Chapter 4
Visuospatial Skill Learning

Seyed Reza Ahmadzadeh and Petar Kormushev

Abstract This chapter introduces Visuospatial Skill Learning (VSL), which is a novel interactive robot learning approach. VSL is based on visual perception that allows a robot to acquire new skills by observing a single demonstration while interacting with a tutor. The focus of VSL is placed on achieving a desired goal configuration of objects relative to another. VSL captures the object's context for each demonstrated action. This context is the basis of the visuospatial representation and encodes implicitly the relative positioning of the object with respect to multiple other objects simultaneously. VSL is capable of learning and generalizing multi-operation skills from a single demonstration, while requiring minimum a priori knowledge about the environment. Different capabilities of VSL such as learning and generalization of object reconfiguration, classification, and turn-taking interaction are illustrated through both simulation and real-world experiments.

4.1 Introduction

During the past decade several robot skill learning approaches based on human demonstrations have been proposed (Ijspeert et al. 2013, 2002; Kormushev et al. 2011; Argall et al. 2009). Many of them address motor skill learning in which new motor skills are transferred to the robot using policy derivation techniques such as mapping function (Vijayakumar and Schaal 2000), system model (Abbeel and Ng 2004), etc. Motor skill learning approaches can be categorized in two main groups: trajectory-based and goal-based. To emulate the demonstrated skill, the former group put the focus on recording and regenerating trajectories (Ijspeert et al. 2002) or

S.R. Ahmadzadeh (✉)
iCub Facility, Istituto Italiano di Tecnologia, via Morego 30, 16163 Genoa, Italy
e-mail: reza.ahmadzadeh@iit.it

P. Kormushev
Dyson School of Design Engineering, Imperial College London,
London SW7 2AZ, UK
e-mail: p.kormushev@imperial.ac.uk

L. Buşoniu and L. Tamás (eds.), *Handling Uncertainty and Networked Structure in Robot Control*, Studies in Systems, Decision and Control 42,
DOI 10.1007/978-3-319-26327-4_4

intermittently forces (Kronander and Billard 2012). In the work by (Bentivegna et al. 2004) for instance, a humanoid robot learns to play air hockey by learning in the work by primitives, or when combined with reinforcement learning in the work by (Kormushev et al. 2010) a robot learns to flip a pancake by observing demonstrations.

In many cases, however, it is not the trajectory that is of central importance, but the goal of the task (e.g. solving a jigsaw puzzle). Learning every single trajectory in such tasks actually increases the complexity of the learning process unnecessarily (Niekum et al. 2012). To address this drawback, several goal-based approaches have been proposed (Verma and Rao 2005; Dantam et al. 2012; Chao et al. 2011). There is a large body of literature on grammars from the linguistic and computer science communities, with a number of applications related to robotics (Niekum et al. 2012; Dantam et al. 2012). Furthermore, a number of symbolic learning approaches exist that focus on goal configuration rather than action execution (Chao et al. 2011). However, in order to ground the symbols, they comprise many steps inherently, namely segmentation, clustering, object recognition, structure recognition, symbol generation, syntactic task modeling, motion grammar, rule generation, etc. Another drawback of such approaches is that they require a significant amount of a priori knowledge to be manually engineered into the system (Niekum et al. 2012; Dantam et al. 2012). In addition, most above-mentioned approaches assume the availability of the information on the internal state of a tutor such as joint angles, while humans usually cannot directly access to imitate the observed behavior.

An alternative to motor skill approaches are visual learning approaches which are based on observing the human demonstration and using human-like visual skills to replicate the task (Kuniyoshi et al. 1994; Lopes and Santos-Victor 2005).

We propose a novel visual skill learning approach for interactive robot learning tasks. Unlike the motor skill learning approaches, our approach utilizes visual perception as the main source of information for learning new skills from demonstration. The proposed approach which is called Visuospatial Skill Learning(VSL), uses *visuospatial skills* to replicate the demonstrated task. *Visuospatial skill* is the capability to visually perceive the spatial relationship between objects. VSL uses a simple algorithm and requires minimum a priori knowledge to learn a sequence of operations from a single demonstration. In contrast to many previous approaches, VSL leverages simplicity, efficiency, and user-friendly human-robot interaction. Rather than relying on complicated models of human actions, labeled human data, or object recognition, our approach allows the robot to learn a variety of complex tasks effortlessly, simply by observing and reproducing the visual relationship among objects. We demonstrate the feasibility of the proposed approach in several simulated and real-world experiments in which the robot learns to organize objects of different shape and color on a tabletop workspace to accomplish a goal configuration. In the conducted experiments the robot acquires and reproduces main capabilities such as, object reconfiguration, absolute and relative positioning, classification, and turn-taking.

The rest of the chapter is organized as follows. Related work is reviewed in Sect. 4.2. The definitions, terminology, and methodology of the VSL approach are

explained in Sect. 4.3. Implementation of the VSL approach is described in Sect. 4.4. Experimental result is reported in Sect. 4.5. And finally conclusions of the research are drawn in Sect. 4.6.

4.2 Related Work

Visual skill learning or learning by watching is one of the most powerful mechanisms of learning in humans. It has been shown that even infants can imitate both facial and manual gestures (Meltzoff and Moore 1977). In cognitive science, learning by watching has been investigated as a source of higher order intelligence and fast acquisition of knowledge (Rizzolatti et al. 1996; Schaal 1999; Park et al. 2008). In the rest of this section we give examples of visual skill learning approaches. We mention works that can be categorized as goal-based and the main source of information is visual perception, and especially those which focus on the learning of object manipulation tasks.

One of the most influential works on the problem of plan extraction from observation was proposed by Ikeuchi and Suehiro (1994). To extract assembly plans from observations a continuous sequence of images is obtained. Using level change in the brightness differences the images are segmented and the objects are detected and recognized using background subtraction and feature match finding respectively. A set of simple features (e.g. edge, face) and a pre-defined geometric model are used in the match finding process. By comparing two sets of object relations, the transition between them are extracted. The system determines the assembly relations by analyzing contact directions. All existing assembly relation transitions are extracted and unnecessary relation transitions are pruned. The system also determines manipulator operations from the assembly relations. The reproduction phase in the conducted experiments are neglected and the authors just focus on the extraction of task plans from observation. One of the drawbacks of their approach is that it requires a geometric model and a predefined coordinate system for each object. The defined coordinate systems are used to determine the grasping configuration and orientation which are predetermined in the system.

The early work of Kuniyoshi et al. (1994) focuses on acquiring reusable high-level task knowledge by watching a demonstration. In their *learning by watching* approach, multiple vision sensors are used to monitor the execution of the task. The focus of the paper is on the demonstration phase and it lacks a reproduction phase. By extracting some basic visual features from the observation, the object recognition system finds a match between the observation and a 3D model of the environment. The system assigns a symbol to each action and the executed actions are recognized from a pre-defined action database. From the set of recognized executed actions a high-level task plan is extracted. To detect the moving object, human-hand is tracked. Since tracking the hand is not sufficient for classifying assembly operations, the meaningful changes are also tracked using temporal subtraction. In addition, the direction to move the search window is detected from the movement of the hand.

The approach can only deal with rectangular objects and translational movements and the system cannot detect rotations. The other disadvantage of the approach is that the model-based shape recognition is computationally expensive and after each operation and before the reproduction phase the model of the environment has to be updated accordingly. This modification increases the complexity of the approach. Finally, the method cannot detect objects in contact as separate objects.

Asada et al. (2000) proposed a method for learning by observation (teaching by showing) based on the tutor's view recovery and adaptive visual servoing. Based on the assumption that coordinate transformation is a time-consuming and error-prone method. Instead, they assume that both the robot and the tutor have the same body structure. They use two sets of stereo cameras, one for observing the robot's motions and the other for observing the tutor's motions. The optic-geometrical constraint, called *epipolar constraint*, is used to reconstruct the view of the agent, on which adaptive visual servoing is applied to imitate the observed motion. In our method, we use coordinate transformation between the sensor and the robot.

