Teaching Assembly Tasks to Robots Using a 3D Simulation Environment

Christos Papadopoulos | Aristotle University of Thessaloniki, Greece
Christos Katsanos | Aristotle University of Thessaloniki, Greece

CHIGreece 2023 | 27 - 28 September 2023

Introduction (1/2)

- **Assembly processes in manufacturing environments**
  - Majority is performed by robots
  - Conventional methods for programming these robots (e.g., with a teach pendant[1]) are
    - costly and time-consuming
    - inflexible to changes or additions to the assembly task
    - require specialized expertise, hardware or non-expert employees interacting with expensive robots

- **Robot Learning from Demonstration (LfD)**[2]
  - Human instructors teach the task(s) to the robot
  - Two categories of LfD methods in existing literature
    - Passive LfD
    - Active LfD

[1] Pan, Z., Polden, J., Larkin, N., Van Duin, S., & Norrish, J. (2012). Recent progress on programming methods for industrial robots. *Robot. Comput. -Integr. Manuf.*, *28*(2), 87–94
[2] Chernova, S., & Thomaz, A. L. (2014). *Robot learning from human teachers*. Springer International Publishing.

Introduction (2/2)

- **Passive LfD [1-3]**
  - User does not interact with the robot or any type of interface and the system passively observes the user through a camera (computer vision) or a unique interface (e.g., sensor gloves for tracking human motions)
  - Intuitive & simple (+), High degrees of freedom (+), Lack of precision (-), Not easy setup (-)

- **Active LfD [4-6]**
  - User physically moves the robot (kinesthetic teaching) or remotely controls the robot (teleoperation) to teach the task to the robot
  - High precision (+), Ease of setup (+), Difficulties in demonstration(-), Low degrees of freedom (-)

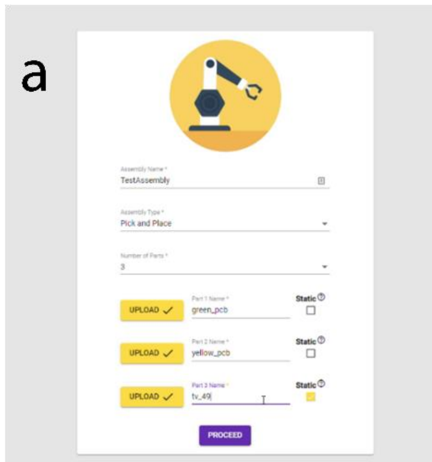
[1] Haage, M., Piperagkas, G., Papadopoulos, C., Mariolis, I., Malec, J., Bekiroglu, Y., Hedelind, M., & Tzovaras, D. (2017). Teaching assembly by demonstration using advanced human robot interaction and a knowledge integration framework. *Proc. Manuf.*, *11*, 164–173
[2] Vogt, D., Stepputtis, S., Grehl, S., Jung, B., & Ben Amor, H. (2017). A system for learning continuous human-robot interactions from human-human demonstrations. *IEEE ICRA 2017*, 2882–2889
[3] Dillmann, R. (2004). Teaching and learning of robot tasks via observation of human performance. *Robot. Auton. Syst.*, *47*(2), 109–116
[4] Ye, G., & Alterovitz, R. (2017). Demonstration-guided motion planning. *ISRR 2017*, 291–307.
[5] Gao, X., Silvério, J., Calinon, S., Li, M., & Xiao, X. (2022). Bilateral teleoperation with object-adaptive mapping. *Complex & Intell. Syst.*, *8*(4), 2983–2990.
[6] Guo, C., & Sharlin, E. (2008). Exploring the use of tangible user interfaces for human-robot interaction: A comparative study. *CHI 2008*, 121–130.

