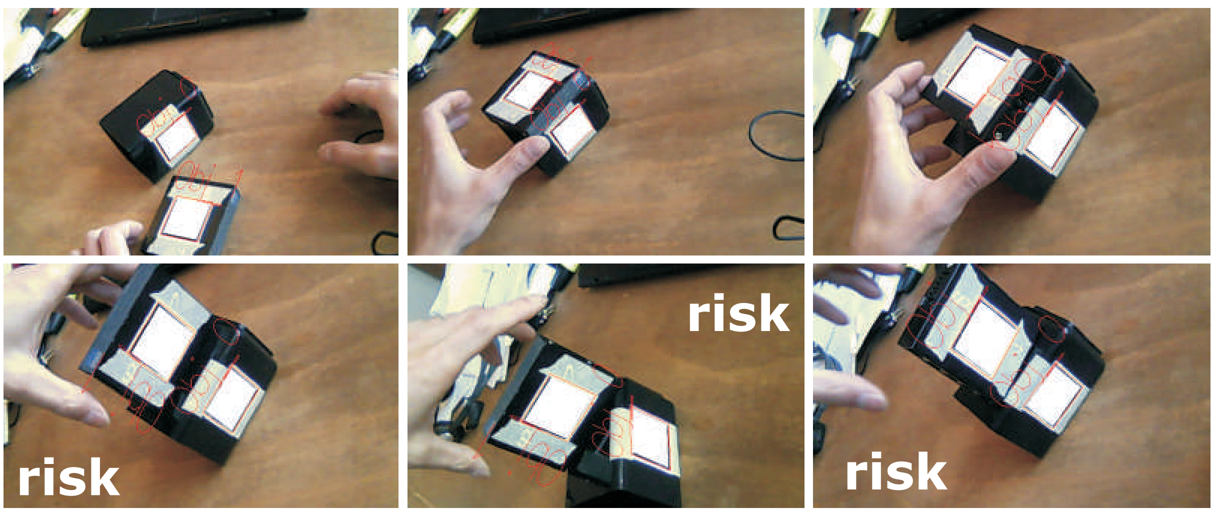

Fig. 9. The stack it safely task: The labeled pictures show *risk* situations (upward stacking of the narrow object). The others show *normal* situations (stacking objects located somewhere else or in correct position at the top of the other object).

[10] P. Geibel, "Reinforcement learning with bounded risk," in *Proceedings of the 18th International Conference on Machine Learning*, ser. ICML'01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 162–169.

[11] A. Hans, D. Schneegaß, A. M. Schäfer, and S. Udluft, "Safe exploration for reinforcement learning," in *Proceedings of the 16th European Symposium on Artificial Neural Networks ESANN 2008*, Bruges, Belgium, 2008, pp. 143–148.

[12] U. Kuter, G. Levine, D. Green, A. Rebguns, D. Spears, and G. Dejong, "Learning constraints via demonstration for safe planning," AAAI Press," Technical Report WS-07, 2007.

[13] C. Gips, S. Wiebrock, and F. Wysotzki, "Learning of constraints and logical implications in the spatial domain," in *Beiträge zum Treffen der GI-Fachgruppe 1.1.3 (Maschinelles Lernen), TR 98/11 in Technischer Bericht des FB Informatik. TU*, 1998.

[14] C. Ericson, *Hazard analysis techniques for system safety*. Hoboken NJ: Wiley-Interscience, 2005.

[15] R. D. Alexander, N. J. Herbert, and T. P. Kelly, "Structuring safety cases for autonomous systems," in *Proceedings of the 3rd IET International Conference on System Safety 2008*, Birmingham, U.K., 2008, pp. 1–6.

[16] C. Harper and G. Virk, "Towards the development of international safety standards for human robot interaction," *International Journal of Social Robotics*, vol. 3, no. 2, pp. 229–234, 2010.

[17] C. Schlegel and R. Worz, "Interfacing different layers of a multilayer architecture for sensorimotor systems using the object-oriented framework SMARTSOFT," in *3rd European Workshop on Advanced Mobile Robots (Eurobot'99).*, Zurich, Switzerland, 1999, pp. 195–202.

[18] R. C. Arkin, *Behavior-based robotics*. Cambridge, Mass.: MIT Press, 1998.

[19] E. Ahle, "Autonomous systems: A cognitive oriented approach applied to mobile robotics," Ph.D. dissertation, University of Duisburg-Essen, 2007.

[20] A. Wardziński, "Safety assurance strategies for autonomous vehicles," in *SAFECOMP'08: Proceedings of the 27th International Conference on Computer Safety, Reliability, and Security*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 277–290.

[21] E. R. Hruschka and N. F. Ebecken, "Extracting rules from multilayer perceptrons in classification problems: A clustering-based approach," *Neurocomputing*, vol. 70, no. 1–3, pp. 384–397, 2006.

[22] D. Vogiatzis and A. Stafylopatis, "Reinforcement learning for rule extraction from a labeled dataset," *Cognitive Systems Research*, vol. 3, no. 2, pp. 237–253, 2002.

[23] T. M. Mitchell, R. M. Keller, and S. T. Kedar-Cabelli, "Explanation-based generalization: A unifying view," *Machine Learning*, vol. 1, no. 1, pp. 47–80, Mar. 1986.

[24] R. Cubek and W. Ertel, "Conceptual similarity as a key to high-level robot programming by demonstration," in *Proc. of the 7th German Conference on Robotics (ROBOTIK 2012), Munich, Germany*, 2012.