Ehrenmann et al. (2001) proposed a learning from observation system using multiple sensors in a kitchen environment with typical household tasks. They focus on pick-and-place operations including techniques for grasping. A data glove, a magnetic field based tracking system and an active trinocular camera head were used in their experiments. Object recognition is done using fast view-based vision approaches. Also, they extract finger joint movements and hand position in 3D space from the data glove. The method is based on pre-trained neural networks to detect hand configurations and to search in a predefined database. However, there was no real-world reproduction with a robot. A similar research focuses on extracting and classifying subtasks for grasping tasks using visual data, generating trajectories and extracting subtasks (Yeasin and Chaudhuri 2000). They use color markers to capture data from the tutor's hand. In our method, we use neither neural networks nor symbol abstraction techniques.

A visual learning by imitation approach was proposed by Lopes and Santos-Victor (2005). The authors utilize neural networks to map visual perception to motor skills (visuo-motor) together with viewpoint transformation. For gesture imitation a Bayesian formulation is adopted. A single camera was used in the experiments.

Ekvall and Kragic (2008) proposed a symbolic learning approach in which a logical model for a STRIPS[1] planner from multiple human demonstrations are learned. In their work, a task planning approach is used in combination with robot learning from demonstration. The robot generates states and identifies constraints of a task incrementally according to the order of the action execution. In this approach a demonstration is assumed to be an image of the target configuration and they do not use observations of each action. Differently from our approach, the objects are first modeled geometrically and a set of SIFT (Scale Invariant Feature Transform algorithm proposed by Lowe (2004)) features for each object is extracted in off-line

[1] STRIPS stands for Stanford Research Institute Problem Solver which is a symbolic planner developed in 1971.

mode and used during the learning phase. The method can only deal with polyhedral objects.

A symbolic goal-based learning approach was proposed by Chao et al. (2011) that can ground discrete concept from continuous perceptual data using unsupervised learning. To learn the task goal, Bayesian inference is used. In the conducted experiments, five non-expert tutors performed multiple demonstrations for five pick-and-place tasks. Each task consists of a single pick-and-place operation. Background subtraction is used for segmentation and the workspace was detected using a marker. The visual set of features consists of width and length of the best fitting ellipsoid, the area of the simplified polygon, and the hue histogram. During each operation, the object that changes most significantly is detected. The starting and ending time of each demonstration is provided using graphical or speech commands. Our approach solely relies on the captured observations to learn the sequence of operations and there is no need to perform any of those steps.

Niekum et al. (2012) proposed a method to learn from continuous demonstrations by segmenting the trajectories and recognizing the skills. Segmentation and recognition are achieved using a Beta-Process Autoregressive Hidden Markov Model (BP-AR-HMM), while Dynamic Movement Primitives are used to reproduce the skill. Their framework can be categorized as a trajectory-based technique that has been used to learn a goal-based task. Joint data, gripper data, and stereo vision data are recorded during the demonstration phase. The method has been applied to rectangular objects which are manipulated without rotation. Object detection has been done just in the simulation and in the real-world experiment AR markers are employed instead. For each experiment, a coordinate frame has been assigned manually to each known object. These coordinate frames are used to relate the objects to the demonstrated skills. One of the drawbacks is that sometimes during the segmentation process an extra skill is extracted and the system fails to identify a coordinate frame for extra skills. Another disadvantage of this framework is that increasing the number of skills in the demonstrations makes the segmentation process more complicated.

Visual analysis of demonstrations and automatic policy extraction for an assembly task was proposed by Dantam et al. (2012). To convert a demonstration of the desired task into a string of connected events, this approach uses a set of different techniques such as image segmentation, clustering, object recognition, object tracking, structure recognition, symbol generation, transformation of symbolic abstraction, and trajectory generation. In our approach, we do not use symbol generation techniques.

Guadarrama et al. (2013) proposed a natural language interface for grounding nouns and spatial relations. The data used for the training phase has been acquired via a virtual world. The PR2 robot equipped with a laser scanner for creating the map and an RGB-D sensor for segmentation and recognition of the objects was employed. Their method learns to classify a database of modeled objects. In contrast to VSL which uses minimum a priori knowledge, there is a huge database of collected images for each object which is used to train a classifier. Also, they apply a language module to learn related spatial prepositions.

Feniello et al. (2014) have built a framework upon our proposed approach, VSL (Ahmadzadeh et al. 2013b), by introducing a stack-based domain specific language

for describing object repositioning tasks. By performing demonstrations on a tablet interface, the time required for teaching is greatly reduced and the reproduction phase can be validated before execution of the task in the real-world. Various types of real-world experiments are conducted including sorting, kitting, and packaging tasks.

4.3 Introduction to Visuospatial Skill Learning

The VSL approach is a goal-based robot learning from demonstration approach. It means that a human tutor should demonstrate a sequence of operations on a set of objects. Each operation consists of a set of actions, for instance a pick action and a place action. In this chapter, we only consider pick-and-place object manipulation tasks in which achieving the goal of the task and retaining the sequence of operations are particularly important. We consider the virtual experimental setup illustrated in Fig. 4.1a which consists of a robot manipulator equipped with a gripper, a tabletop workspace, a set of objects, and a vision sensor. The sensor can be mounted above the workspace to observe the tutor performing the task. Using VSL, the robot learns new skills from the demonstrations by extracting spatial relationships among objects. Afterwards starting from a random initial configuration of the objects, the robot can perform a new sequence of operations which ultimately results in reaching the same goal as the one demonstrated by the tutor. A high-level flow diagram shown in Fig. 4.1b illustrates that VSL consists of two main phases: demonstration and reproduction. In the demonstration phase for each action a set of observations is recorded which is utilized for the match finding process during the reproduction phase. In this section, first, the basic terms for describing VSL are defined. Then the problem statement is described and finally the VSL approach is explained in details.

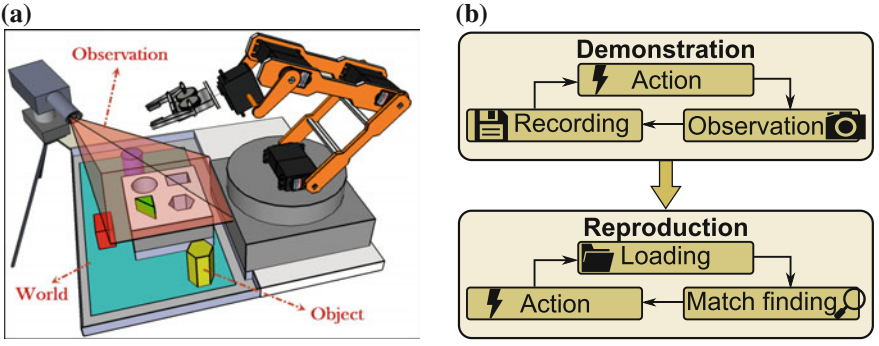

Fig. 4.1 Virtual setup and flow diagram for the VSL approach. **a** Virtual experimental setup for a VSL task consisting of a robot manipulator, a vision sensor, a set of objects, and a workspace. **b** A high-level flow diagram of VSL illustatrating the demonstration and reproduction phases

4.3.1 Terminology

The basic terms that are used to describe VSL consist of:

World: the workspace of the robot which is observable by the vision sensor. The *world* includes objects which are being used during the learning task, and can be reconfigured by the human tutor and the robot.

Frame: a bounding box which defines a cuboid in 3D space or a rectangle in 2D space. The size of the *frame* can be fixed or variable. The maximum size of the *frame* is equal to the size of the *world*.

Observation: the captured context of the *world* from a predefined viewpoint using a specific *frame*. An *observation* can be a 2D image or a cloud of 3D points.

Pre-action observation: an *observation* which is captured just before the action is executed. The robot searches for preconditions in the *pre-action observations* before selecting and executing an action.

Post-action observation: an *observation* which is captured just after the action is executed. The robot perceives the effects of the executed actions in the *post-action observations*.

The set of actions contains different primitive skills for instance, pick, place, push, pull, etc. We assume that actions are known to the robot and the robot can execute each action when required. For an extension to this assumption see Ahmadzadeh et al. (2015).