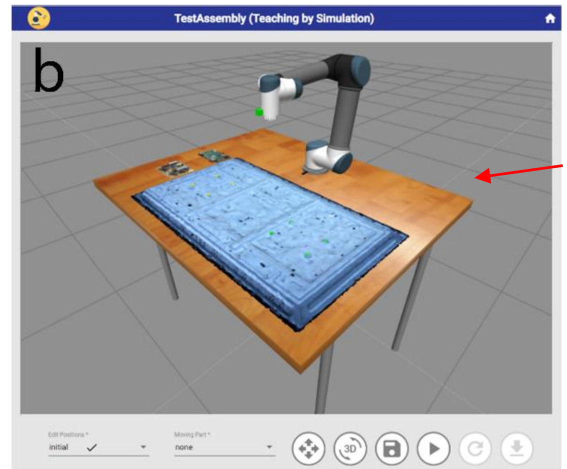
Research motivation and goal

- **Research motivation**
  - Teaching assembly tasks to robots remains challenging, particularly for small and medium-sized businesses that lack required resources
- **Research goal**
  - Propose a new LfD method that eliminates the need for elaborate setups or specialized equipment while combining the intuitiveness of passive LfD methods with the efficiency and accuracy of active LfD methods
- **Proposed LfD method (in a nutshell)**
  - User demonstrates the assembly task using a web browser in 3D
  - User simulates the robotic execution to promptly assess the outcome of the teaching process

Proposed LfD method: Typical user scenario

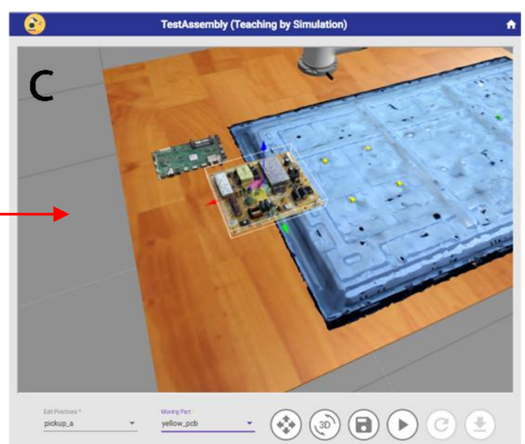
1. User enters assembly info and uploads 3D CAD files





