

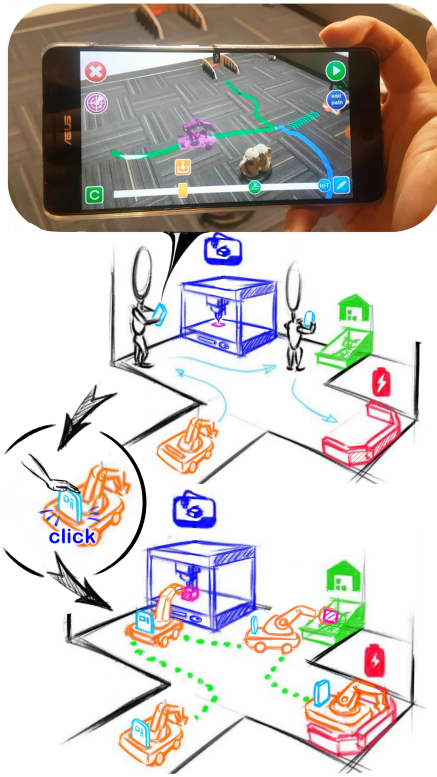


Figure 1: V.Ra workflow. Using an AR-SLAM mobile device, the user first spatially plan the task with the AR interface, then place the device onto the mobile robot for execution.

V.Ra: An In-Situ Visual Authoring System for Robot-IoT Task Planning with Augmented Reality

Yuanzhi Cao

Purdue University
West Lafayette, IN, USA
cao158@purdue.edu

Fan Li

Tsinghua University
Beijing, China
li-f15@mails.tsinghua.edu.cn

Ke Huo

Purdue University
West Lafayette, IN, USA
khuo@purdue.edu

Zhuangying Xu

Purdue University
West Lafayette, IN, USA
xu970@purdue.edu

Wentao Zhong

Purdue University
West Lafayette, IN, USA
zhong133@purdue.edu

Karthik Ramani

Purdue University
West Lafayette, IN, USA
ramani@purdue.edu

ABSTRACT

We present V.Ra, a visual and spatial programming system for robot-IoT task authoring. In V.Ra, programmable mobile robots serve as binding agents to link the stationary IoTs and perform collaborative tasks. We establish an ecosystem that coherently connects the three key elements of robot

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland UK

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5971-9/19/05.

<https://doi.org/10.1145/3290607.3312797>

task planning (human-robot-IoT) with one single AR-SLAM device. Users can perform task authoring in an analogous manner with the Augmented Reality (AR) interface. Then placing the device onto the mobile robot directly transfers the task plan in a what-you-do-is-what-robot-does (WYDWRD) manner. The mobile device mediates the interactions between the user, robot and IoT oriented tasks, and guides the path planning execution with the SLAM capability.

CCS CONCEPTS

• **Computer systems organization** → **Embedded and cyber-physical systems**; *Robotics*; • **Information systems** → **Information systems applications**; *Multimedia content creation*.

KEYWORDS

Robotic task authoring, Augmented-Reality, SLAM, Path planning, Internet-of-Robotic-thing

INTRODUCTION

The vision of *ubiquitous computing* has been emerging rapidly as the Internet of Things (IoT) based electronics are getting smaller, lower in cost, proliferating and being embedded in our everyday environment. Typically, human-IoT interactions take the form of transforming IoT data into informative knowledge, augmenting human sensory capabilities, and assisting humans to make correct and efficient decisions. However, the IoT devices are mostly stationary and have limited physical interactions particularly with each other. In conjunction, the concept of Internet of Robotic Things (IoRT) has not been widely explored in practice across the IoT and robotics communities, and an authoring system for such robot-IoT interactive task planning is underdeveloped. We envision the emergence of programmable mobile robots in a near future to serve as key medium to conduct coordinated and collaborative tasks with surrounding IoTs. In this vision, the mobile robots are combined with the embedded multiple stationary IoTs to create new types of workflows and in addition also extend humans' motor capabilities. We leverages the advancing SLAM techniques to globally associate the user, IoTs, and robots together. Users first freely examine and explore the IoT environment within a mobile AR scene, then seamlessly transfer their insight about the tasks regarding the environmental factors such as the path planning, as well as the semantic knowledge such as the situational awareness from IoTs to the robots. Further, SLAM also enables a novel embodied programming modality, namely, users demonstrate a sequential tasks to the robots by physically visiting the IoTs. To this end, we present V.Ra (Virtual Robotic assistant), an in-situ authoring interface for robot-IoT task planning using a mobile AR-SLAM device.