4.3.2 Problem Statement

Formally, we define a process of visuospatial skill learning as a tuple

$$\mathcal{V} = \{\mathcal{W}, \mathcal{O}, \mathcal{F}, \mathcal{A}, \mathcal{C}, \Pi, \phi\},$$

where $\mathcal{W} \in \mathbb{R}^{m \times n}$ is a matrix which represents the context of the *world* including the workspace and all objects. $\mathcal{W}_D$ and $\mathcal{W}_R$ indicate the *world* during the demonstration and reproduction phases respectively; $\mathcal{O}$ is a set of *observation* dictionaries $\mathcal{O} = \left\{\mathcal{O}^{Pre}, \mathcal{O}^{Post}\right\}$; $\mathcal{O}^{Pre}$ and $\mathcal{O}^{Post}$ are *observation* dictionaries comprising a sequence of *pre-action* and *post-action observations* respectively. $\mathcal{O}^{Pre} = \langle\mathcal{O}^{Pre}(1), \mathcal{O}^{Pre}(2), \ldots, \mathcal{O}^{Pre}(\eta)\rangle$, and $\mathcal{O}^{Post} = \langle\mathcal{O}^{Post}(1), \mathcal{O}^{Post}(2), \ldots, \mathcal{O}^{Post}(\eta)\rangle$. η is the number of operations performed by the tutor during the demonstration phase. Thereby, for example, $\mathcal{O}^{Pre}(i)$ represents the *pre-action observation* captured during the ith operation.

$\mathcal{F} \in \mathbb{R}^{m \times n}$ is an *observation frame* which is used for capturing the *observations*. $\mathcal{A}$ is a set of primitive actions defined in the learning task (e.g. pick). $\mathcal{C}$ is a set of *constraint* dictionaries $\mathcal{C} = \left\{\mathcal{C}^{Pre}, \mathcal{C}^{Post}\right\}$; $\mathcal{C}^{Pre}$ and $\mathcal{C}^{Post}$ are *constraint* dictionaries comprising a sequence of *pre-action*, and *post-action constraints* respectively. Π is

a policy or an ordered action sequence extracted from demonstrations. ϕ is a vector containing extracted features from *observations* (e.g. SIFT features). Pseudo-code of VSL is given in Algorithm 4.7.

Algorithm 4.7 Visuospatial Skill Learning (VSL)

Input: $\{\mathcal{W}, \mathcal{F}, \mathcal{A}\}$
Output: $\{\mathcal{O}, \mathcal{P}, \Pi, \mathcal{C}, \mathcal{B}, \phi\}$
1: $\mathcal{L}$=detectLandmarks($\mathcal{W}$)
2: $i = 1$, $j = 1$
3: // **Part I : Demonstration**
4: **for** each operation **do**
5: $\mathcal{O}_i^{Pre}$ = getPreActionObs($\mathcal{W}_D, \mathcal{F}_D$)
6: $\mathcal{O}_i^{Post}$ = getPostActionObs($\mathcal{W}_D, \mathcal{F}_D$)
7: $[\mathcal{B}_i, \mathcal{P}_i^{Pre}, \mathcal{P}_i^{Post}, \phi_i]$ = getObject($\mathcal{O}_i^{Pre}, \mathcal{O}_i^{Post}$)
8: $[\mathcal{C}_i^{Pre}, \mathcal{C}_i^{Post}]$ = getConstraint($\mathcal{B}_i, \mathcal{P}_i^{Pre}, \mathcal{P}_i^{Post}, \mathcal{L}$)
9: Π_i = getAction($\mathcal{A}, \mathcal{C}_i^{Pre}, \mathcal{C}_i^{Post}$)
10: $i = i + 1$
11: **end for**
12: // **Part II: Reproduction**
13: **for** $j = 1$ to i **do**
14: $\mathcal{P}_j^{*Pre}$ = findBestMatch ($\mathcal{W}_R, \mathcal{O}_j^{Pre}, \phi_j, \mathcal{C}_j^{Pre}, \mathcal{L}$)
15: $\mathcal{P}_j^{*Post}$ = findBestMatch ($\mathcal{W}_R, \mathcal{O}_j^{Post}, \phi_j, \mathcal{C}_j^{Post}, \mathcal{L}$)
16: executeAction($\mathcal{P}_j^{*Pre}, \mathcal{P}_j^{*Post}, \Pi_j$)
17: **end for**

4.3.3 Methodology

At the beginning of the demonstration, the objects are randomly placed in the *world* ($\mathcal{W}_D$). The *world*, the size of the *frame*, and the set of primitive actions $\mathcal{A}$ are known to the robot. In the demonstration phase, the size of the *frame* ($\mathcal{F}_D$) is equal to the size of the *world* ($\mathcal{W}_D$). We consider that the robot is capable of executing the primitive actions from $\mathcal{A}$. For instance, the robot is capable of moving towards a desired given pose and execute a pick action. Implementation details can be found in Sect. 4.4.

In some tasks the tutor utilizes a landmark in order to specify different concepts. For instance, a vertical borderline can be used to divide the workspace into two areas illustrating right and left zones. Another example is a horizontal borderline that can be used to specify far and near concepts. A landmark can be either static or dynamic. A static landmark is fixed with respect to the *world* during both phases. A dynamic landmark can be replaced by the tutor before reproduction phase. Both types are shown in our experiments. In case that, any landmark is being used in the demonstration (e.g. label, borderline), the robot should be able to detect them in the *world* (line 1 in Algorithm 4.7). However, the robot cannot manipulate a landmark.

During the demonstration phase, VSL captures one *pre-action observation* ($\mathcal{O}^{pre}$) and one *post-action observation* ($\mathcal{O}^{post}$) for each operation executed by the tutor using the specified *frame* ($\mathcal{F}_D$) (lines 5, 6). The *pre-action* and *post-action observations* are used to detect the object on which the action is executed. The *observations* are also used to detect the place where the object is repositioned at. For each detected object, a symbolic representation ($\mathcal{B}$) is created. The symbolic object can be used in case that the reproduction phase of the algorithm is replaced with a high-level symbolic planner (Ahmadzadeh et al. 2015). In addition, VSL extracts a feature vector (ϕ) for each detected object. In order to extract ϕ any feature extracting method can be used (e.g. SIFT). The extracted features are used for detecting the objects during the reproduction phase.

VSL also extracts the pose of the object before and after action execution ($\mathcal{P}^{Pre}$, $\mathcal{P}^{Post}$). The pose vectors together with the detected landmarks ($\mathcal{L}$) are used to identify preconditions and effects of the executed action through spatial reasoning (line 8). For instance, if $\mathcal{P}^{Pre}$ is above a horizontal borderline and $\mathcal{P}^{Post}$ is below the line, the precondition of the action is that the object is above the line and the effect of the execution of the action is that the object is below the line. In other words, by observing the predicates of an executed action, the action can be identified from the set of actions $\mathcal{A}$ (line 9). The sequence of identified actions are then stored in a policy vector Π.

Furthermore, the second part of the algorithm is able to execute the learned sequence of actions independently (Ahmadzadeh et al. 2013a, b). In such case, VSL observes the new *world* ($\mathcal{W}_R$) in which the objects are replaced randomly. Comparing the recorded *pre-* and *post-action observations*, VSL detects the best matches for each object and executes the actions from the learned policy. Although, the findBestMatch function can use any metric to find the best matching *observation*, to be consistent, in all of our experiments we use the same metric (see Sect. 4.4.2). After finding the best match, the algorithm extracts the pose of the object before and after action execution, $\mathcal{P}_j^{*Pre}$, $\mathcal{P}_j^{*Post}$ (lines 14, 15). Finally, an action is selected from the policy Π and together with pre and post pose is sent to the executeAction function. This function selects the $\mathcal{A}_j$ primitive action. As we mentioned before, the robot knows how to perform a primitive action, for instance it uses a trajectory generation module and a grasping strategy to perform a pick action. More details about the action execution can be found in Sect. 4.4. It is worth noting that to reproduce the task with more general capabilities, a generic symbolic planner can be utilized instead of the reproduction part of the algorithm (Ahmadzadeh et al. 2015).

4.4 Implementation of VSL

In this section the steps required to implement the VSL approach for real-world experiments are described.