2. Teaching UI loads (3D setting, assembly items, robot, UI controls to showcase the assembly process)

3. User selects a named keyframe and specifies object position (repeat for all named keyframes)



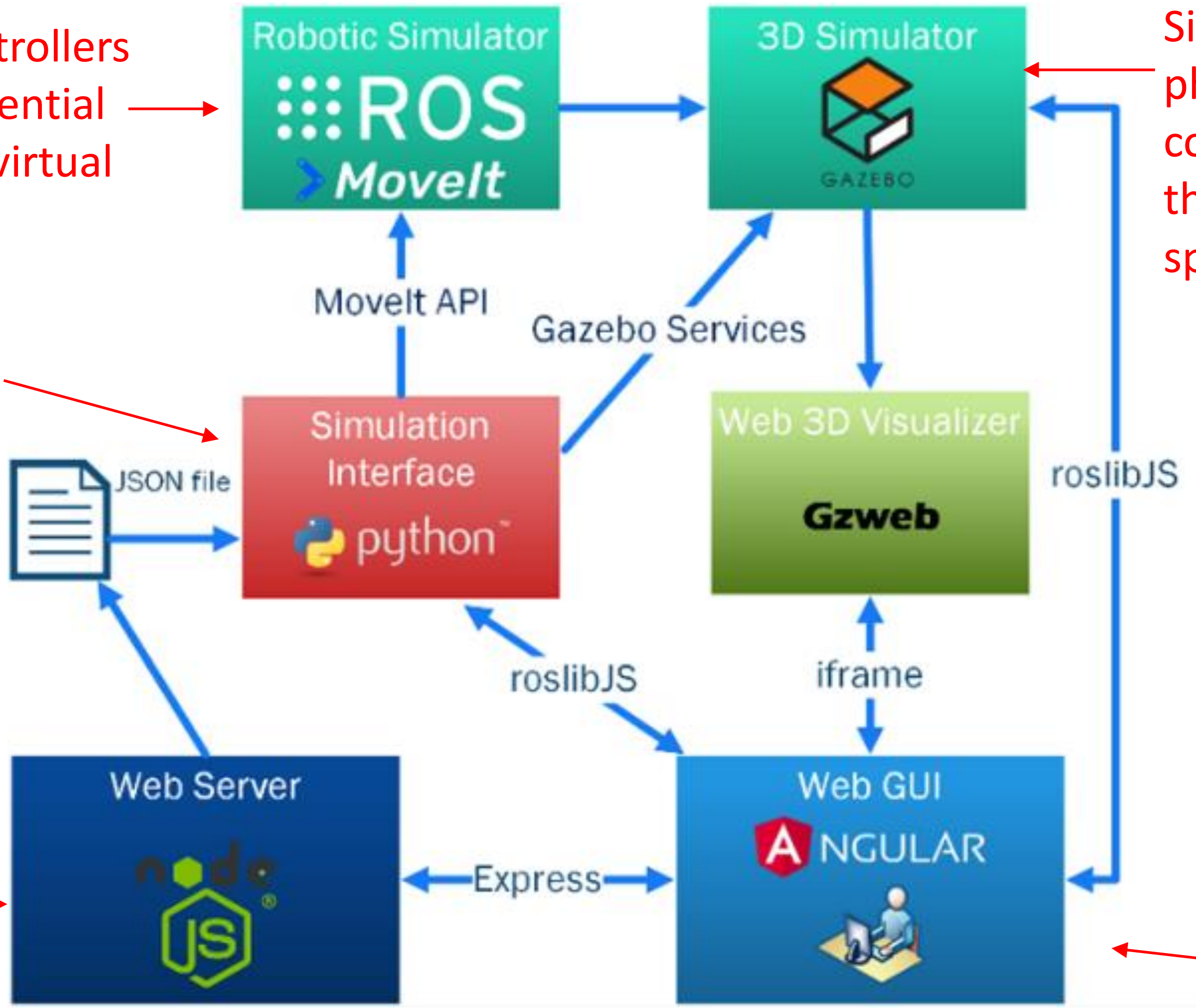
4. User may press the play button to run the simulation and make changes to the keyframes. Assembly process can be exported in JSON file

Proposed LfD method: System architecture and technologies

Simulates the controllers and interfaces essential for operating the virtual robot

Functions as a bridge between the front-end web GUI and the simulation platform

Handles file transfer between the client and the ROS server



Simulates the physical components of the assembly space

Implements the user interface

User evaluation study: Goal and Methodology (1/3)

- **Goal**
  - Compare usability metrics of the proposed simulation-based system vs. a cutting-edge computer vision system for the same assembly task
- **Methodology**
  - Within-subjects research design
  - Participants
    - 25 participants (9 females, 16 males)
    - 19 to 52 years old (M=28.1, SD=8.1)
    - No robotics expertise, highly experienced computer users
  - Assembly task
    - Position two distinct PCB boards at the rear of an LCD TV
    - Task in a real-world factory (OP1 in the photo)

User evaluation study: Goal and Methodology (2/3)

- **Methodology (cont.)**
  - Computer vision-based system
    - Existing system proposed in the literature[1]
    - User performs the assembly task physically while an RGBD sensor records the process. ML algorithms extract relevant information
    - Requires robust graphics card (we used Nvidia GTX2080Ti)
    - GUI on tablet used while physically executing the assembly (set up assembly, capture LfD, add info for keyframes, visualize learnt assembly)

[1] Haage, M., Piperagkas, G., Papadopoulos, C., Mariolis, I., Malec, J., Bekiroglu, Y., Hedelind, M., & Tzovaras, D. (2017). Teaching assembly by demonstration using advanced human robot interaction and a knowledge integration framework. *Proc. Manuf. 11*, 164–173