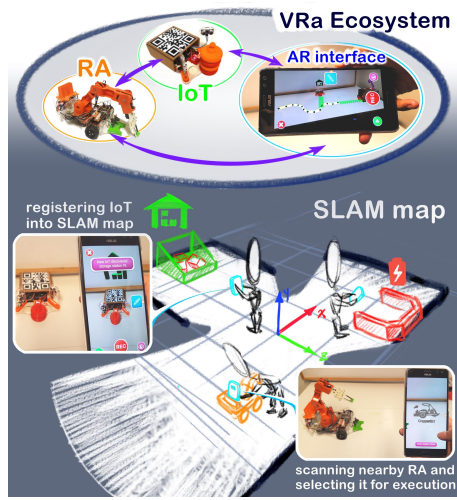


Figure 2: The ecosystem design of VRa. The three key elements for robot task planning (authoring interface, RA, and IoT) are cohesively connected together by the SLAM map.

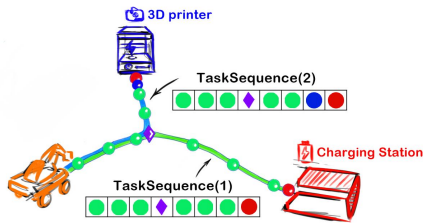


Figure 3: Tasks are represented by TaskSequence in V.Ra system, which are formed by four types of Nodes.

RELATED WORK

Within the authoring interface for mobile robot, users need to be spatially aware of the physical environment and the robots. Previous works introduced an external vision system to track the robots and fed the live camera view to the interface [3, 8]. However, this approach limits the authoring scene to the perspective of the camera only, which is usually fixed. Recent researchers employed mobile AR interfaces and associated the robots within the AR scene, with hand-held [4, 6] or head-mounted [1] devices. However, the limited field-of-view constrains the human-robot interaction experience. Other works separated the authoring interface and navigation by equipping robots with on-board SLAM capabilities [5]. This way, user referred to a scanned map of the real scene as authoring context and the robot conducted tasks using the same map [2, 7]. However, the pre-scanned SLAM map, once created, remains static and cannot adapt to the changes in the environment. Informed by these previous works, we propose a mobile AR authoring interface with which users can spatially author the tasks by either explicitly defining navigation paths or implicitly visiting the IoTs by just walking to each of them. Moreover, we emphasize a transparent knowledge transferring between human and the robots by allowing robots to use the same AR device as ‘eyes’ and ‘brain’ directly. We further increase the adaptability of the robots against environment changes as we rely only on on-the-fly updated SLAM maps that enables in-situ task authoring.

V.RA SYSTEM DESIGN

System Workflow

In VRa’s ecosystem, these three key elements are cohesively connected by the SLAM map, as is illustrated in Figure 2. We walk through our workflow with a typical use scenario (Figure 1). In a household environment, users first select a robot for the desired tasks from the available nearby ones. This allows an AR authoring interface to be specialized based on the capabilities of this particular robot. The spread IoTs can be registered into the SLAM map through a one-time QR code scanning. Users then access the embedded knowledge from the IoTs in AR view. Using our authoring interface, users formulate a group of navigation paths, IoT interactions, and other time and logic constructs to achieve the desired robot-IoT coordination. After the authoring is finished, users physically place the authoring device onto the modular slot of the robot, and the system guides the robot to execute the tasks. Because of the transparency between the users’ intents and robots’ actions in the AR authoring phase, we achieve programming a robot in a WYDWRD fashion.

