Interactive Robot Learning of Visuospatial Skills

Seyed Reza Ahmadzadeh, Petar Kormushev and Darwin G. Caldwell
Department of Advanced Robotics
Istituto Italiano di Tecnologia
via Morego 30, 16163, Genova
Email: {reza.ahmadzadeh, petar.kormushev, darwin.caldwell}@iit.it

***Abstract*—This paper proposes a novel interactive robot learning approach for acquiring visuospatial skills. It allows a robot to acquire new capabilities by observing a demonstration while interacting with a human caregiver. Most existing learning from demonstration approaches focus on the trajectories, whereas in our approach the focus is placed on achieving a desired goal configuration of objects relative to one another. Our approach is based on visual perception which captures the object's context for each demonstrated action. The context embodies implicitly the visuospatial representation including the relative positioning of the object with respect to multiple other objects simultaneously. The proposed approach is capable of learning and generalizing different skills such as object reconfiguration, classification, and turn-taking interaction. The robot learns to achieve the goal from a single demonstration while requiring minimum *a priori* knowledge about the environment. We illustrate the capabilities of our approach using four real world experiments with a Barrett WAM robot.**

I. Introduction

Several robot skill learning approaches based on human demonstrations have been proposed during the past decade. Motor skill learning approaches can be categorized in two main subsets: trajectory-based and goal-based approaches. Trajectory-based approaches put the focus on recording and re-generating trajectories and forces for object manipulation [1], [2]. However, in many cases, it is not the trajectory that is important but the goal of the action, for example, solving a jigsaw puzzle [3] or assembling an electric circuit board. In such examples, trajectory-based approaches actually increase the complexity of the learning process unnecessarily.

Several goal-based approaches such as [4] have been developed to address this issue. For instance, there is a large body of literature on grammars from the linguistic and computer science communities, with a number of applications related to robotics [5], [6]. Other symbolic learning approaches are focused on goal configurations rather than action execution [7]. Such approaches inherently comprise many steps, for instance, segmentation, clustering, object recognition, structure recognition, symbol generation, syntactic task modeling, motion grammar and 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.

Furthermore, most above-mentioned approaches assume the availability of the information on the internal state of a demonstrator such as joint angles, while humans usually cannot directly access to imitate the observed behavior.

An alternative to motor skill learning approaches are visual skill learning approaches [8], [9]. These approaches are based on observing the human demonstration and using human-like visuospatial skills to replicate the task [10]. Visuospatial skill is the capability to visually perceive the spatial relationship between objects.

In this paper, we propose a novel visuospatial skill learning approach for interactive robot learning tasks. Unlike the motor skill learning approaches, our approach utilizes visual perception as the main information source for learning new skills from demonstration. The proposed visuospatial skill learning (VSL) approach uses a simple algorithm and 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 four 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 real world experiments, the robot acquires and reproduces three main capabilities: object reconfiguration; classification; and turn-taking. This work is an extension of our previous research on visuospatial skill learning [11], with two major novelties: (*i*) ability to learn and reproduce not only the position but also the orientation of objects; and (*ii*) application to more challenging tasks including object classification and human-robot turn-taking.

II. Related Work

Visual skill learning or learning by watching is one of the most powerful mechanisms of learning in humans. Researchers have shown that even newborns can imitate simple body movements such as facial gestures [12]. In cognitive science, learning by watching has been investigated as a source of higher order intelligence and fast acquisition of knowledge [8], [13]. In the rest of this section we give examples for visual skill learning approaches.

In [8], a robot agent watches a human teacher performing a simple assembly task in a tabletop environment. The motor movements of the human are classified as actions known to the robot (pick, move, place etc.). The robot can reproduce the sequence of actions even if the initial configuration of the objects are changed. In their approach, the set of actions is already known to the robot. And they use symbolic labels to reproduce the sequence of actions. In order to detect the movement of an object, they detect and track the demonstrator's hand. Since they rely solely on passive observations of a teacher demonstration, this method has to make use of complex computer vision techniques, in carefully structured

environments, in order to infer all the information necessary for the task. In our method we do not use symbolic labels.
Asada et al. [14] propose a method for learning by observation (teaching by showing) based on the demonstrator's view recovery and adaptive visual servoing. They believe that coordinate transformation is a time-consuming and error-prone method. Instead, they assume that both the robot and the demonstrator have the same body structure. They use two sets of stereo cameras, one for observing the robot's motion and the other for observing the demonstrator's motion. 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.
In [15], the authors propose 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 is used in their experiment. 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 symbol 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 from demonstration, generating trajectory and extracting subtasks [16]. They use color markers to capture data from the caregiver's hand. In our method, we use neither neural networks nor symbol abstraction techniques.
A visual learning by Imitation approach is presented in [9]. 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. They used a single camera in their experiments. A method for segmenting demonstrations, recognizing repeated skills, and generalizing complex tasks from unstructured demonstration is presented in [5]. The authors use Beta Process Autoregressive HMM for recognizing and generalizing. They apply simple metrics on the captured object's context to distinguish between observations instead. For object detection they use pre-defined coordinated frames and visual fiducial fixed on each object. Furthermore, they clustered points in each frame and for each segment in the demonstration a DMP is created. In our method we use captured object's context to find pick and place points for each operation instead of object recognition and point clustering.Visual analysis of demonstrations and automatic policy extraction for an assembly task is presented in [6]. 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.
A robot goal learning approach is presented in [7] that can ground discrete concepts from continuous perceptual data using unsupervised learning. The authors provided the demonstrations of five pick-and-place tasks in a tabletop workspace by non-expert users. They utilize object detection, segmentation, and localization methods using color markers. During each operation, the approach detects the object that changes most significantly. They provide the starting and ending time of each demonstration 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.