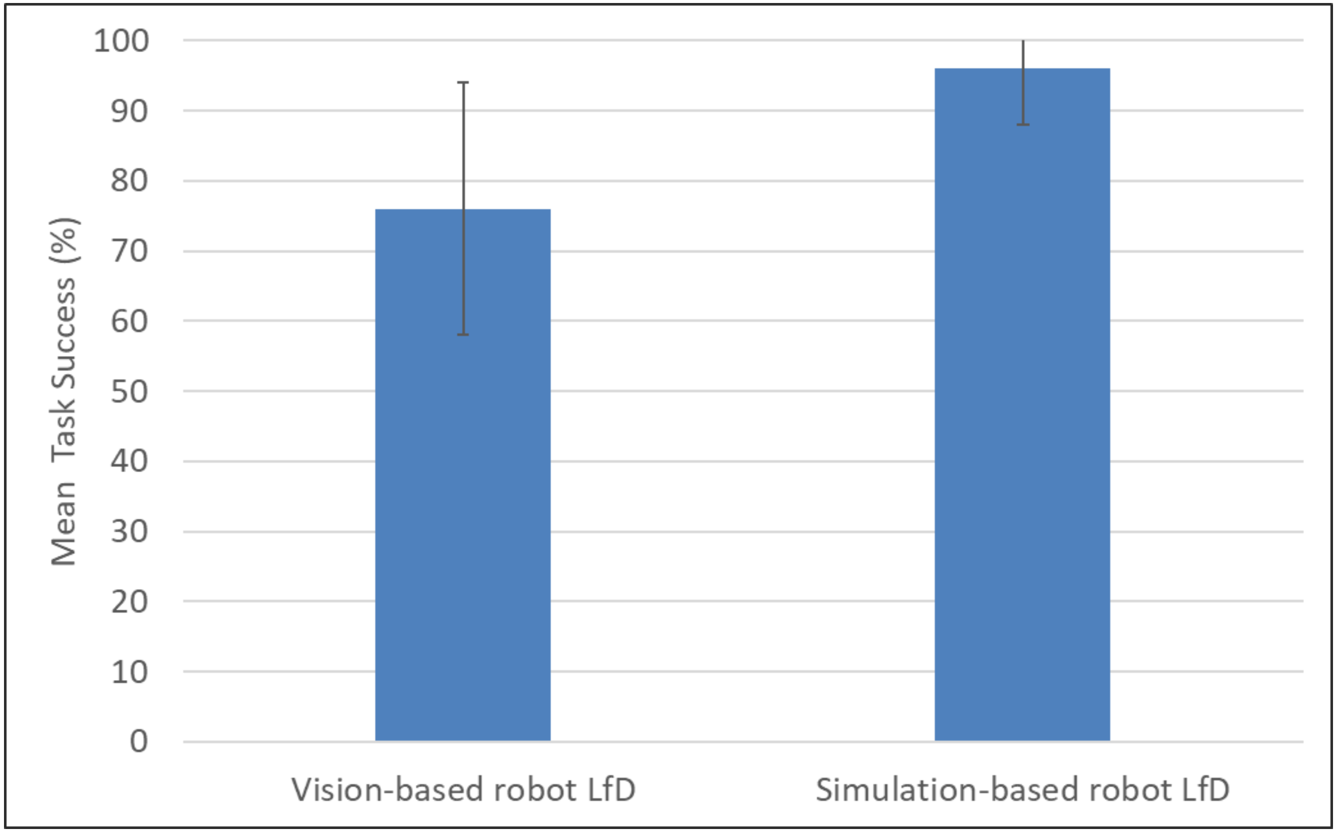
User evaluation study: Goal and Methodology (3/3)

- **Methodology (cont.)**
  - Procedure
    - Users see brief video on the real-world LCD TV assembly task and physically execute it on a table
    - Users perform the assembly task using the two robot LfD systems (simulation-based, vision-based). Randomized presentation order
    - At the end, users complete questionnaire: a) demographic questions, b) Greek SUS[1,2], c) system preference (one or the other)
  - Usability metrics collected
    - Task success (%), Time on task (sec), SUS score (0-100), System preference

[1] Orfanou, K., Tselios, N., & Katsanos, C. (2015). Perceived usability evaluation of learning management systems: Empirical evaluation of the System Usability Scale. *IRRODL*, 16(2), 227–246.
[2] Katsanos, C., Tselios, N., & Xenos, M. (2012). Perceived usability evaluation of learning management systems: A first step towards standardization of the System Usability Scale in Greek. *PCI 2012*, 302–307.

