Iterative Path Optimisation for Personalised Dressing Assistance using Vision and Force Information

Conference Paper · October 2016

DOI: 10.1109/IROS.2016.7759647

3 authors:

Yixing Gao
Imperial College London

Hyung Jin Chang
Imperial College London

Yiannis Demiris
Imperial College London

Preprint version; final version available at ieeexplore.ieee.org
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2016)
Published by: IEEE

Iterative Path Optimisation for Personalised Dressing Assistance using Vision and Force Information

Yixing Gao Hyung Jin Chang Yiannis Demiris

***Abstract*— We propose an online iterative path optimisation method to enable a Baxter humanoid robot to assist human users to dress. The robot searches for the optimal personalised dressing path using vision and force sensor information: vision information is used to recognise the human pose and model the movement space of upper-body joints; force sensor information is used for the robot to detect external force resistance and to locally adjust its motion. We propose a new stochastic path optimisation method based on adaptive moment estimation. We first compare the proposed method with other path optimisation algorithms on synthetic data. Experimental results show that the performance of the method achieves the smallest error with fewer iterations and less computation time. We also evaluate real-world data by enabling the Baxter robot to assist real human users with their dressing.**

I. INTRODUCTION

Assistive robots are increasingly used to provide help to elderly and disabled people in home environments [1–3]. As individuals differ in physical conditions, skill sets and habits, being able to provide personalised assistance becomes a crucial capability for such robots. Assistive dressing is a useful and common daily task for humans; in this paper, we aim to enable a humanoid robot to learn the optimal personalised dressing path for an individual user.

Assistive robots can use the vision information of a human body when dressing a user [4–6]. However, occlusions could occur when the robot's arms, the clothes and the human body are in close contact, which leads to human pose recognition failures. Thus, other sensor information about humans should be introduced to compensate for the disadvantages of using vision information only [7]. Additionally, due to varying dressing habits, different people may behave differently when wearing clothes. For instance, their arm movement paths during dressing may not be the same. To increase human comfort, assistive robots should be able to find the preferred arm movement path of a person and then perform the dressing assistance following the preferred path.

In this paper, we present an online iterative update method for searching the optimal path in space. We apply this method to enable a Baxter humanoid robot to learn the optimal personalised path for dressing a user using vision and force sensor information. We show the general idea in Figure 1. As in [4], a top-view depth camera is used to recognise human upper-body pose and to collect motion information of human upper-body joints. The movement space of human hands is modelled with Gaussian mixture models (GMMs) for the robot to choose starting dressing positions for the user. Initial human elbow and shoulder positions are detected with the camera. The robot starts dressing a user following the initial dressing path and iteratively updates this path with detected external force information until it finds the optimised dressing path.

The authors are with the Personal Robotics Lab, Department of Electrical and Electronic Engineering, Imperial College London, United Kingdom {y.gao12,hj.chang,y.demiris}@imperial.ac.uk
This work was supported in part by the EU FP7 project WYSIWYD under Grant 612139.

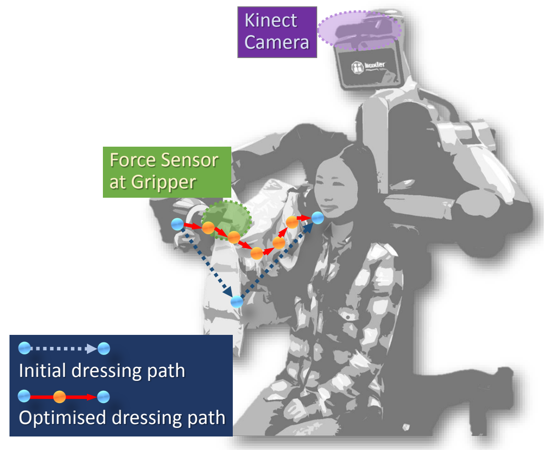


Fig. 1: The proposed iterative path optimisation method enables a Baxter humanoid robot to search for the optimal personalised dressing path for a real human user. Vision information (purple) is used to model the movement space of the human's hands and to decide the initial dressing path (blue circles). Force sensor information (green) is used by the robot to detect external force resistance in order to locally adjust its motion. The robot iteratively updates the current dressing path until finding the optimised one (red path connecting orange circles). Best viewed in colour.