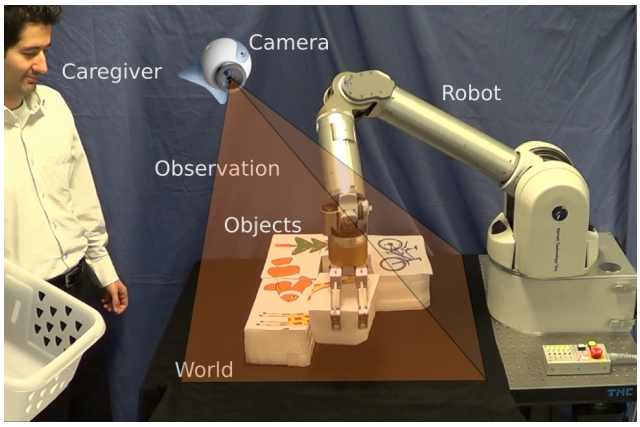


Fig. 1. The experimental setup for a visuospatial skill learning (VSL) task.

III. Visuospatial Skill Learning

In this section, we introduce the visuospatial skill learning (VSL) approach. After stating our assumptions and defining the basic terms, we describe the problem statement and explain the VSL algorithm.
As can be seen in Fig. 1 the experimental set-up for all the conducted experiments consists of a torque-controlled 7-DOF Barrett WAM robotic arm with 3-finger Barrett Hand attached, a tabletop working area, a set of objects, and a CCD camera which is mounted above the workspace (not necessarily perpendicular to the workspace).
A human caregiver demonstrates a sequence of operations on the available objects. Each operation consists of one pick action and one place action which are captured by the camera. Afterwards, using our proposed method, and starting from a random initial configuration of the objects, the robot can perform a sequence of new operations which ultimately results in reaching the same goal as the one demonstrated by the caregiver.

A. Terminology

First, we need to define some basic terms to describe our learning approach accurately.
***World*:** In our method, the intersection area between the robot's workspace and the caregiver's workspace which is observable by the camera (with a specific field of view) is called a *world*. The *world* includes objects which are being used during the learning task, and can be reconfigured by the human caregiver and the robot.
***Frame*:** A bounding box which defines a cuboid in 3D or a rectangle in 2D coordinate system. 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 and using a specific *frame*. Each *observation* is a still image which holds the content of that part of the

world which is located inside the *frame*.
***Pre-pick observation*:** An *observation* which is captured just before the pick action and is centered around the pick location.
***Pre-place observation*:** An *observation* which is captured just before the place action and is centered around the place location.
The *pre-pick* and *pre-place* terms in our approach are analogous to the *precondition* and *postcondition* terms used in logic programming languages (e.g. Prolog).

B. Problem Statement

Formally, we define a process of visuospatial skill learning as a tuple $\mathcal{V} = \{\eta, \mathcal{W}, \mathcal{O}, \mathcal{F}, \mathcal{S}\}$ where, η defines the number of operations which can be specified in advance or can be detected during the demonstration phase; $\mathcal{W} \in \Re^{m \times n}$ is a matrix containing the image which represents the context of the *world* including the workspace and all objects; $\mathcal{O}$ is a set of *observation* dictionaries $\mathcal{O} = \{\mathcal{O}_{Pick}, \mathcal{O}_{Place}\}$; $\mathcal{O}_{Pick}$ is an *observation* dictionary comprising a sequence of *pre-pick observations* $\mathcal{O}_{Pick} = \langle \mathcal{O}_{Pick}(1), \mathcal{O}_{Pick}(2), \ldots, \mathcal{O}_{Pick}(\eta) \rangle$, and $\mathcal{O}_{Place}$ is an *observation* dictionary comprising a sequence of *pre-place observations* $\mathcal{O}_{Place} = \langle \mathcal{O}_{Place}(1), \mathcal{O}_{Place}(2), \ldots, \mathcal{O}_{Place}(\eta) \rangle$. For example, $\mathcal{O}_{Pick}(i)$ represents the *pre-pick observation* captured during the i^{th} operation. $\mathcal{F} \in \Re^{m \times n}$ is an *observation frame* which is used for recording the *observations*; and $\mathcal{S}$ is a vector which stores scalar scores related to each detected match during a match finding process. The score vector is calculated using a metric by comparing the new observations captured during the reproduction phase with the dictionary of recorded observations in the demonstration phase.
The output of the reproduction phase are the set $\mathcal{P} = \{\mathcal{P}_{Pick}, \mathcal{P}_{Place}\}$ and the set $\theta = \{\theta_{Pick}, \theta_{Place}\}$ which contain the identified positions and orientations for performing the pick-and-place operations by the VSL algorithm.
$\mathcal{P}_{Pick}$ is an ordered set of *pre-pick* positions $\mathcal{P}_{Pick} = \langle \mathcal{P}_{Pick}(1), \mathcal{P}_{Pick}(2), \ldots, \mathcal{P}_{Pick}(\eta) \rangle$. $\mathcal{P}_{Place}$ is an ordered set of *pre-place* positions $\mathcal{P}_{Place} = \langle \mathcal{P}_{Place}(1), \mathcal{P}_{Place}(2), ..., \mathcal{P}_{Place}(\eta) \rangle$. θ_{Pick} is an ordered set of pick rotations $\theta_{Pick} = \langle \theta_{Pick}(1), \theta_{Pick}(2), \ldots, \theta_{Pick}(\eta) \rangle$ and θ_{Place} is an ordered set of place orientations $\theta_{Place} = \langle \theta_{Place}(1), \theta_{Place}(2), ..., \theta_{Place}(\eta) \rangle$.
For example, $\mathcal{P}_{Pick}(i)$ and and $\theta_{Pick}(i)$ represent the pick-position and the pick orientation during the i^{th} operation respectively.