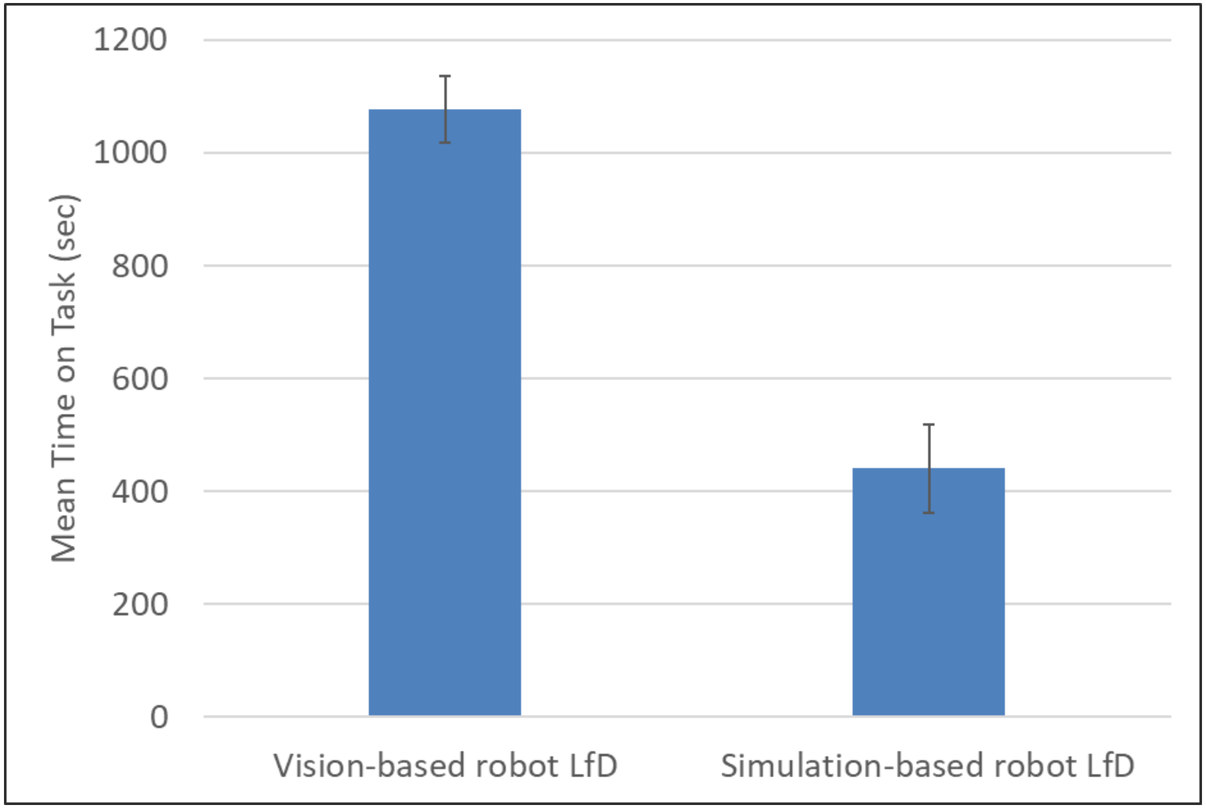
Evaluation study: Results (1/4)

- LfD system version **did not significantly affect** users' task success
  - McNemar's chi-square test: p=0.125, ns