II. RELATED WORK

There has been interesting prior research on assistive dressing by humanoid robots in home environments. Tamei *et al.* [8] proposed to use the reinforcement learning method to teach a dual-arm robot to learn the dressing motion. For a human mannequin with different head and shoulder inclinations, the robot learned how to dress it with a T-shirt. In the experiments, the topological relationships between the cloth and the mannequin were observed by the robot to optimise its joints trajectories. To improve the real-time estimation of human-cloth relationships, Koganti *et al.* [9] proposed the offline learning of a cloth dynamics model with different sensor data and applied this learned

Preprint version; final version available at ieeexplore.ieee.org
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2016)
Published by: IEEE

model to track human-cloth relationships online using a depth sensor. In [4], Gao *et al.* modelled the movement space of human upper-body joints using GMMs and enabled a Baxter robot to plan the dressing motion according to the user models and real-time human pose. The order of dressing sequences, which were hand, forearm, upper arm and shoulder, remained the same for all users. Considering individual user's comfort, assistive robots should be able to adjust or update the dressing sequences during human-robot interactions. Besides, human arms could be occluded by the robot's arms and the clothes during dressing, which caused failures in human pose recognition.

Apart from vision information, force sensors also provide useful feedback during human-robot interactions and they have been widely used along with vision in various applications. Rozo *et al.* [10] taught a robot arm to cooperatively transport an object with a human through demonstrations. Force sensors were attached at the end of the robot arm and the robot learned how to perform a new task by fulfilling both the force and position constraints. Kruse *et al.* [11] presented a feedback controller using both force and vision information to enable a Baxter robot to collaboratively unfold a piece of cloth with a human by responding to the force and vision changes. For the assistive dressing purpose, we aim to use real-time force sensor feedback for the assistive robot to iteratively update the dressing path, which requires a path optimisation process.

There are a number of robotic research in stochastic path optimisation, such as robot motion planning [12,13] and path planning [14]. Hegels *et al.* [15] proposed an approach to post-optimisation using the nonlinear conjugate gradient method and they applied their iterative path optimisation algorithm in real-world applications of robot-guided thermal spraying. The optimal path was planned offline with a simulation tool and real-time feedback was not taken into consideration. Stochastic optimisation recently becomes an active area of research because of the popularity of deep networks. To perform a parameter update with the gradients, there has been some well-established methods. Vanilla update which uses stochastic gradient descent (SGD) [16] is the simplest form, but it may not converge or it converges slowly when the learning rate is small enough. With momentum update [17], the parameter vector builds up velocity in any direction which has the consistent gradient, thus the convergence rate is faster. Duchi *et al.* [18] proposed an adaptive learning rate method AdaGrad, which performed well with sparse gradients. Tieleman & Hinton [19] presented RMSProp which adjusted the AdaGrad method by using a moving average of squared gradients. The recently proposed Adam method [20] combines the advantages of the two popular methods AdaGrad and RMSProp, which has shown its robustness to a variety of non-convex optimisation problems in machine learning.

III. Iterative path optimisation

The Adam method is designed to find the global optimum for a stochastic objective function [20]. In this work, we propose an iterative path optimisation method based on Adam which can search for the optimal path starting with an initial path. For assistive dressing with a humanoid robot, the starting and ending positions of the optimised path are set from the vision sensor. We use detected force information to guide the search process. Specifically, when the current path is far away from the optimal path, external force resistance will be detected by the robot. The robot should use the force information to locally adjust its motion and iteratively find the optimal dressing path. How to detect external force resistance during human-robot interactions will be described in detail in section IV-B.