C. Methodology

A high-level flow diagram illustrating the VSL approach is shown in Fig. 2. Pseudo-code of the proposed implementation is given in Algorithm 1.
The VSL approach consists of two phases: demonstration and reproduction. In both phases, initially, the objects are randomly placed in the *world*, $\mathcal{W} = \{\mathcal{W}_D, \mathcal{W}_R\}$. (So, VSL does not depend on initial configuration of the objects.) $\mathcal{W}_D$ and $\mathcal{W}_R$ represent the *world* during the demonstration and the reproduction phases respectively.
The initial step in each learning task is calculating the (2D to 2D) coordinate transformation (line 1 in Algorithm 1) which means computing a homography matrix for the current set-up

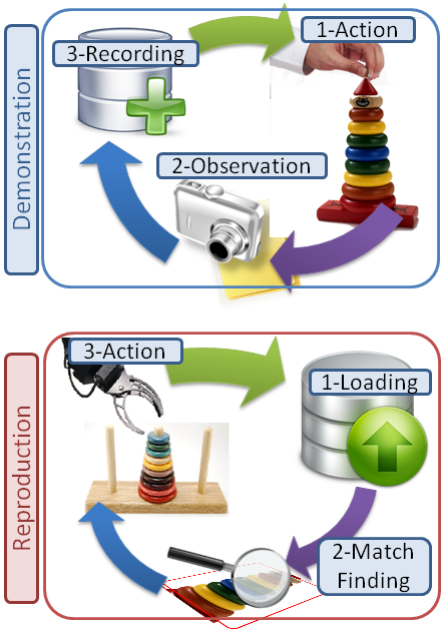


Fig. 2. A high-level flow diagram illustrating the demonstration and reproduction phases in the VSL approach.

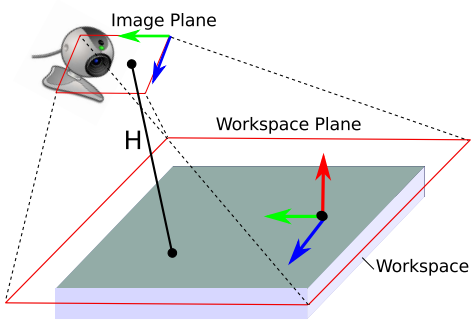


Fig. 3. Coordinate transformation from the image plane to the workspace plane (H represents the homography matrix).

of the robot and the camera, $\mathcal{H} \in \Re^{3 \times 3}$. This transformation matrix is used to transform points from the image plane to the workspace plane (Fig. 3). In the demonstration phase, the caregiver starts to reconfigure the objects to fulfill a desired goal which is not known to the robot. During this phase the size of the *observation frame*, $\mathcal{F}_D$, is fixed and can be equal or smaller than the size of the *world*, $\mathcal{W}_D$. Two functions, *RecordPrePickObs* and *RecordPrePlaceObs*, are developed to capture and rectify one *pre-pick observation* and one *pre-place observation* for each operation. The captured images are rectified using the calculated homography matrix $\mathcal{H}$. Using image subtraction and thresholding techniques, the two mentioned functions (lines 3 and 4 in Algorithm 1) extract a pick point and a place point in the captured *observations* and then center the *observations* around the extracted positions. These functions also extract the initial pick and place orientations of the objects in the *world*. The recorded sets of *observations*, $\mathcal{O}_{Pick}$ and $\mathcal{O}_{Place}$, together with the extracted orientations, form an *observation* dictionary which is used as the input for the reproduction phase of VSL.
Before starting the reproduction phase, if the position and