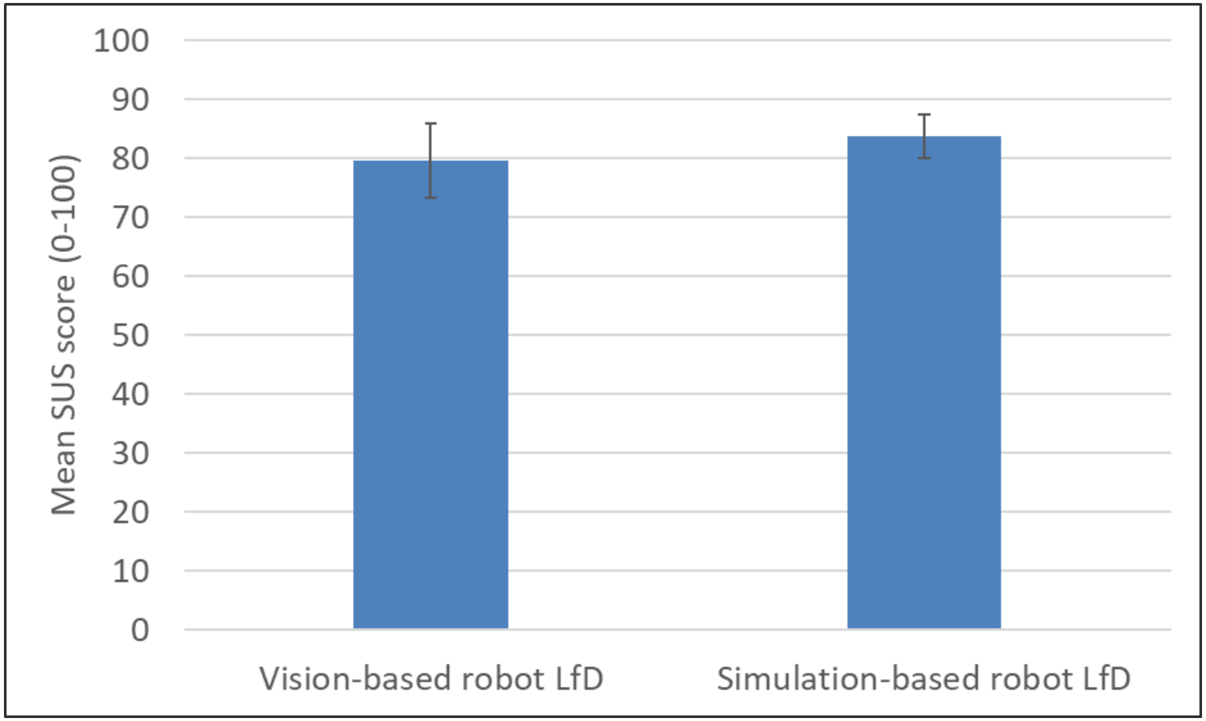
Evaluation study: Results (2/4)

- Users of the vision-based system **required significantly and largely more time** compared to the simulation-based system
  - Two-tailed dependent t-test: $t(24)=15.124$, $p<0.001$, $r=0.95$



Evaluation study: Results (3/4)