Algorithm 1: Iterative path optimisation

Input : initial path $\mathcal{W}^0$
Output: optimised path $\tilde{\mathcal{W}}$
Initialisation m^t,v^t,$t \leftarrow 0$
while $t < t_{max}$ *or* $\mathcal{E}_{energy} > \tau_{energy}$ **do**
    $t \leftarrow t+1$
    $\mathcal{E}_{energy} \leftarrow 0$
    for *all* P_i *in* $\mathcal{W}^{t-1}$ **do**
        UpdatePath(P_i,P_{i+1},m^{t-1},v^{t-1},t,$\mathcal{W}^t$,$\mathcal{E}_{energy}$)
    $m^t \leftarrow$ get average of all $m^t(n)$
    $v^t \leftarrow$ get average of all $v^t(n)$
Function
UpdatePath(P_{start},P_{end},m^{t-1},v^{t-1},t,$\mathcal{W}^t$,$\mathcal{E}_{energy}$) **is**
    Generate path p from P_{start} to P_{end} using motion planning library [21]
    for *each* n^{th} *path point* $p(n)$ **do**
        Detect $g(n)$
        if $g(n) > \tau_g$ **then**
            $m^t(n) \leftarrow \beta_1 \cdot m^{t-1} + (1-\beta_1) \cdot g(n)$
            $v^t(n) \leftarrow \beta_2 \cdot v^{t-1} + (1-\beta_2) \cdot (g(n))^2$
            $\hat{m}^t(n) \leftarrow m^t(n)/(1-(\beta_1)^t)$
            $\hat{v}^t(n) \leftarrow v^t(n)/(1-(\beta_2)^t)$
            $p(n) \leftarrow p(n) - \alpha \cdot \hat{m}^t(n)/(\sqrt{\hat{v}^t(n)} + \epsilon)$
            Add updated $p(n)$ to $\mathcal{W}^t$
            $\mathcal{E}_{energy} \leftarrow \mathcal{E}_{energy} + g(n)$
            $P_{end} \leftarrow P_{end} - \alpha \cdot \hat{m}^t(n)/(\sqrt{\hat{v}^t(n)} + \epsilon)$
            UpdatePath($p(n)$,P_{end},m^{t-1},v^{t-1},t,$\mathcal{W}^t$,$\mathcal{E}_{energy}$)
        end
    end
    Add P_{end} to $\mathcal{W}^t$
end

The proposed method is described in Algorithm 1. With an initial path, we keep updating until it converges to the optimal path. We define a path after the t^{th} iteration as $\mathcal{W}^t = \{P_1, ..., P_{i-1}, P_i, P_{i+1}, ..., P_N\}$, where $i \in \{1, ..., N\}$, $P_i = \{x_i, y_i, z_i\}$. We represent the initial path as $\mathcal{W}^0$ and the final optimised path as $\tilde{\mathcal{W}} = \{P_1, ..., P_N\}$. In assistive dressing, P_i of $\mathcal{W}^t$ is one of the main goal positions for the robot gripper.

When we start the current iteration, we first update the counter t, thus $\mathcal{W}^{t-1}$ represents the current path after the last

Preprint version; final version available at ieeexplore.ieee.org
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2016)
Published by: IEEE

iteration. We use $\mathcal{E}_{energy}$ to represent the energy which is the total amount of detected external force resistance in assistive dressing. For all the path waypoints P_i in $\mathcal{W}^{t-1}$, function *UpdatePath*(P_i,P_{i+1},m^{t-1},v^{t-1},t,$\mathcal{W}^t$,$\mathcal{E}_{energy}$) is called to generate $\mathcal{W}^t$. Inside the function *UpdatePath*, P_i is passed to P_{start} and P_{i+1} to P_{end}. P_{start} is the current starting position for the robot gripper and P_{end} is the initial goal position. The gripper path p is planned from P_{start} to P_{end} using the motion planning library [21].

For each n^{th} path waypoint $p(n)$, we check $g(n)$ which is the detected external force. If $g(n)$ is larger than the threshold τ_g, it means that external resistance is detected. In Adam, $g(n)$ denotes the gradients with respect to the stochastic objective at the current timestep. In our algorithm, $g(n)$ is the force information where its directions and values guide the current path towards the optimal path. If $g(n) > \tau_g$, we calculate $m^t(n)$, $v^t(n)$, $\hat{m}^t(n)$, $\hat{v}^t(n)$ and update the current path waypoint $p(n)$ following the Adam method. $m^t(n)$ and $v^t(n)$ are the biased first and second moment estimates of $g(n)$; $\hat{m}^t(n)$ and $\hat{v}^t(n)$ are the bias-corrected first and second moment estimates; β_1 and β_2 are the exponential decay rates for the moment estimates; α represents the learning rate and ϵ is the smoothing term. The updated $p(n)$ represents how the gripper locally adjusts its position based on the force information. The robot stops the current execution and moves the gripper to the updated position $p(n)$. This $p(n)$ is added to $\mathcal{W}^t$ and taken as the new starting position for the gripper. The reason to add $p(n)$ to $\mathcal{W}^t$ is because we expect that the same force resistance could be avoided in the next iteration by letting the gripper move towards the updated $p(n)$ directly instead of following the previous path.