Input : $\{\eta, \mathcal{W}, \mathcal{F}\}$
Output: $\{\mathcal{P}, \theta\}$
1 **Initialization:** $CalculateTransformation$
// Part I : Demonstration
2 **for** $i = 1$ ***to*** η **do**
3 $\quad \mathcal{O}_{Pick}(i) = RecordPrePickObs(\mathcal{W}_D, \mathcal{F}_D)$
4 $\quad \mathcal{O}_{Place}(i) = RecordPrePlaceObs(\mathcal{W}_D, \mathcal{F}_D)$
5 **end**
// Part II : Reproduction
6 $Re-calculate\,Transformation(if\ necessary)$
7 **for** $i = 1$ ***to*** η **do**
8 $\quad \mathcal{F}^*$ = $FindBestMatch(\mathcal{W}_R, \mathcal{O}_{Pick}(i))$
9 $\quad \mathcal{P}_{Pick}(i) = FindPickPosition(\mathcal{F}^*)$
10 $\quad \theta_{Pick}(i) = FindPickRotation(\mathcal{F}^*)$
11 $\quad PickObjectFrom(\mathcal{P}_{Pick}(i), \theta_{Pick}(i))$
12 $\quad \mathcal{F}^*$ = $FindBestMatch(\mathcal{W}_R, \mathcal{O}_{Place}(i))$
13 $\quad \mathcal{P}_{Place}(i) = FindPlacePosition(\mathcal{F}^*)$
14 $\quad \theta_{Place}(i) = FindPlaceRotation(\mathcal{F}^*)$
15 $\quad PlaceObjectTo(\mathcal{P}_{Place}(i), \theta_{Place}(i))$
16 **end**

Algorithm 1: Pseudo-code for the VSL (Visuospatial Skill Learning) approach.

orientation of the camera with respect to the robot is changed we need to re-calculate the homography matrix for the new configuration of the coordinate systems (lines 6 in Algorithm 1). The next phase, reproduction, starts with randomizing the objects in the *world* to create a new *world*, $\mathcal{W}_R$. Then the algorithm selects the first *pre-pick observation* and searches for similar visual perception in the new *world* to find the best match (using the *FindBestMatch* function in lines 8 and 12). 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 section IV-B). This metric produces a scalar score with respect to each comparison. If more than one matching object is found, the *FindBestMatch* function returns the match with higher score. After the algorithm finds the best match, the *FindPickPosition* and *FindPickRotation* functions (line 9, 10) identify the pick position and rotation which the position is located at the center of the corresponding *frame*, $\mathcal{F}^*$. The *PickObjectFrom* uses the results from the previous step to pick the object from the identified position. Then, the algorithm repeats the searching process to find the best match with the corresponding *pre-place observation* (line 12). The *FindPlacePosition* and *FindPlaceRotation* functions identify the place position and rotation at the center of the corresponding *frame*, $\mathcal{F}^*$ and the *PlaceObjectTo* function uses the results from the previous step to place the object to the identified position. The *frame* size in the reproduction phase is fixed and equal to the size of the *world* ($\mathcal{W}_R$). One of the advantages of the VSL approach is that it provides the exact position of the object where the object should be picked without using any *a priori* knowledge about the object. The reason is, when the algorithm is generating the *observations*, it centers the *observation frame* around the point that the caregiver is operating the object.

IV. Implementation of VSL

In this section we describe the steps required to implement the VSL approach for real world experiments.

A. Calculating Coordinate Transformation (Homography)

In each experiment the camera's frame of reference may vary with respect to the robot's frame of reference. To transform points between these coordinate systems we calculate the homography $\mathcal{H}$. Using, at least, 4 real points on the workspace plane and 4 corresponding points on the image plane, the homography is calculated using Singular Value Decomposition (SVD) method [17]. The extracted homography matrix is used not only for coordinate transformation but also to rectify the raw captured images from the camera. For each task the process of extracting the homography should be repeated whenever the camera's frame of reference is changed with respect to the robot's frame of reference.
After the robot acquires a new skill, it still can reproduce the skill afterwards even if the experimental set-up is changed. The algorithm just needs to use the new coordinate transformation. It means that VSL is a view-invariant approach.

B. Image Processing

Image processing methods have been used in both demonstration and reproduction phases of VSL. In the demonstration phase, for each operation the algorithm captures a set of raw images consist of *pre-pick*, *pre-place* images. Firstly, the captured raw images are rectified using the homography matrix $\mathcal{H}$. 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 1). Although any metric can be used in this function (e.g. window search method), we use Scale Invariant Feature Transform (SIFT) algorithm [18]. 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 RANSAC in order to estimate the transformation matrix $\mathcal{H}_{sift}$ from the set of the matches. Since the calculated transformation matrix $\mathcal{H}_{sift}$ has 8 degrees of freedom, with 9 elements in the matrix, to have a unique normalized representation we pre-multiply $\mathcal{H}_{sift}$ with a normalization constant:

$$\alpha = \frac{sign(\mathcal{H}_{sift}(3,3))}{\sqrt{(\mathcal{H}_{sift}(3,1)^2 + \mathcal{H}_{sift}(3,2)^2 + \mathcal{H}_{sift}(3,3)^2)}} \quad (1)$$

This normalization constant is selected to make the decomposed projective matrix have a vanishing line vector of unit magnitude and that avoids unnatural interpolation results. The normalized matrix $\alpha\mathcal{H}_{sift}$ can be decomposed into simple transformation elements,

$$\alpha\mathcal{H}_{sift} = \mathcal{T}\mathcal{R}_\theta\mathcal{R}_{-\phi}\mathcal{S}_v\mathcal{R}_\phi\mathcal{P} \quad (2)$$

where $\mathcal{R}_{\pm\phi}$ are rotation matrices to align the axis for horizontal and vertical scaling of $\mathcal{S}_v$; $\mathcal{R}_\theta$ is another rotation matrix to