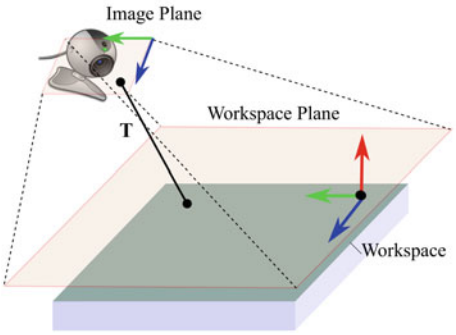

Fig. 4.2 Coordinate transformation from the image plane to the workspace plane (**T** represents the homography matrix)

4.4.1 Coordinate Transformation

In order to transform points between the coordinate frame of the sensor (image plane) and the coordinate frame of the robot (workspace plane) and vice versa a coordinate transformation is required (see Fig. 4.2). The coordinate transformation between the sensor's frame of reference and the robot's frame of reference is done using coordinate transformation matrix **T**. This matrix is calculated using Singular Value Decomposition (SVD) by collecting two sets of points, one set from the workspace and the other set from the captured image (Hartley and Zisserman 2000). Whenever the sensor's frame of reference is changed with respect to the robot's frame of reference, **T** has to be recalculated. Using the updated coordinate transformation matrix, VSL can reproduce the learned skill even if the experimental setup is altered. It means that VSL is a view-invariant approach.

4.4.2 Image Processing

Image processing methods have been employed in both demonstration and reproduction phases of VSL. In the demonstration phase, for each operation the algorithm captures a set of raw images consisting *pre-pick* and *pre-place* images. Firstly, the captured raw images are rectified using the homography matrix **T**. Secondly image-subtraction and thresholding techniques are applied on the couple of images to generate *pre-pick* and *pre-place observations*. The produced *observations* are centered around the *frame*. In the reproduction phase, for each operation the algorithm rectifies the captured *world observation*. Then, the corresponding recorded *observations* are loaded from the demonstration phase and a metric is applied to find the best match (the findBestMatch function in the Algorithm 4.7). Although any metric can be used in this function (e.g. window search method), we use Scale Invariant Feature Transform (SIFT) algorithm (Lowe 2004). SIFT is one of the most popular feature-based methods which is able to detect and describe local features that are invariant to scaling

and rotation. Afterwards, we apply Random Sample Consensus method (RANSAC) in order to estimate the transformation matrix $\mathbf{T}_{\text{sift}}$ from the set of matches. Since the calculated transformation matrix $\mathbf{T}_{\text{sift}}$ has 8 degrees of freedom, with 9 elements in the matrix, to have a unique normalized representation we pre-multiply $\mathbf{T}_{\text{sift}}$ with a normalization constant, α as defined in (4.1). This constant is selected to make the decomposed projective matrix have a vanishing line vector of unit magnitude and that avoids unnatural interpolation results.

$$\alpha = \frac{sign(\mathbf{T}_{\text{sift}}(3,3))}{\sqrt{(\mathbf{T}_{\text{sift}}(3,1)^2 + \mathbf{T}_{\text{sift}}(3,2)^2 + \mathbf{T}_{\text{sift}}(3,3)^2)}}. \tag{4.1}$$

The normalized matrix $\alpha\mathbf{T}_{\text{sift}}$ can be decomposed into simple transformation elements,

$$\alpha\mathbf{T}_{\text{sift}} = \mathbf{T}\mathbf{R}_{\theta}\mathbf{R}_{-\phi}\mathbf{S}_{v}\mathbf{R}_{\phi}\mathbf{P}, \tag{4.2}$$

where $\mathbf{R}_{\pm\phi}$ are rotation matrices to align the axis for horizontal and vertical scaling of $\mathbf{S}_v$; $\mathbf{R}_\theta$ is another rotation matrix to orientate the shape into its final orientation; $\mathbf{T}$ is a translation matrix; and lastly $\mathbf{P}$ is a pure projective matrix that can be written as:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \alpha\mathbf{T}(3,1) & \alpha\mathbf{T}(3,2) & \alpha\mathbf{T}(3,3) \end{bmatrix}.$$

An affine matrix $\mathbf{T}_A$ is the remainder of $\alpha\mathbf{T}$ by extracting $\mathbf{P}$; $\mathbf{T}_A = \alpha\mathbf{T}\mathbf{P}^{-1}$. $\mathbf{T}$ is extracted by taking the 3rd column of $\mathbf{T}_A$ and $\mathbf{A}$, which is a 2×2 matrix, is the remainder of $\mathbf{T}_A$. $\mathbf{A}$ can be further decomposed using SVD such that,

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \tag{4.3}$$

where $\mathbf{D}$ is a diagonal matrix, and $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices. Finally we can calculate the matrices in (4.2) as follows

$$\mathbf{S}_v = \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \mathbf{R}_\theta = \begin{bmatrix} \mathbf{U}\mathbf{V}^T & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \mathbf{R}_\phi = \begin{bmatrix} \mathbf{V}^T & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$$

Since $\mathbf{R}_\theta$ is calculated for both pick and place operations ($\mathbf{R}_\theta^{pick}$, $\mathbf{R}_\theta^{place}$), the pick and place rotation angles of the objects are extracted,

$$\theta_{pick} = \arctan\left(\frac{\mathbf{R}_\theta^{pick}(2,2)}{\mathbf{R}_\theta^{pick}(2,1)}\right) \tag{4.4}$$

$$\theta_{place} = \arctan\left(\frac{\mathbf{R}_\theta^{place}(2,2)}{\mathbf{R}_\theta^{place}(2,1)}\right). \tag{4.5}$$

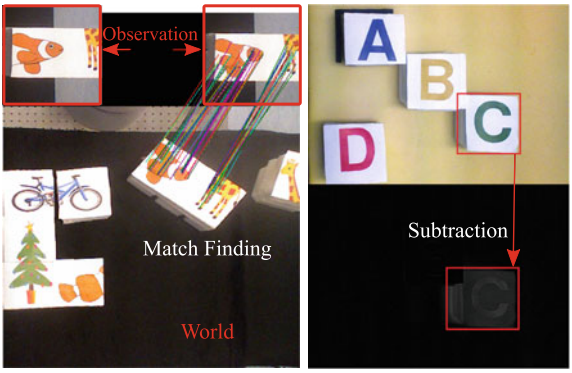

Fig. 4.3 The result of the image subtracting and thresholding for a place action (*right*), match finding result between the 4th observation and the world in the 4th operation during reproduction of the Domino task using SIFT (*left*). See Sect. 4.5.2 for more details

Note that projective transformation is position-dependent compared to the position-independent affine transformation. More details about homography estimation and decomposition can be found in Wong et al. (2007).

VSL relies on vision, which might be obstructed by other objects, by the tutor's body, or during the reproduction by the robot's arm. Therefore, for physical implementation of the VSL approach special care needs to be taken to avoid such obstructions. Finally, we should mention that the image processing part is not the focus of our research, and we use the SIFT-RANSAC algorithms because of their popularity and the capability of fast and robust match finding. Figure 4.3 shows the result of match finding using SIFT applied to an *observation* and a new *world*.