Since external resistance is detected, $\mathcal{E}_{energy}$ is updated with $g(n)$ and the initial goal position P_{end} is updated following the same update rule as $p(n)$. The reason to update P_{end} is because the goal position has changed after external resistance is detected. Then the function calls itself again with the new starting position $p(n)$ and goal position P_{end}. The final updated P_{end} is added to $\mathcal{W}^t$. In another condition, if the $g(n)$ of each $p(n)$ in the planned path is smaller than τ_g, then the original P_{end} is directly added to $\mathcal{W}^t$. For the Adam in [20], $m^t(n)$, $v^t(n)$, $\hat{m}^t(n)$, and $\hat{v}^t(n)$ are calculated only once within each iteration to update parameters. However, when our goal is to search for an optimal path, $m^t(n)$, $v^t(n)$, $\hat{m}^t(n)$, and $\hat{v}^t(n)$ are calculated multiple times for different waypoints. In the proposed method, after we finish checking all the P_i in $\mathcal{W}^{t-1}$, we update m^t and v^t by calculating the mean value of all the $m^t(n)$ and $v^t(n)$ within this iteration and the updated m^t and v^t will be used as m^{t-1} and v^{t-1} in the next iteration.

There are two terminating conditions for the whole iteration process. The first condition is when the total number of iterations exceed the maximum iterations t_{max} and the second condition is when the energy $\mathcal{E}_{energy}$ is smaller than the energy threshold τ_{energy}. According to [20], good default settings for the Adam parameters are $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

IV. Personalised Assistive Dressing

We apply the proposed method in solving real-world applications for home-environment assistive robots. Our goal is for the robot to iteratively find the optimal personalised dressing path for a person using vision and force sensor information. We use a Baxter humanoid robot and a sleeveless jacket for the dressing assistance.

A. *Movement space modelling of human hand*

For dressing assistance, it is significant for assistive robots to know the reachable area of human hands. For instance, it would be ineffective if an assistive robot selects a starting dressing position which cannot be reached by the user. Thus the movement space of human hands should be studied before assistive dressing.

For modelling the movement space of human hands, Gao *et al.* [4] proposed a method using GMMs. With a top-view depth camera, the human upper-body pose can be recognised from a single depth image using randomised decision forests [22]. GMMs is used to model human joints movement and parameters of each human joint model are estimated by an unsupervised expectation-maximisation (EM) learning algorithm [23]. In this paper, we use the estimates of the means of hands GMMs as prior knowledge for assistive robots to choose starting dressing positions.

B. *Detecting external force resistance*

During assistive dressing, due to the movement of human arms, some external force resistance can occur and it should be detected by the robot to adjust its dressing motion. The Baxter robot is equipped with force sensor at the endpoint of each limb. The force coordinates of the robot grippers are with respect to the endpoints of robot limbs. Whatever the orientations of robot limb endpoints are, the spatial relationships of frame axes between the robot coordinates and endpoint force coordinates can always be acquired using standard techniques of robot operating system (ROS) coordinate transformation. For dressing purpose, we let the Baxter robot use two grippers to grasp the shoulder parts of a sleeveless jacket and we fix the orientations of robot grippers during the whole dressing process. In our experimental set-up, the spatial relationships of frame axes between the robot coordinates and endpoint force coordinates are $(x, y, z)_{gripperForce} = -(x, y, z)_{robotCoordinate}$. Translations between two coordinates are not considered, because we only concern the force directions in the robot coordinates.

The force value read from each force sensor is the current force which is applied to the robot's limb endpoint in each force axis. To detect external force resistance, we use a fixed size moving window to calculate force difference at each time step. We represent the combined force of a robot limb endpoint at time step t as $f^t = \sqrt{(f_x^t)^2 + (f_y^t)^2 + (f_z^t)^2}$, where f_x^t, f_y^t and f_z^t can be directly read. At time step $t+1$, the combined force difference is $\Delta f^{t+1} = f^{t+1} - f^t$ and the force difference in each axis is $\Delta f_x^{t+1} = f_x^{t+1} - f_x^t$, $\Delta f_y^{t+1} = f_y^{t+1} - f_y^t$, $\Delta f_z^{t+1} = f_z^{t+1} - f_z^t$. We use N

Preprint version; final version available at ieeexplore.ieee.org
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2016)
Published by: IEEE

to represent the time steps for the moving window. Within the moving window, the sum of combined force difference is $F = \sum_{t=t_i}^{t_{i+N}} \Delta f^t$ and the sum of force difference in each axis is $F_x = \sum_{t=t_i}^{t_{i+N}} \Delta f_x^t$, $F_y = \sum_{t=t_i}^{t_{i+N}} \Delta f_y^t$ and $F_z = \sum_{t=t_i}^{t_{i+N}} \Delta f_z^t$. During robot execution of an action, without external disturbance Δf^{t+1} remains a small positive or negative value at each time step, therefore $|F|$ should be within a force range. If there is external force resistance, Δf^{t+1} will keep being positive or negative within a short time period, thus there will be a quick increase in $|F|$. If $|F| > \tau_f$, where τ_f is defined as a force threshold, it means that external force disturbance is detected. In real assistive dressing experiments, this τ_f is set to 5N and N is set to 15. Each of the robot's gripper force is read at 100Hz.