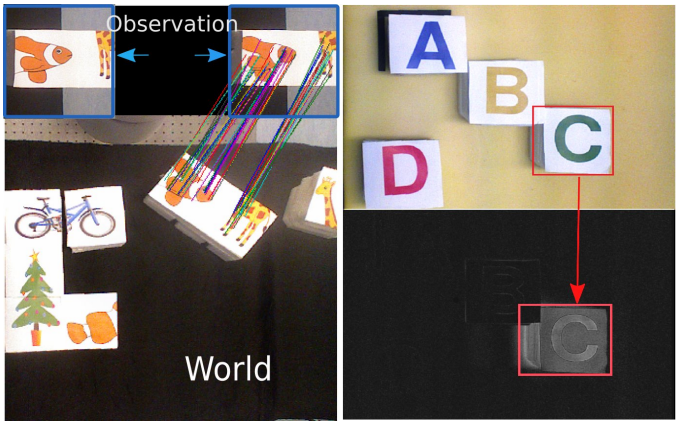


Fig. 4. The result of the image subtracting and thresholding for a place action (right), Match finding result between the 4th observation and the world in the 4th operation of the Domino task using SIFT (left).

orientate the shape into its final orientation; $\mathcal{T}$ is a translation matrix; and lastly $\mathcal{P}$ is a pure projective matrix:

$$\mathcal{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \alpha\mathcal{H}_{(3,1)} & \alpha\mathcal{H}_{(3,2)} & \alpha\mathcal{H}_{(3,3)} \end{bmatrix} \tag{3}$$

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

$$\mathcal{A} = UDV^T \tag{4}$$

where D is a diagonal matrix, and U and V are orthogonal matrices. Finally we can calculate

$$\mathcal{S}_v = \begin{bmatrix} D & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \mathcal{R}_\theta = \begin{bmatrix} UV^T & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \mathcal{R}_\phi = \begin{bmatrix} V^T & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \tag{5}$$

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

$$\theta_{pick} = \arctan(\mathcal{R}_\theta^{pick}(2,2)/\mathcal{R}_\theta^{pick}(2,1)) \tag{6}$$

$$\theta_{place} = \arctan(\mathcal{R}_\theta^{place}(2,2)/\mathcal{R}_\theta^{place}(2,1)) \tag{7}$$

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 [19].

VSL relies on vision, which might be obstructed by other objects, by the caregiver'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 this paper, and we use the SIFT-RANSAC algorithms because of their popularity and the capability of fast and robust match finding. Fig. 4 shows the result of the SIFT algorithm applying to an *observation* and a new *world*.

C. Trajectory Generation

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. Fig 5(a) illustrates two different views of a generated trajectory. Also, four different profiles of rotation angle are depicted in Fig 5(b). The robot starts from the rest point while the rotation angle is equal to zero (point no.1) and moves smoothly (along the red curve) towards the pick point (point no.2). During this movement the robot's hand rotates to reach the pick rotation angle according to the rotation angle profile. Then the robot picks up an object, relocates it (along the green curve) to the place-point (point no.3), while the hand is rotating to meet the place rotation angle. Next, the robot places the object in the place point, and finally moves back (along the blue curve) to the rest point. For each part of the trajectory, including the grasping and the releasing parts, we define a specific duration and initial boundary conditions (initial positions and velocities). In addition, we assumed that the initial and final rotation angles to be zero. We defined a geometric path in workspace which can be expressed in the parametric form of the equations (1) to (3), where s is defined as a function of time t, ($s = s(t)$), and we define the different elements of the geometric path as $p_x = p_x(s)$,$p_y = p_y(s)$, and $p_z = p_z(s)$. Polynomial function of order three with initial condition of position and velocity is used for p_x, and p_y. Also, We used equation (3) for p_z. Distributing the time smoothly with a 3rd order polynomial starting from initial time to final time, together with the above mentioned equations, the generated trajectories reduce the probability of the collision of the moving object with the adjacent objects. In order to generate the rotation angle trajectory for the robot's hand the trapezoidal profile is used together with the extracted θ_{pick} and θ_{place} from the equations (6) and (7).

$$p_x = a_3 s^3 + a_2 s^2 + a_1 s + a_0 \tag{8}$$

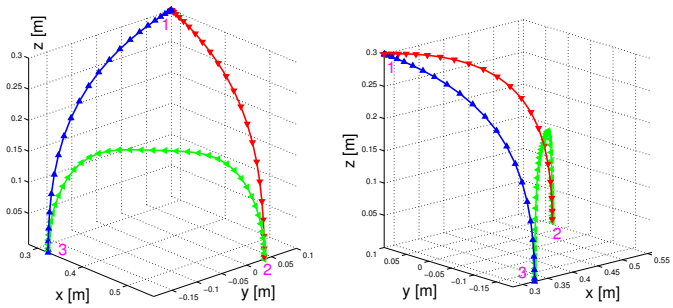
$$p_y = b_3 s^3 + b_2 s^2 + b_1 s + b_0 \tag{9}$$