4.4.3 Trajectory Generation

In order to perform a pick-and-place operation, the robot must execute a set of actions consisting of reaching, grasping, relocating, and releasing. Either reaching and relocating actions correspond to a trajectory. These trajectories can either be manually programmed into the system or a tutor can teach them to the robot for instance through learning by demonstration technique (Ahmadzadeh et al. 2015). However, a simple trajectory generation strategy has been used in this chapter. The pick and place points together with the pick and place rotation angles extracted from the image processing section, are used to generate a trajectory for the corresponding operation. For each pick-and-place operation the desired Cartesian trajectory of the end-effector is a cyclic movement between three key points: rest point, pick point, and place point. Figure 4.4a illustrates a complete trajectory generated for a pick-and-place operation. Also, four different profiles of rotation angles are depicted in Fig. 4.4b. The robot starts from the rest point while the rotation angle is equal to zero and moves smoothly along the red curve towards the pick point. During this movement the robot's hand rotates to satisfy the pick rotation angle according to the rotation angle profile. Then the robot picks up an object, relocates it along the green

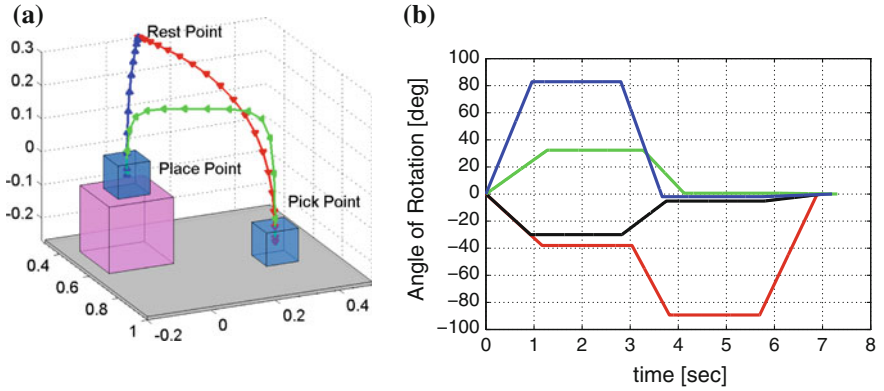

Fig. 4.4 The generated trajectory including position and orientation profiles for a spatial pick-and-place. **a** A full cycle of spatial trajectory generated for a pick-and-place operation. **b** The generated angle of rotation, θ, for the robot's hand

curve to the place-point, while the hand is rotating to meet the place rotation angle. Then, the robot places the object in the place-point, and finally moves back along the blue curve to the rest-point. The initial and final rotation angles are considered to be zero. In order to form each trajectory, initial conditions (i.e. initial positions and velocities) and a specific duration must be defined. Thereby, a geometric path is defined which can be expressed in the parametric form of the following equations:

$$p_x = a_3 s^3 + a_2 s^2 + a_1 s + a_0, \tag{4.6}$$

$$p_y = b_3 s^3 + b_2 s^2 + b_1 s + b_0, \tag{4.7}$$

$$p_z = h[1 - |(\tanh^{-1}(h_0(s - 0.5)))^{\kappa}|], \tag{4.8}$$

where, s is a function of time t, $(s = s(t))$, $p_x = p_x(s)$, $p_y = p_y(s)$, and $p_z = p_z(s)$ are the 3D elements of the geometric spatial path; The a_i and b_i coefficients in (4.6) and (4.7) are calculated using the initial and final conditions. κ in (4.8) denotes the curvature, h_0 and h are initial height and height of the curve in the middle point of the path respectively. h and h_0 can be either provided by the tutor or detected through depth information provided by an RGB-D sensor. In addition, the time is smoothly distributed with a 3rd order polynomial between the t_{start} and t_{final} which both are instructed by the user.

Moreover, to generate a rotation angle trajectory for the robot's hand, a trapezoidal profile is used together with the θ_{pick} and θ_{place} angles calculated in (4.4), (4.5). As shown in Fig. 4.4a, the trajectory generation module can also deal with objects placed in different heights (different z-axis levels). We discuss the grasp and release actions in the next section.

4.4.4 Grasp Synthesis

There are many elaborate ways to do grasp synthesis for known or unknown objects (Bohg et al. 2014; Su et al. 2012). Since the problem of grasping is not the main focus of our research, we implement a simple but efficient grasping method using the torque sensor of the Barrett Hand. The grasping position is calculated using the center of the corresponding *pre-action* observation. The grasping module firstly opens all the fingers and after the hand is located above the desired object, closes them. The fingers stop closing when the measured torque is more than a pre-defined threshold value. In addition, by estimating a bounding box for the target observation, the values are used to decide which axis is more convenient for grasping.

4.5 Experimental Results

In this section a set of simulated and real-world experiments are carried out. The simulated experiments are designed to gain an understanding of how VSL operates and to show the main idea of VSL without dealing with practical limitations and implementation difficulties. The real-world experiments, on the other hand, are carried out to show the main capabilities and limitations of VSL in practice while dealing with uncertainties. Before describing the conducted experiments, it is worth noting that, in all the illustrations, the straight and curved arrows are used just to show the sequence of operations, not the actual trajectories for performing the movements.

4.5.1 Simulated Experiments

In this section three simulated experiments are performed to illustrate the main idea behind the VSL approach. For each simulated experiment a set of 2D objects is made which the tutor can manipulate and assemble them on an empty workspace using keyboard or mouse. Each operation consists of a *pick* and a *place* action, which are executed by holding and releasing a mouse button.

A House Scene

In the first VSL task, the *world* includes seven 2D objects with different colors. However, the objects are detected based on their shapes not their color features. The *world* also includes a fixed baseline (i.e. a landmark $\mathcal{L}$) which cannot be manipulated. The goal is to assemble the set of objects on the baseline according to the demonstration. This task emphasizes VSL's capability of relative positioning of an object with respect to other surrounding objects in the *world*. This inherent capability of VSL is achieved through the use of visual *observations* which capture both the object of interest and its surrounding objects (i.e. its context). In addition, the baseline is provided to show the capability of absolute positioning of the VSL approach. It shows the

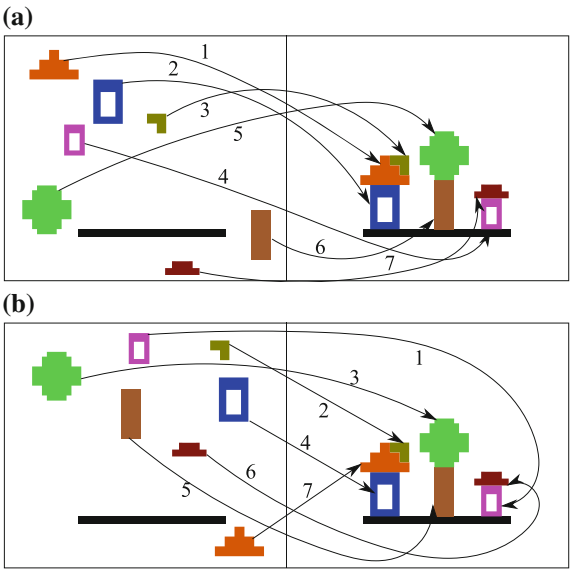

Fig. 4.5 House scene simulated experiment to illustrate the relative and absolute positioning capabilities of VSL. **a** Demonstration. **b** Reproduction

fact that we can teach the robot to attain absolute positioning of objects without defining any explicit a priori knowledge. Figure 4.5a shows the sequence of operations in the demonstration phase. Recording *pre-pick* and *pre-place observations*, $\mathcal{O}^{Pre}$, $\mathcal{O}^{Post}$, the robot learns the spatial relationships among objects. Figure 4.5b shows the sequence of operations produced by VSL on a novel *world* which is achieved by reshuffling the objects randomly in the *world*.

Roof Placement

In this task, the *world* includes two sets, each containing three visually identical objects (i.e. three blue 'house bodies' and three brown 'roofs'). As can be seen in Fig. 4.6a, the tutor selects the 'roof' objects arbitrarily and places them on top of the 'bodies'. However, in the 3rd operation, the tutor intentionally puts the 'roof' at the bottom of the 'house body'. The goal of this experiment is to show the VSL's capability of disambiguation of multiple alternative matches. If the algorithm uses a fixed search *frame* ($\mathcal{F}_R$) that is smaller than the size of the 'bodies' (i.e. blue objects in the *world*), then, as shown in the first and second sub-figures of Fig. 4.6b, the two captured observations can become equivalent (i.e. $\mathcal{O}_1 = \mathcal{O}_2$) and the 3rd operation might be performed incorrectly (see the incorrect reproduction in Fig. 4.6c). The reason is that, due to the size of the *frame*, the system perceives a section of the *world* not bigger than the size of a 'house body'. The system is not aware that an object is already assembled on the top and it will select the first matching *pre-place* position to place the last object there. To resolve this problem we use adaptive size *frames* during the match finding process in which the size of the *frame* starts from the smallest feasible size and grows up to the size of the *world*. The function findBestMatch in Algorithm 4.7, is responsible for creating and changing the size of the *frame*