C. Learning optimal personalised dressing path

For different people, their arms may behave differently while putting on clothes. However, an individual usually tends to follow certain behaviour pattern for daily activities. Our target is to let the assistive robot learn the optimal personalised dressing path for a user which consists of a sequence of endpoint positions for robot grippers.

With a sleeveless jacket, the robot can choose to wear first either the left part or the right part of the jacket for a human and then wear the opposite part. In this work, we let the robot assist a human to wear the right part of a jacket and then the left part. As the robot uses two grippers to hold the shoulder parts of the jacket, the robot gripper positions also represent the jacket shoulder positions. The robot first chooses the starting dressing position for a user according to the movement space model of user right hand (section IV-A), then the user moves the right hand to this starting dressing position. The starting dressing position also becomes the starting position of the human right hand P_{hand}. Starting positions for the human's right elbow P_{elbow} and shoulder $P_{shoulder}$ can then be detected with the top-view depth camera. The initial dressing path for the robot right gripper is defined as $\mathcal{W}^0 = \{P_{hand}, P_{elbow}, P_{shoulder}\}$. Our goal is for the robot to iteratively update the dressing path by detecting external force resistance (section IV-B) and locally adjusting its motion until finding the optimised dressing path $\mathcal{W}^* = \{P_{hand}, P_1^*, ..., P_{N^*}^*, P_{shoulder}\}$. For the optimal path $\mathcal{W}^*$, P_{hand} and $P_{shoulder}$ remain the same as in the initial path $\mathcal{W}^0$. This is because P_{hand} is chosen according to the movement space model of the human hand. For $P_{shoulder}$, although the human arm can move during dressing, the position of human shoulder is not affected. Therefore, only middle waypoints positions need to be updated during assistive dressing. This update process works similarly for the robot left gripper.

In section III, we check $g(n)$ for each path waypoint. In assistive dressing, we let $g(n) = F$ where F represents the combined force difference, which is illustrated in section IV-B. For the update of path waypoint, the adjustment of position is calculated in x, y and z axis using F_x, F_y and F_z respectively. We let $\tau_g = \tau_f$ to represent the force threshold. τ_{energy} is set to 0, which means that when no external force resistance is detected after the current path update, the robot thinks that the user feels comfortable with this path and will stop the iterations. The final updated path is then taken as the optimal personalised dressing path. Different with an optimisation process in simulation which can run a large number of times, we set the maximum iterations t_{max} to 8 since the robot is expected to find the optimal path for a person quickly.

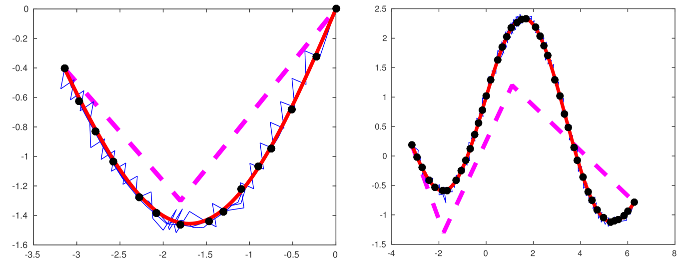


Fig. 2: This figure shows 2 examples of iteration process with synthetic data. In each example, the initial path is in the magenta line and the ground-truth optimal path is in the red line. The updated paths after each iteration are in blue lines and the waypoints of the final optimised path are indicated with black dots. Best viewed in colour.

V. Experiments

A. Synthetic dataset

We first evaluate the proposed method with $2D$ synthetic data by randomly generating 100 pairs of ground-truth optimal paths and initial paths. We cannot run the assistive dressing experiments in simulation since real-time force interaction data is required. In real assistive dressing, the detected force information for the current path waypoint is proportional to the distance from this waypoint to the optimal path. Thus we can use the distance to simulate the force information. With synthetic data, we first find the closest point on the optimal path to the current path point and let $g(n)$ represent the euclidean distance. Then τ_g is used to represent the distance threshold, which is set to 0.02. With synthetic data, the path between two waypoints is planned using linear regression, where the step length is set to 0.05. The energy threshold τ_{energy} is set to 0.05 and the maximum iterations t_{max} is set to 40.