$$p_z = h[1 - |(\tanh^{-1}(h_0(s - 0.5)))^\kappa|] \tag{10}$$

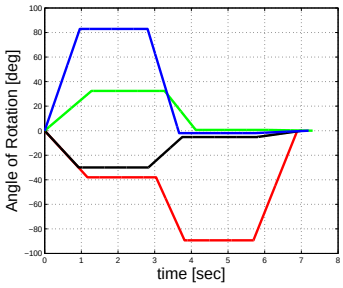
V. Experimental Results

In this section we demonstrate four experiments to illustrate the capabilities and the limitations of our approach. We show five different capabilities of VSL, which are summarized in Table I, including absolute and relative positioning, user intervention to modify the reproduction, classification, and turn-taking interaction. The video accompanying this paper shows the execution of the tasks and is available online at [20]. In all the real-world experiments the demonstration, learning, and reproduction phases are performed online, but during the demonstration phase the caregiver should take special care to avoid obstruction. In all of the experiments, in the demonstration phase, we set the size of the *frame* for the *pre-pick observation* equal to the size of the biggest object in the *world*, and the size of the *frame* for the *pre-place observation* 2 to 3 times bigger than the size of the biggest objects in the *world*. In the reproduction phase, on the other hand, we set the size of the *frame* equal to the size of the *world*.

The camera is mounted above the table and looks down on the workspace. The resolution of the captured images are



(a) The generated trajectory from two viewpoints



(b) The generated angle of rotation, θ, for the robot's hand

Fig. 5. (a) The generated trajectory for the first pick-and-place operation of the alphabet ordering task from two viewpoints. Point 1: rest-point, Point 2: pick-point, and Point 3: place-point. (b) Four generated profiles of rotation angle for the robot's hand used in different tasks.

TABLE I. CAPABILITIES OF VSL ILLUSTRATED IN EACH TASK

Capability / **Task**	Animal Puzzle	Alphabet Ordering	Animals vs. Machines	Domino
Relative Positioning	✓	✓	-	✓
Absolute Positioning	-	✓	-	-
Classification	-	-	✓	-
Turn-taking	-	-	✓	✓
User intervention to modify the reproduction	-	-	✓	✓

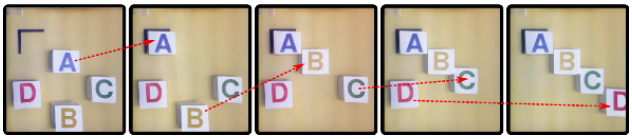
1280×960 pixels.

Although the trajectory is 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 spatial orientation of the robot's hand (end-effector) perpendicular to the workspace, in order to facilitate the pick-and-place operation. The robot's hand still can rotate parallel to the workspace plane for grasping the objects from different angles.

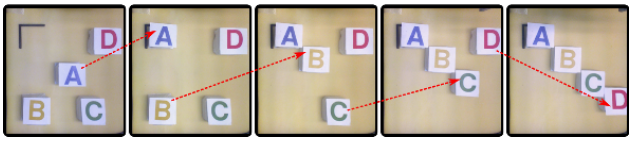
Table II lists the execution time for different steps of the implementation of the VSL approach.

A. Alphabet Ordering

In this VSL task, the *world* includes four cubic objects with A, B, C, and D labels in addition to a fixed right angle baseline which is a static part of the *world*. The goal is to assemble the set of objects with respect to 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* (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 positioning of objects without defining any explicit *a priori* knowledge. Fig. 6(a) shows the sequence of operations in the demonstration phase. Recording *pre-pick* and *pre-place observations*, the robot learns the sequence of operations. Fig. 6(b) shows the sequence of operations reproduced by VSL on a novel *world* which is achieved by randomizing the objects in the *world*.



(a) The sequence of the operations by the caregiver



(b) The sequence of the operations by the robot

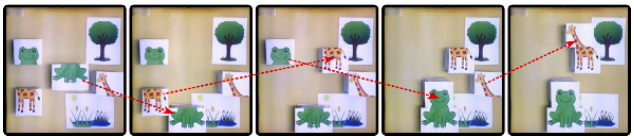
Fig. 6. Alphabet ordering (Details in Section V-A). The initial configuration of the objects in the *world* is different in (a) and (b). The red arrows show the operations.

B. Animal Puzzle

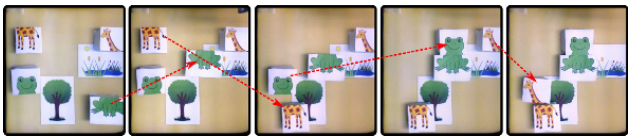
In the previous task, duo to the absolute positioning capability of VSL, the final configuration of the objects in the reproduction and the demonstration phases are always the same. In this experiment, however, the final result can be a entirely new configuration of objects by removing the fixed baseline from the *world*. The goal of this experiment is to show the VSL's capability of relative positioning. In this VSL task, the *world* includes two sets of objects which complete a 'frog' puzzle beside a 'pond' label together with a 'giraffe' puzzle beside a 'tree' label. Fig. 7(a) shows the sequence of operations in the demonstration phase. To show the capability of generalization, the 'tree' and the 'pond' are randomly replaced by the caregiver before the reproduction phase. The goal is to assemble the set of objects for each animal with

TABLE II. EXECUTION TIME FOR DIFFERENT STEPS OF VSL

Steps	**Time (sec)**	**For each**
Homography	0.2-0.35	learning task
SIFT+RANSAC	18-26	operation (reproduction)
Image subtraction	0.09-0.11	comparison
Image thresholding	0.13-0.16	comparison
Trajectory generation	1.3-1.9	operation (reproduction)
Trajectory tracking	6.6-15	operation (reproduction)
Image rectification	0.3-0.6	image
Demonstration (calculation time)	3.8-5.6	operation
Reproduction	27.5-44.1	operation



(a) The sequence of the operations by the caregiver



(b) The sequence of the operations by the robot

Fig. 7. Animal puzzle (Details in Section V-B). The initial and final configurations of the objects in the *world* are different in (a) and (b). The red arrows show the operations.

respect to the labels according to the demonstration. Fig. 7(b) shows the sequence of operations reproduced by VSL. This is one of the explicit merits of this approach that without any changes in the implementation it can deal with two different tasks.