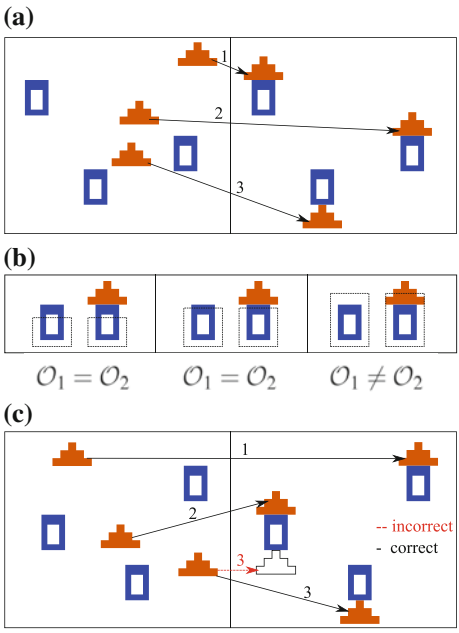

Fig. 4.6 Roof placement simulated experiment to illustrate VSL's capability of disambiguation of multiple alternative matches. **a** Demonstration. **b** Three steps of adaptive frame-size. **c** Reproduction

adaptively in each step. This technique helps the robot to resolve the ambiguity issue by adding more context inside the observation, which in effect narrows down the possible matches and leaves only a single matching observation. Figure 4.6c shows the sequence of reproduction including correct and incorrect operations.

Tower of Hanoi

The last simulated experiment is the famous mathematical puzzle, Tower of Hanoi, which consists of a number of disks of different sizes and three bases or rods which actually are landmarks. The objective of the puzzle is to move the entire stack to another rod. This experiment demonstrates almost all capabilities of the VSL approach. Two of these capabilities are not accompanied by the previous experiments. Firstly, our approach enables the user to intervene to modify the reproduction. Such capability can be used to move the disks to another base (e.g. to move the stack of disks to the third base, instead of the second). This can be achieved only if the user performs the very first operation in the reproduction phase and moves the smallest disk on the third base instead of the second. Secondly, VSL enables the user to perform multiple operations on the same object during the learning task. Figure 4.7 illustrates the reproduction sequence of the Tower of Hanoi puzzle, including three disks.

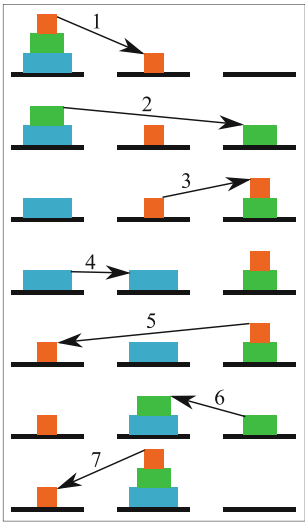

Fig. 4.7 Tower of Hanoi simulated experiment to illustrate the capabilities of VSL to perform interactive reproduction, disambiguation, and absolute positioning

4.5.2 Real-World Experiments

After performing some simulated experiments that illustrate the main idea of VSL, in order to show the capabilities and limitations of the proposed learning approach, in this section five real-world experiments are conducted. Table 4.1 summarizes the capabilities of VSL which are emphasized in each task.

Experimental Setup

As can be seen in Fig. 4.8, the experimental setup for all the conducted real-world experiments consists of a torque-controlled 7 DOF Barrett WAM robotic arm

Table 4.1 Capabilities of VSL illustrated in each real-world experiment

Capability	**Task**				
	Animal Puzzle	Alphabet Ordering	Tower of Hanoi	Animals versus Machines	Domino
Relative positioning	✓	✓	✓	-	✓
Absolute positioning	–	✓	✓	–	–
Classification	–	–	–	✓	–
Turn-taking	–	–	✓	✓	✓
User intervention	–	–	✓	✓	✓

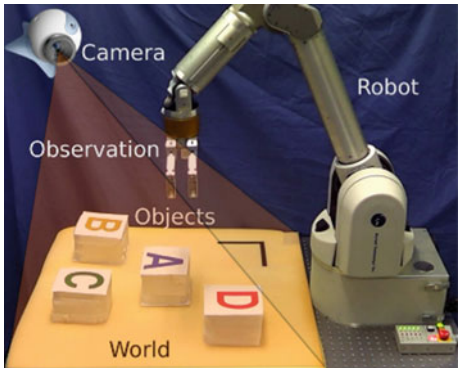

Fig. 4.8 The experimental setup for a Visuospatial Skill Learning (VSL) task

equipped with a 3-finger Barrett Hand, a tabletop working area, a set of objects, and a CCD camera which is mounted above the workspace (not necessarily perpendicular to the workspace).

In all the conducted experiments, the robot learns simple object manipulation tasks including pick-and-place actions. In order to perform a pick-and-place operation, the extracted pick and place poses are used to make a cyclic trajectory as explained in Sect. 4.4.3. The grasp strategy is implemented based on the method explained in Sect. 4.4.4. In the demonstration phase, the size of the *frame* for the *pre-pick observation* is set equal to the size of the biggest object in the *world*, and the size of the *frame* for the *pre-place observation* 2–3 times bigger than the size of the biggest objects in the *world*. In the reproduction phase, on the other hand, the size of the *frame* is set equal to the size of the *world*.

The vision sensor is mounted above the table facing the workspace. The resolution of the captured images are 1280×960 pixels. Although the trajectories are created in the end-effector space, we control the robot in the joint-space based on the inverse dynamics to avoid singularities. Also, during the reproduction phase, our controller keeps the orientation of the robot's hand (end-effector) perpendicular to the workspace plane, in order to facilitate the pick-and-place operation.

Alphabet Ordering

In the first real-world VSL task, the *world* includes four cubic objects labeled with A, B, C, and D letters. Similar to the first simulated experiment the *world* also includes a fixed right angle baseline which is a landmark ($\mathcal{L}$). The goal is to reconfigure and sort the set of objects with respect to the baseline according to the demonstration. As reported in Table 4.1, this task emphasizes VSL's capability of relative positioning of an object with respect to other surrounding objects in the *world* (a visuospatial skill). This inherent capability of VSL is achieved through the use of visual *observations* which capture both the object of interest and its surrounding objects (i.e. its context). In addition, the baseline is provided to show the capability of absolute positioning of the VSL approach. It shows the fact that we can teach the robot to attain absolute

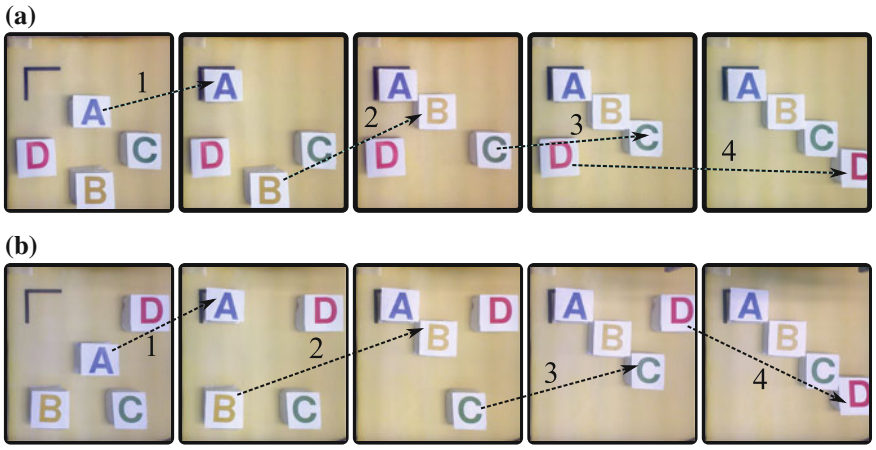

Fig. 4.9 Alphabet ordering. The initial configuration of the objects in the *world* is different in (**a**) and (**b**). The *black arrows* show the operations. **a** The sequence of the operations in the demonstration phase by the tutor. **b** The sequence of the operations in the reproduction phase by the robot

positioning of objects without defining any explicit a priori knowledge. Figure 4.9a shows the sequence of operations in the demonstration phase. Recording *pre-pick* and *pre-place observations*, the robot learns the sequence of operations. Figure 4.9b shows the sequence of operations produced by VSL starting from a novel *world* (i.e. new initial configuration) which is achieved by randomizing the objects in the *world*.