We compare the proposed method with methods using vanilla SGD update [16], momentum update [17], Adagrad [18] and RMSProp [19] by running 100 experiments with each method. The learning rate α in each method is set to 0.1. The momentum hyperparameter of momentum update is set to its typical value 0.9; the decay rate hyperparameter of RMSProp is set to its typical value 0.99. Figure 2 shows 2 examples of iteration process with the proposed method. In each example, the magenta line represents the initial path and the red line represents the ground-truth optimal path. The blue lines represent the paths after each iteration and the black dots are the path points on the final optimised path.

Preprint version; final version available at ieeexplore.ieee.org
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2016)
Published by: IEEE

TABLE I: Comparison results with synthetic data

Method	Error	Iteration number	Computation time (s)
SGD [16]	12.23 ($\pm$4.79)	35.51 ($\pm$3.01)	1143.58 ($\pm$781.94)
Momentum [17]	7.23 ($\pm$3.49)	30.89 ($\pm$5.65)	1118.02 ($\pm$724.63)
Adagrad [18]	4.99 ($\pm$2.34)	17.17 ($\pm$4.54)	609.83 ($\pm$402.64)
RMSProp [19]	3.27 ($\pm$1.28)	13.42 ($\pm$3.25)	554.62 ($\pm$297.35)
Proposed	**2.08** ($\pm$0.93)	**7.99** ($\pm$2.18)	**377.90** ($\pm$253.71)

TABLE II: Assistive dressing experiments with real users

User	Iteration number	Computation time (s)	Energy (N)
No.1	2.6 ($\pm$1.34)	57.32 ($\pm$50.69)	25.15 ($\pm$32.80)
No.2	2.4 ($\pm$0.89)	57.79 ($\pm$24.09)	25.94 ($\pm$11.06)
No.3	2.4 ($\pm$0.55)	143.04 ($\pm$44.12)	56.24 ($\pm$43.36)
No.4	2.4 ($\pm$0.55)	60.31 ($\pm$19.40)	20.79 ($\pm$8.93)
No.5	2.4 ($\pm$0.89)	47.78 ($\pm$23.90)	18.43 ($\pm$15.60)
No.6	2.6 ($\pm$0.55)	48.35 ($\pm$18.52)	21.31 ($\pm$12.45)

In each experiment, we calculate the error ε between the final updated path $\mathcal{W}_{final} = \{[x_1, y_1], ..., [x_i, y_i], ..., [x_m, y_m]\}$ and its corresponding ground-truth optimal path. We use d_i to represent the distance from $[x_i, y_i]$ to the optimal path and the error is defined as $\varepsilon = \sum_{i=1}^{i=m} d_i$. For each experiment, we record the iteration number and computation time. For each method, we calculate the mean and the standard deviation of the error, iteration number and computation time among 100 experiments. The experiment results are shown in Table I. It can be seen that the proposed method achieves the smallest average error, iteration number and computation time comparing with the other four methods. Besides, the proposed method also achieves the smallest standard deviations for the error, iteration number and computation time. Experimental results with synthetic data show that the proposed method can iteratively update a path online and converge within a smaller times of iterations.

B. *Real-world personalised assistive dressing*

We evaluate the proposed method by enabling the Baxter robot to find the optimal personalised dressing paths for human users. Six healthy participants (four female) ages 27-32 (mean: 28, std: 2.19) participated in the experiments.

We run five experiments for each participant. We let the robot assist the human to wear the right part of the jacket first followed by the left part. In each experiment, the robot keeps updating the dressing path for a user by detecting external force resistance and adjusting path points with the proposed method.

For each user, we record the total iteration number, computation time and energy in each experiment. The energy indicates the total amount of detected external force resistance, which is described by $\mathcal{E}_{energy}$ in section III. For experiments with synthetic dataset, the error can be calculated between the final updated path and the ground-truth path. However, the ground-truth dressing path of a user is not known before in real-world assistive dressing applications. Thus, we show the results of the energy instead of the error.

Experiment results are shown in Table II. For all the participants, the robot found the optimal dressing paths within a maximum of five iterations. The average iteration number for all users were similar. Apart from the 3^{rd} user, the robot averagely spent around 1 minute to finish the path update in one experiment. For the 3^{rd} user, the robot averagely spent around 2 minutes in each experiment and the average detected energy was about twice than the others. The more external force resistance was detected, the more time the robot spent to locally adjust its motion. Thus the average computation time for the 3^{rd} user was larger.