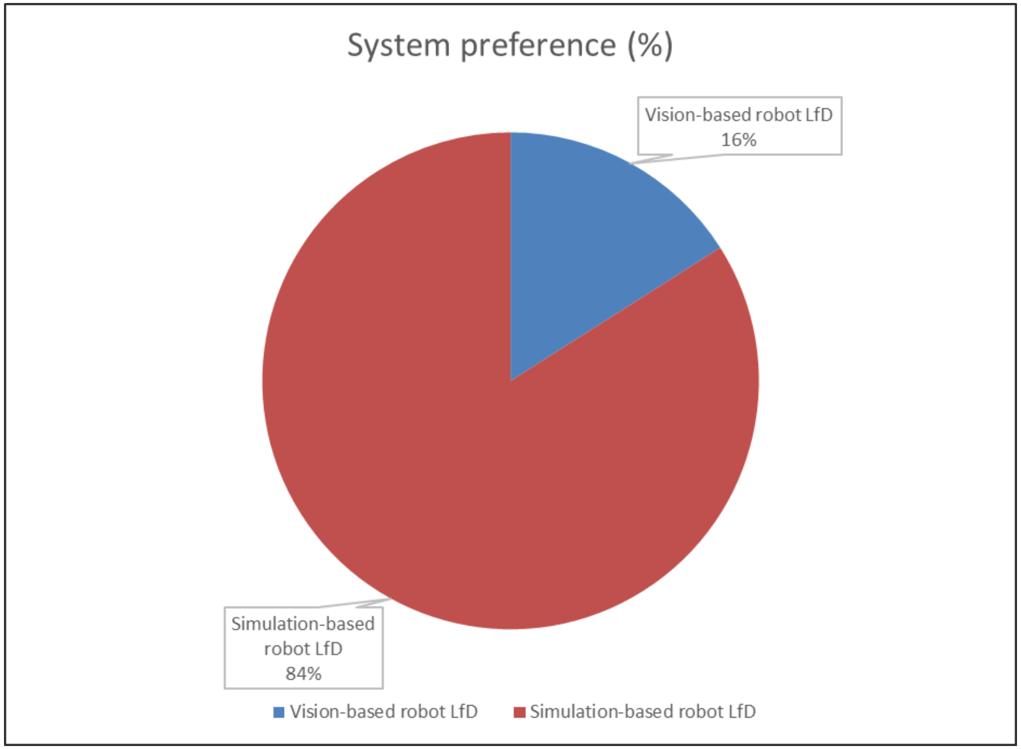
- LfD system version **did not significantly affect** the SUS score
  - Two-tailed Wilcoxon signed-rank: z=1.107, p=0.268, ns
  - Both systems => Good to Excellent[1]



[1] Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS Scores mean: Adding an adjective rating scale. JUS.

Evaluation study: Results (4/4)

- The **% of participants who preferred the simulation-based system was significantly higher** than the expected (50%)
  - Two-tailed one-sample binomial test: $p<0.001$
  - Users would choose 5.25 times more the proposed system



Future directions

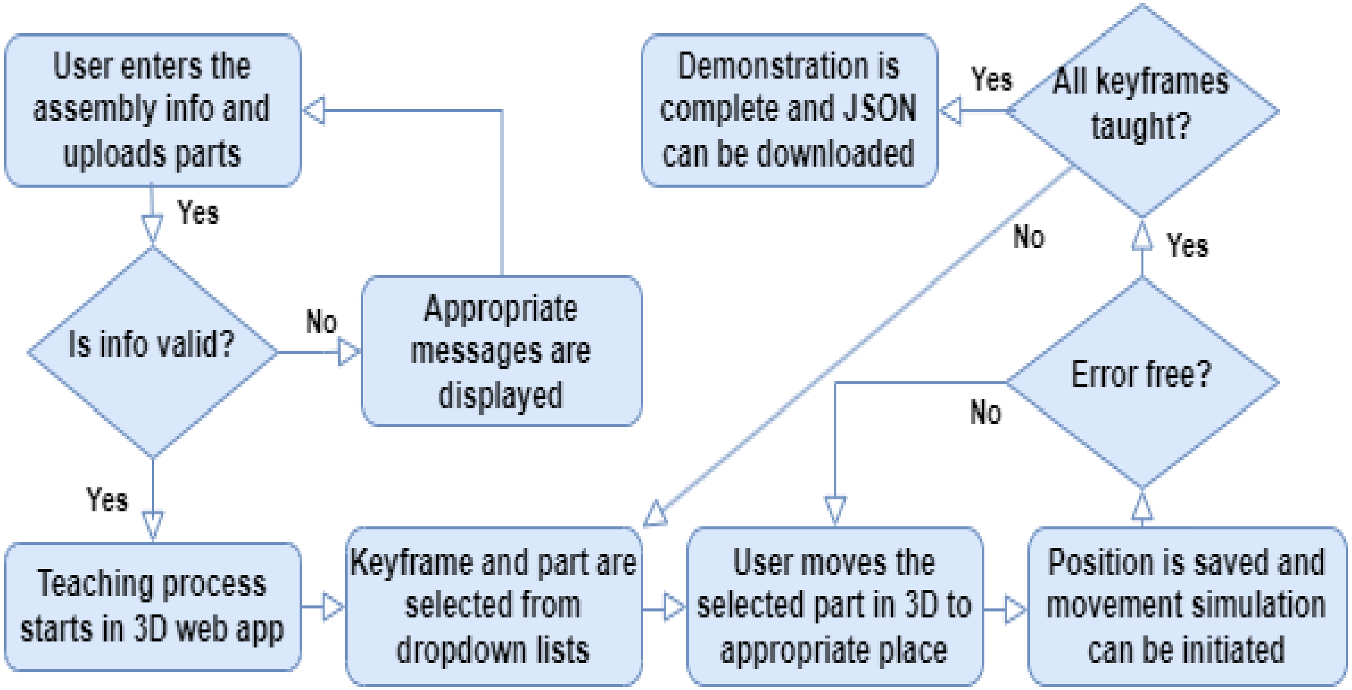
- Investigate applicability of the proposed LfD method in **different types of assembly tasks**
  - Some assemblies require a specific amount of torque and force, which isn't easy to describe through our UI without specialized hardware
- Implement a **grasp planning module**
  - Robot grasping plays an important role in many assembly processes
- Compare **proposed LfD approach vs. teleoperated LfD**
  - User study compared only against a passive LfD method (computer vision)