C. Animals vs. 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 caregiver randomly picks objects one by one and places them in the corresponding bins. In the reproduction phase, the caregiver places one of the objects each time in a different sequence with respect to the demonstration. This is an interactive task between the human and the robot. The human caregiver can modify the sequence of operations in the reproduction phase by presenting the objects to the robot in a different sequence with respect to the demonstration.
To achieve the mentioned capabilities the algorithm is modified so that 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 the reproduction phase as before.
Fig. 8-1 and Fig. 8-2 show one operation during the demonstration phase in which the caregiver is classifying an object as an 'Animal'. To show that the VSL approach is rotation-invariant, in the reproduction phase the caregiver places each object in a different place with a different rotation. Fig. 8-3 and Fig. 8-4 show two reproduced operations by VSL.

D. Domino: A Turn-taking Task

The goal of this experiment is to show that VSL can 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 divided into two square ends and each end is labeled with a half object. (Due to lack of space in the workspace, one of the domino objects is cut in half). In this task first the caregiver demonstrates all the operations. The 3rd and 5th operations during the demonstration phase

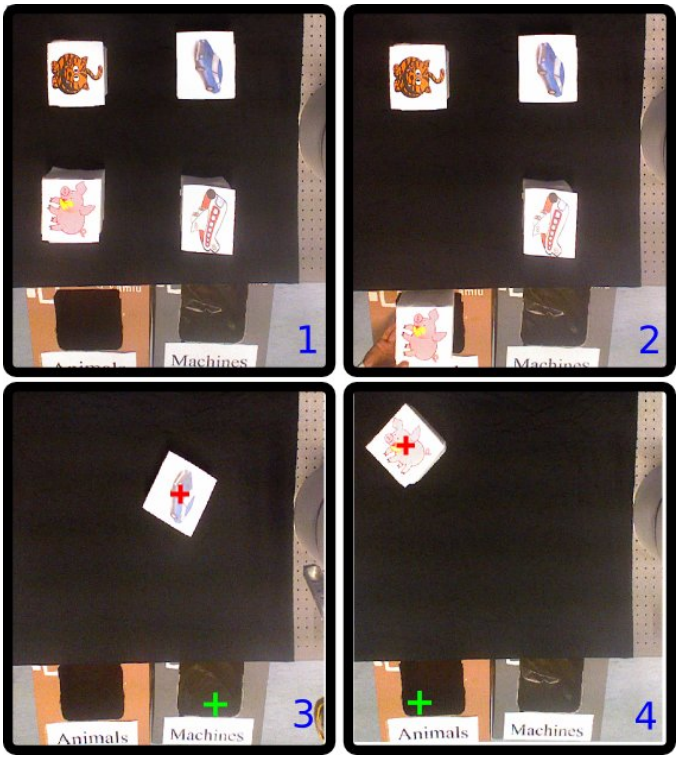


Fig. 8. Classifying of one object by the caregiver in the third task is shown in subfigures 1 and 2. Two sets of match detection results in the reproduction phase are shown in subfigures 3 and 4. The red and green crosses on the objects and on the bins, show the detected positions for pick and place actions respectively. (details in section V-C).

are shown in Fig. 9-1 and Fig. 9-2. In the reproduction phase, the caregiver starts the game by placing the first object (or another) in a random place. The robot then takes the turn and finds and places the next matching domino piece. In this task we use the modified algorithm from the previous task. The human caregiver can also modify the sequence of operations in the reproduction phase by presenting the objects to the robot in a different sequence with respect to the demonstration.
Fig. 9-3 and Fig. 9-4 show the match finding result by the VSL and the final reproduced operation by the robot respectively.

VI. Discussion

In order to test the repeatability of our approach and to identify the possible factors of failure, we used the captured images from the real world experiments while excluding the robot from the loop. We kept all other parts of the loop intact and repeated each experiment three times. The result shows that 2 out of 48 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 our algorithm is scale-invariant, color-invariant, and view-invariant, it has some limitations. For instance, if the caregiver 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