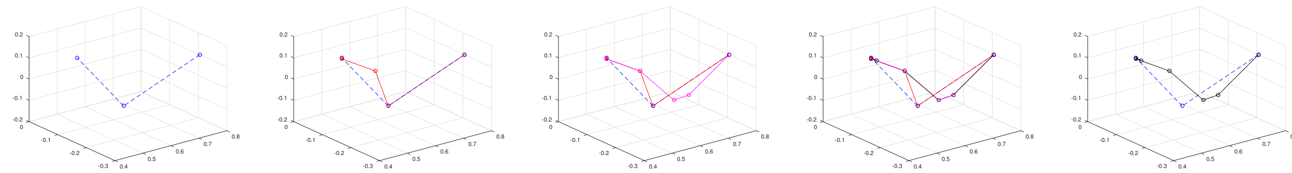
We show one of the path iteration process for wearing the right arm of one participant in Figure 3. Figure 3a shows the initial path in a blue dotted line where the initial path waypoints are indicated by blue circles. Figure 3b shows the updated path in a red line with new path waypoints indicated by red circles after 1^{st} iteration. It can be seen that one new path waypoint is added in the updated path. The energy of this iteration is $20.2N$. Figure 3c shows the updated path in a magenta line with new path waypoints indicated by magenta circles after 2^{nd} iteration. Comparing with the path after 1^{st} iteration, one old path waypoint is updated and a new path point is added. The energy of this iteration is $11.6N$. Figure 3d shows the updated path in a black line with new path waypoints indicated by black circles. Comparing with the path after 2^{nd} iteration, one new path waypoint is added. Since there is only a small change in the updated path, the energy is $5.7N$ in this iteration which is smaller than the energy in the last iteration. Figure 3e shows the final updated path in a black line comparing with the initial path. In the 4^{th} iteration, the robot assists the user to wear the right arm following the updated path after the 3^{rd} iteration. Since there is no external force resistance detected, the final energy is $0N$ and the robot thinks that the user feels comfortable with the current path. Thus the updated path after the 3^{rd} iteration becomes the final updated path after the 4^{th} iteration. Some screenshots of assistive dressing are shown in Figure 4.

In the real-world experiments, we use a fixed size moving window to detect external force resistance by using the force sensor information from the endpoints of robot limb. However, the robot grippers which grasp the jacket have no force sensors on them. Although we try to minimise the detected force noise by calculating the sum of force difference within a sliding window, some environment noise still cannot be avoided. If the robot grippers were attached with force sensor, the robot could react more sensitively to the external force disturbance.

VI. Conclusions

In this work we propose an online iterative path optimisation method for searching the optimal path. We enable a Baxter robot to find the optimal personalised dressing path for different users through iteratively updating the current dressing path using vision and force information. We validate

Preprint version; final version available at ieeexplore.ieee.org
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2016)
Published by: IEEE



(a) Initial path (b) Iter.#1: energy $20.2N$ (c) Iter.#2: energy $11.6N$ (d) Iter.#3: energy $5.7N$ (e) Iter.#4: energy $0N$

Fig. 3: This figure shows how the path is updated from one of the assistive dressing experiment results of one participant. (a) shows the initial path in a blue dotted line. (b) shows the updated path in a red line. (c) shows the updated path in a magenta line. (d) shows the updated path in a black line. (e) shows the final optimised path in a black line comparing with the initial path. Best viewed in colour.

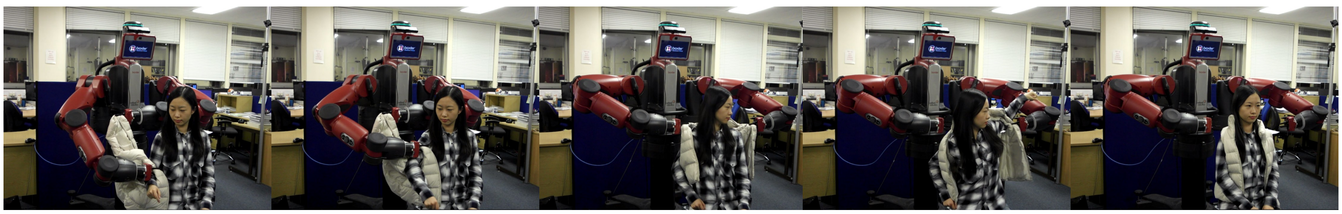


Fig. 4: Screenshots of assistive dressing by a Baxter humanoid robot. Best viewed in colour.