Summary & Questions

- **Summary**
  - We presented a 3D simulation-based approach for instructing robots to perform tasks in factories (learning from demonstration)
  - Proposed LfD method is simple to set up, doesn't require any additional hardware, allows for instant assessment of teaching effectiveness
  - Within-subjects user testing study found that the proposed method is significantly more time-efficient and preferred than vision-based LfD. No significant difference was observed for task success and SUS scores
- **Questions?**
  - Shoot!
- **More questions and not enough time! No worries ☺**
  - Christos Katsanos (ckatsanos@csd.auth.gr)

Backup/Extra slides

Proposed LfD method: Simplified user's workflow



Proposed LfD method: More on the system architecture (1/3)

- **Web GUI Module**
  - User can configure the assembly attributes and perform the robot LfD through the 3D simulation environment
  - Takes and stores object locations allowing it to use those coordinates to start or stop the simulation using the simulation interface module
  - Built using Angular, incorporates a customized version of gzweb, a web viewer for Gazebo (implements the 3D simulation environment)
  - roslibJS library for communication between the web GUI and the ROS (direct control of the 3D environment, extraction of assembly part states)

- **3D Simulator Module**
  - Gazebo framework used to simulate the physical components of the assembly space (construct space, gravity, friction, collision handling etc)
  - gzweb is used to display the 3D simulation in the web GUI (tools to move or rotate 3D objects, change simulation characteristics)

Proposed LfD method: More on the system architecture (2/3)

- **Web Server Module**
  - Oversees upload of 3D CAD files of assembly objects, accepts demonstration data objects and saves them to JSON files, and exports JSON files from the server to the client
  - Developed using Node.JS and the Express library, with a RESTful API to simplify communication between the front-end web GUI and the web server module

- **Robotic Simulator Module**
  - MoveIt library is used for this module, providing improved flexibility for navigation planning towards desired 3D places as well as making it easier to adjust the motion planning algorithm for the best results.
  - The popular Universal Robots UR5 robotic arm (carry up to 5kg, operating radius of 850 mm) was chosen to simulate assembly execution

Proposed LfD method: More on the system architecture (3/3)

- **Simulation Interface Module**
  - Functions as a bridge between the front-end web GUI and the simulation platform
  - Receives user input, directs the robot to move to desired locations, manages its gripper operations, and pauses the simulation physics when necessary
  - Designed as a ROS node in Python. Interacts with ROS, MoveIt, and Gazebo to execute the needed operations
  - Python API for MoveIt is deployed to guide the robot towards locations that the user has demonstrated (recorded in a JSON file) and carry out the assembly process
  - The module also employs the Cartesian path planning algorithm to produce motion trajectories