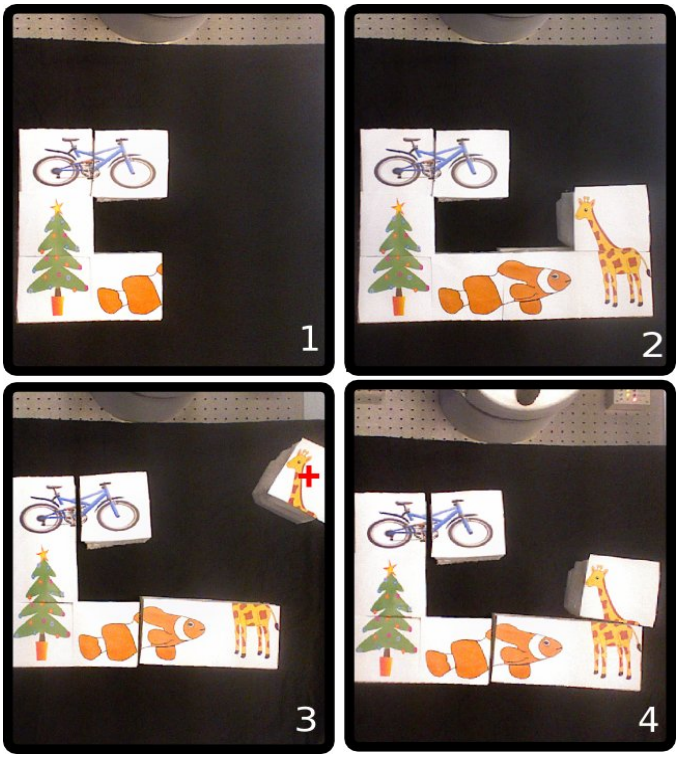


Fig. 9. Two operations during the demonstration phase by the caregiver in the domino task are shown in subfigures 1 and 2. Match detection result and the reproduced operation by the robot in the reproduction phase are shown in subfigures 3 and 4. The red cross on the object shows the detected positions for pick action by VSL. (details in section V-D).

the image subtraction and thresholding techniques to detect multi-object movements.

VII. Conclusion And Future Work

We proposed a visuospatial skill learning approach that has powerful capabilities as shown in the four real world experiments. The method possesses the following capabilities: relative and absolute positioning, user intervention to modify the reproduction, classification and turn-taking. These characteristics make VSL approach a suitable choice for interactive robot learning tasks which rely on visual perception. Moreover, our approach is convenient for the vision-based robotic platforms which are designed to perform a variety of repetitive and interactive production tasks (e.g. Baxter). Because applying our approach to such robots, requires no complex programming or costly integration.

Further perspectives include using a camera (e.g. Kinect or a stereo camera) which is robust to changes in illumination conditions and can provide depth information to perform assembly tasks along z-axis as well (e.g. stacking objects on top of each other). Finally, in our future work, we aim to improve the grasping technique for objects with different size and material. However, in this paper we applied a simple grasping method by measuring the closing torque of the Barrett Hand.

Acknowledgment

This research was partially supported by the PANDORA EU FP7 project under the grant agreement no. ICT-288273.

References

[1] B. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.

[3] B. Burdea and H. Wolfson, "Solving jigsaw puzzles by a robot," *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 6, pp. 752–764, 1989.

[4] D. Verma and R. Rao, "Goal-based imitation as probabilistic inference over graphical models," *Advances in neural information processing systems*, vol. 18, p. 1393, 2006.

[5] S. Niekum, S. Osentoski, G. Konidaris, and A. Barto, "Learning and generalization of complex tasks from unstructured demonstrations," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[6] N. Dantam, I. Essa, and M. Stilman, "Linguistic transfer of human assembly tasks to robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[7] C. Chao, M. Cakmak, and A. Thomaz, "Towards grounding concepts for transfer in goal learning from demonstration," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2. IEEE, 2011, pp. 1–6.

[8] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *Robotics and Automation, IEEE Transactions on*, vol. 10, no. 6, pp. 799–822, 1994.

[9] M. Lopes and J. Santos-Victor, "Visual learning by imitation with motor representations," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 35, no. 3, pp. 438–449, 2005.

[10] A. K. Pandey and R. Alami, "Towards task understanding through multi-state visuo-spatial perspective taking for human-robot interaction," in *IJCAI workshop on agents learning interactively from human teachers (ALIHT-IJCAI)*, 2011.

[11] S. R. Ahmadzadeh, P. Kormushev, and D. G. Caldwell, "Visuospatial skill learning for object reconfiguration tasks," in *Intelligent Robots and Systems (IROS) 2013, IEEE/RSJ International Conference on*. IEEE, 2013.

[12] A. Meltzoff and M. Moore, "Newborn infants imitate adult facial gestures," *Child development*, pp. 702–709, 1983.

[13] G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi, "Premotor cortex and the recognition of motor actions," *Cognitive brain research*, vol. 3, no. 2, pp. 131–141, 1996.

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

[15] M. Ehrenmann, O. Rogalla, R. Zöllner, and R. Dillmann, "Teaching service robots complex tasks: Programming by demonstration for workshop and household environments," in *Proceedings of the 2001 International Conference on Field and Service Robots (FSR)*, vol. 1, 2001, pp. 397–402.

[16] M. Yeasin and S. Chaudhuri, "Toward automatic robot programming: learning human skill from visual data," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 30, no. 1, pp. 180–185, 2000.

[17] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge Univ Press, 2000, vol. 2.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[19] T. Y. Wong, P. Kovesi, and A. Datta, "Projective transformations for image transition animations," in *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*. IEEE, 2007, pp. 493–500.

[20] Video, "Video accompanying this paper, available online," http://kormushev.com/goto/ICAR-2013, 2013.