the method with synthetic data by comparing the proposed method with optimisation algorithms using vanilla SGD update, momentum update, Adagrad and RMSProp, and results show that the proposed method can converge quickly with much smaller error comparing with other methods. Experiment results show that the Baxter robot successfully finds the optimal dressing path for different participants with a few iterations. Feedback from the participants show that they feel safe and comfortable during the interactions with the robot. In the current work, the dynamics of the clothes, or the friction with the skin or other clothes are not considered. Future research work could study the dynamics of the clothes to be able to assist real users with more complex clothes, for example to wear a jacket with sleeves.

REFERENCES

[1] R. Bemelmans, G. J. Gelderblom, P. Jonker, and L. De Witte, "Socially assistive robots in elderly care: A systematic review into effects and effectiveness," *Journal of the American Medical Directors Association*, vol. 13, no. 2, pp. 114–120, 2012.

[2] J. Broekens, M. Heerink, and H. Rosendal, "Assistive social robots in elderly care: a review," *Gerontechnology*, vol. 8, no. 2, pp. 94–103, 2009.

[3] E. Broadbent, R. Stafford, and B. MacDonald, "Acceptance of healthcare robots for the older population: review and future directions," *International Journal of Social Robotics*, vol. 1, no. 4, pp. 319–330, 2009.

[4] Y. Gao, H. J. Chang, and Y. Demiris, "User modelling for personalised dressing assistance by humanoid robots," in *IROS*, 2015, pp. 1840–1845.

[5] S. D. Klee, B. Q. Ferreira, R. Silva, J. P. Costeira, F. S. Melo, and M. Veloso, "Personalized assistance for dressing users," in *International Conference on Social Robotics*. Springer, 2015, pp. 359–369.

[6] A. Colomé, A. Planells, and C. Torras, "A friction-model-based framework for reinforcement learning of robotic tasks in non-rigid environments," in *ICRA*, 2015, pp. 5649–5654.

[7] Y. Gao, H. J. Chang, and Y. Demiris, "Personalised assistive dressing by humanoid robots using multi-modal information," in *ICRA Workshop on Human-Robot Interfaces for Enhanced Physical Interactions*, 2016.

[8] T. Tamei, T. Matsubara, A. Rai, and T. Shibata, "Reinforcement learning of clothing assistance with a dual-arm robot," in *Humanoids*, 2011, pp. 733–738.

[9] N. Koganti, J. G. Ngeo, T. Tomoya, K. Ikeda, and T. Shibata, "Cloth dynamics modeling in latent spaces and its application to robotic clothing assistance," in *IROS*, 2015, pp. 3464–3469.

[10] L. Rozo, S. Calinon, and D. G. Caldwell, "Learning force and position constraints in human-robot cooperative transportation," in *RO-MAN*, 2014, pp. 619–624.

[11] D. Kruse, R. J. Radke, and J. T. Wen, "Collaborative human-robot manipulation of highly deformable materials," in *ICRA*, 2015, pp. 3782–3787.

[12] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.

[13] E. Masehian and D. Sedighizadeh, "Classic and heuristic approaches in robot motion planning-a chronological review," *World Academy of Science, Engineering and Technology*, vol. 23, pp. 101–106, 2007.

[14] P. Raja and S. Pugazhenthi, "Optimal path planning of mobile robots: A review," *International Journal of Physical Sciences*, vol. 7, no. 9, pp. 1314–1320, 2012.

[15] D. Hegels, T. Wiederkehr, and H. Müller, "Simulation based iterative post-optimization of paths of robot guided thermal spraying," *Robotics and Computer-Integrated Manufacturing*, vol. 35, pp. 1–15, 2015.

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Parallel Distributed Processing*, pp. 318–362, 1986.

[17] D. E. Rumelhart, G. E. Hintont, and R. J. Williams, "Learning representations by back-propagating errors," *NATURE*, vol. 323, p. 9, 1986.

[18] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[19] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop," *COURSERA: Neural networks for machine learning*, 2012.

[20] D. Kingma and J. Ba, "A method for stochastic optimization," in *International Conference on Learning Representation*, 2015.

[21] S. Chitta, I. Sucan, and S. Cousins, "MoveIt!" *IEEE Robotics Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[22] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[23] M. A. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.