Animal Puzzle

In the previous task, due to the absolute positioning capability of VSL, the final configuration of the objects in the reproduction and the demonstration phases are always the same. In this experiment, however, by removing the fixed baseline from the *world*, the final result can be a totally new configuration of objects. The goal of this experiment is to show the VSL's capability of relative positioning which is reported in Table 4.1. In this VSL task, the *world* includes two sets of objects which complete a 'frog' and a 'giraffe' puzzle. There are also two labels (i.e. landmarks) in the *world*, a 'pond' and a 'tree'. The goal is to assemble the set of objects for each animal with respect to the labels according to the demonstration. Figure 4.10a shows the sequence of operations in the demonstration phase. To show the capability of generalization, the 'tree' and the 'pond' labels are randomly replaced by the tutor before the reproduction phase. Figure 4.10b shows the sequence of operations reproduced by VSL after learning the spatial relationships among objects.

Tower of Hanoi

In this experiment, the Tower of Hanoi puzzle is performed again, this time in real-world. As mentioned before, this experiment demonstrates almost all capabilities of VSL comprising relative and absolute positioning, user intervention to modify the

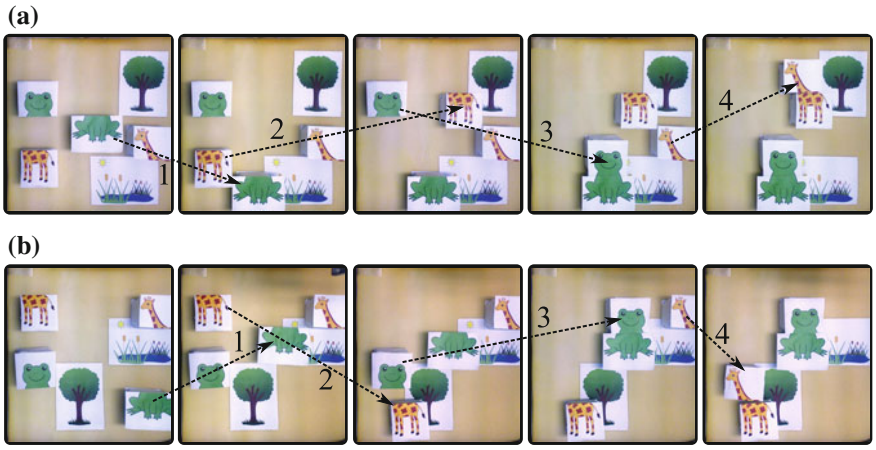

Fig. 4.10 Animal puzzle. The initial and the final configurations of the objects in the *world* are different in (**a**) and (**b**). The *black arrows* show the operations. **a** The sequence of the operations in the demonstration phase by the tutor. **b** The sequence of the operations in the reproduction phase by the robot

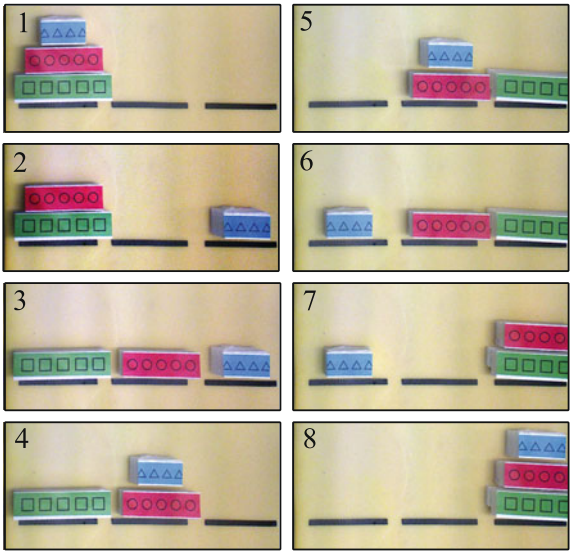

Fig. 4.11 The sequence of the reproduction for the Tower of Hanoi experiment to illustrate the main capabilities of VSL

reproduction, and multiple operations performed on the same object. The sequence of reproduction is shown in Fig. 4.11.

Animals versus Machines: A Classification Task

In this interactive task we demonstrate the VSL capability of classification of objects. We provided the robot with four objects, two 'animals' and two 'machines'.

Also, two labeled bins are used in this experiment for classifying the objects. Similar to previous tasks, the objects, labels and bins are not known to the robot initially. In this task, firstly, all the objects are randomly placed in the *world*. The tutor randomly picks objects one by one and places them in the corresponding bins. In the reproduction phase, the tutor places one of the objects each time, in a different sequence with respect to the demonstration. The robot detects the object and classifies it. This is an interactive task between the human and the robot. The human tutor can modify the sequence of operations in the reproduction phase by presenting the objects to the robot in a different order with respect to the demonstration.

Interestingly the same VSL algorithm is utilized to learn a classification task. However in this task the robot doesn't follow the operations sequentially but searches in the *pre-pick observation* dictionary to find the best matching *pre-pick observation*. Then, it uses the selected *pre-pick observation* for reproduction as before. The sequence of operations in the demonstration phase are illustrated in Fig. 4.12. Each row represents one pick-and-place operation. During each operation, the tutor picks an object and moves it to the proper bin. The set of *pre-pick* and *pre-place observation* can be seen in left and right columns respectively. The match finding process is done using SIFT. Figure 4.13 shows two operations during the reproduction phase.

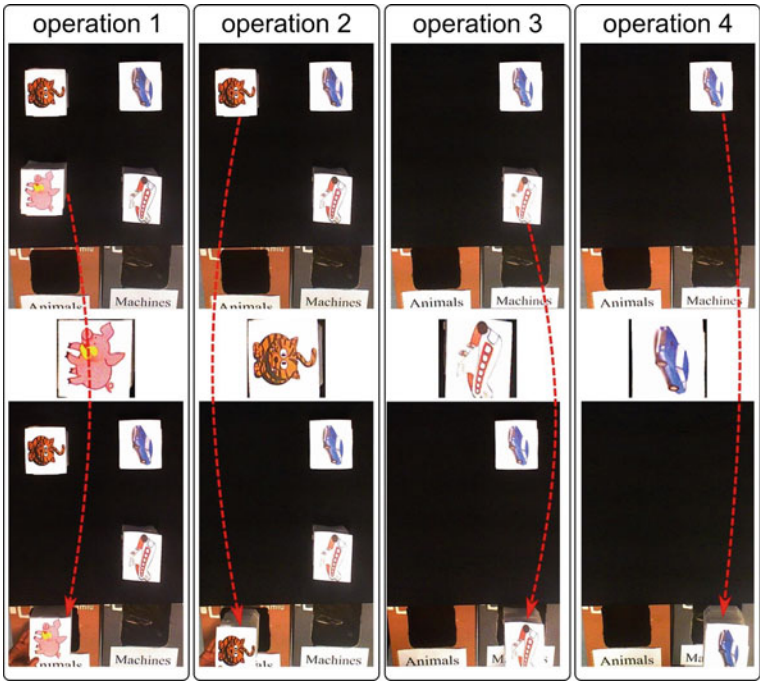

Fig. 4.12 The sequence of operations in the demonstration phase. Each *column* represents one pick-and-place operation. In each operation, the tutor picks one object and classifies it either as an '*animal*' or a '*machine*'. The selected object in each operation is shown in the *middle row*

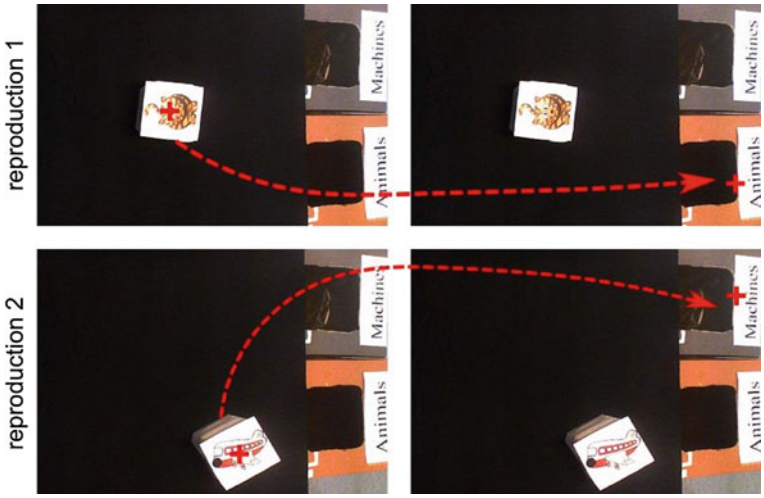

Fig. 4.13 Two operations during the reproduction phase are shown. The *red crosses* on the objects and on the bins, show the detected positions for pick and place actions respectively

Domino: A Turn-Taking Task

The goal of this experiment is to show that VSL can also deal with the tasks including the cognitive behaviour of turn-taking. In this VSL task, the *world* includes a set of objects all of which are rectangular tiles. Each two pieces of the puzzle fit together to form an object (see Fig. 4.14). In the demonstration phase, the tutor first demonstrates all the operations. To learn the spatial relationships, the system uses the modified algorithm from the classification task. In the reproduction phase, the tutor starts the game by placing the first object (or another) in a random place. The robot then takes the turn, finds and places the next matching domino piece. The tutor can also modify the sequence of operations in the reproduction phase by presenting the objects to the robot in a different order with respect to the demonstration. The sequence of reproduction is shown in Fig. 4.14.

Fig. 4.14 The sequence of reproduction performed by the robot and the tutor are shown for the turn-taking task of domino

Results

Table 4.1 summarizes the main capabilities of VSL which are emphasized in each real-world experiment.

In order to test the repeatability of VSL and to identify the possible factors of failure, the captured observations from the real world experiments were used while excluding the robot from the loop. All other parts of the loop were kept intact and each experiment was repeated three times. The result shows that less than 5 % of pick-and-place operations failed. The main failure factor is the match finding error which can be resolved by adjusting the parameters of SIFT-RANSAC or using alternative match finding algorithms. The noise in the images and the occlusion of the objects can be listed as two other potential factors of failure. Despite the fact that VSL is scale-invariant, color-invariant, and view-invariant, it has some limitations. For instance, if the tutor accidentally moves one object while operating another, the algorithm may fail to find a pick/place position. One possible solution is to combine classification techniques together with the image subtraction and thresholding techniques to detect multi-object movements.

4.6 Conclusions

In this chapter, a visuospatial skill learning approach has been introduced that has powerful capabilities as shown in the simulated and real-world experiments. The introduced approach possesses capabilities such as relative and absolute positioning, user intervention to modify the reproduction, classification, and turn-taking. These characteristics make VSL a suitable choice for interactive robot learning tasks which rely on visual perception. Moreover, VSL is convenient for the vision-based robotic platforms which are designed to perform a variety of repetitive and interactive production tasks (e.g. Baxter). The main reason is that applying VSL to such platforms, requires neither complex programming skills nor costly integration.

References

Abbeel P, Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In: 21st International Conference on Machine learning (ICML), ACM, p 1

Ahmadzadeh SR, Kormushev P, Caldwell DG (2013a) Interactive robot learning of visuospatial skills. In: IEEE 16th International Conference on Advanced Robotics (ICAR), pp 1–8

Ahmadzadeh SR, Kormushev P, Caldwell DG (2013b) Visuospatial skill learning for object reconfiguration tasks. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 685–691

Ahmadzadeh SR, Paikan A, Mastrogiovanni F, Natale L, Kormushev P, Caldwell DG (2015) Learning symbolic representations of actions from human demonstrations. In: IEEE International Conference on Robotics and Automation (ICRA), Seattle, Washington

Argall BD, Chernova S, Veloso M, Browning B (2009) A survey of robot learning from demonstration. Robot Auton Syst 57(5):469–483

Asada M, Yoshikawa Y, Hosoda K (2000) Learning by observation without three-dimensional reconstruction. Intelligent Autonomous Systems (IAS-6) pp 555–560

Bentivegna DC, Atkeson CG, Ude A, Cheng G (2004) Learning to act from observation and practice. Int J Humanoid Robot 1(4):585–611

Bohg J, Morales A, Asfour T, Kragic D (2014) Data-driven grasp synthesis: a survey. IEEE Trans Robot 30:289–309

Chao C, Cakmak M, Thomaz AL (2011) Towards grounding concepts for transfer in goal learning from demonstration. In: IEEE International Conference on Development and Learning (ICDL), vol 2, pp 1–6

Dantam N, Essa I, Stilman M (2012) Linguistic transfer of human assembly tasks to robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 237–242

Ehrenmann M, Rogalla O, Zöllner R, Dillmann R (2001) Teaching service robots complex tasks: programming by demonstration for workshop and household environments. In: International Conference on Field and Service Robots (FSR), vol 1, pp 397–402

Ekvall S, Kragic D (2008) Robot learning from demonstration: a task-level planning approach. Int J Adv Robot Syst 5(3):223–234

Feniello A, Dang H, Birchfield S (2014) Program synthesis by examples for object repositioning tasks. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 4428–4435

Guadarrama S, Riano L, Golland D, Gouhring D, Jia Y, Klein D, Abbeel P, Darrell T (2013) Grounding spatial relations for human-robot interaction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 1640–1647

Hartley R, Zisserman A (2000) Multiple view geometry in computer vision. Cambridge University Press, cambridge

Ijspeert AJ, Nakanishi J, Schaal S (2002) Learning attractor landscapes for learning motor primitives. Adv Neural Inf Process Syst 15:1523–1530

Ijspeert AJ, Nakanishi J, Hoffmann H, Pastor P, Schaal S (2013) Dynamical movement primitives: learning attractor models for motor behaviors. Neural Comput 25(2):328–373

Ikeuchi K, Suehiro T (1994) Toward an Assembly Plan from Observation. I. Task Recognition with Polyhedral Objects. IEEE Trans Robot Autom 10(3):368–385

Kormushev P, Calinon S, Caldwell DG (2010) Robot motor skill coordination with EM-based reinforcement learning. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 3232–3237

Kormushev P, Calinon S, Caldwell DG (2011) Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. Adv Robot 25(5):581–603

Kronander K, Billard A (2012) Online learning of varying stiffness through physical human-robot interaction. In: IEEE International Conference on Robotics and Automation (ICRA), pp 1842–1849

Kuniyoshi Y, Inaba M, Inoue H (1994) Learning by watching: extracting reusable task knowledge from visual observation of human performance. IEEE Trans Robot Autom 10(6):799–822

Lopes M, Santos-Victor J (2005) Visual learning by imitation with motor representations. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 35(3):438–449

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110

Meltzoff AN, Moore MK (1977) Imitation of facial and manual gestures by human neonates. Science 198(4312):75–78

Niekum S, Osentoski S, Konidaris G, Barto AG (2012) Learning and generalization of complex tasks from unstructured demonstrations. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 5239–5246

Park G, Ra S, Kim C, Song J (2008) Imitation learning of robot movement using evolutionary algorithm. In: 17th World Congress, International Federation of Automatic Control (IFAC), pp 730–735

Rizzolatti G, Fadiga L, Gallese V, Fogassi L (1996) Premotor cortex and the recognition of motor actions. Cognit Brain Res 3(2):131–141

Schaal S (1999) Is imitation learning the route to humanoid robots? Trends Cognit Sci 3(6):233–242

Su Y, Wu Y, Lee K, Du Z, Demiris Y (2012) Robust grasping for an under-actuated anthropomorphic hand under object position uncertainty. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp 719–725

Verma D, Rao R (2005) Goal-based imitation as probabilistic inference over graphical models. Adv Neural Inf Process Syst 18:1393–1400

Vijayakumar S, Schaal S (2000) Locally weighted projection regression: an O(n) algorithm for incremental real time learning in high dimensional space. In: 17th International Conference on Machine Learning (ICML), vol 1, pp 288–293

Wong TY, Kovesi P, Datta A (2007) Projective transformations for image transition animations. In: 14th IEEE International Conference on Image Analysis and Processing, ICIAP, pp 493–500

Yeasin M, Chaudhuri S (2000) Toward automatic robot programming: Learning human skill from visual data. IEEE Trans Syst Man Cybern Part B: Cybern 30(1):180–185