Task Planning Construct

To start designing the authoring interface for mobile robot task planning, we first extract the basic elements of the task. The nature of our work is robot planning for physical tasks that involves



Figure 4: Main interface design of V.Ra system (top). An icon reference list for interactive functions (bottom).



Figure 5: Navigation authoring with (1) spatial movement, (2) hand-drawn segment line, and (3) hand-drawn curve.



Figure 6: User first scan IoT's QR code (1) to register it into the AR scene (2), then touch its virtual icon (3) to access the function list (4). A green path for visual confirmation (5)

interactions with different devices at various locations. The planned task may take a long period of time to execute, and it involves logic conditions that handle unexpected situations dynamically. By referring to previous programming protocols for IoTs and robots and catering them to our specifics, we develop the following **Nodes** to represent task elements and construct a TaskSequence.

Navigation Node ● : contains 3D coordinates to represent the path for the robot to travel through.

Action Node ● : defines an action event that relates to the robot and/or the IoT device.

Time Node ● : contains information that allows the system to perform time based behaviours.

Logic Node ◆ : contains check condition that allows the system to perform logic driven tasks.

These **Nodes** are the basic abstractions that form any user authored task in V.Ra, namely, a construct array in our system, called **TaskSequence**. User can add new Nodes or manipulate the existing Nodes in the TaskSequence. When executing in the Play mode, the system guides the robot to run through each Node sequentially thus accomplish the authored task. The logic driven events are realized by multiple TaskSequences with each one representing one task line. Figure 4 illustrates a logic event with its corresponding TaskSequences. The robot checks the condition at the Logic Node and decides which path to take. If the battery is low, it will continue on TaskSequence(1) and go to the Charging Station; otherwise it will proceed on TaskSequence(2) and go pick up the 3D printed part when it is finished. Note that the *wait...until* function is realized by the blue Time Node.

V.Ra Interface and Interaction

The interface design of V.Ra system is shown in Figure 3. Users can create TaskSequence and preview it in the AR view and also in the **EventLine**. To start a new task planning after selecting a robot, a user first defines the basic robot action with *AddPath* and *ScanIoT*. User then preview the authored task with *EventLine*, user can *Insert* new function, or *Edit* existing tasks. The user has the option to create periodic robot tasks (i.e. repeat everyday) using the *Repeat* function. When task authoring is finished, user can activate the *Play Mode* and place the mobile device onto the robot. The robot then starts the execution of the planned tasks by sequentially running all the Nodes in the TaskSequence.

Add robot path. Navigation Nodes are the majority Nodes that form the TaskSequence in our system as it defines the path for the robot to navigate in the environment. There are two ways to add navigation nodes: 1) record spatial movement (REC), or 2) hand-draw the path on the screen, as illustrated in Figure 5. The hand-drawn method are suitable for path planning in a smaller area with finer control, while the REC is designed for conveniently creating large room-level navigation paths through embodied spatial movement. Each created path is broken into a series of Navigation Nodes and are added to the end of the TaskSequence. After a navigation node is added, a green path will be displayed in the AR scene giving the user active visual feedback.

Add IoT interaction. Robot-IoT interaction encompasses the majority of the Action Node in the system. Other Action Nodes include IoT-only and robot-only functions. To add a new robot-IoT

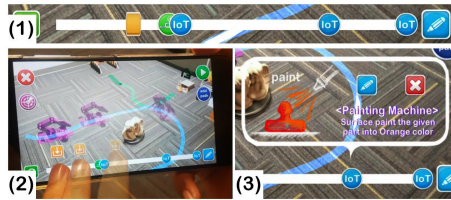


Figure 7: EventLine represents the task in a linear and compact format (1). User can drag the handlebar to preview task (2), and tap on the icon to access its detailed information (3).



Figure 8: User can insert (1) non-robotic IoT function (2), Time Node (3), or Logic Node with an alternative task line (5) triggered by working status of the connected devices (4).

interaction Node, the user first needs to register the IoT device into the AR scene, which is achieved through a one-time scan of the IoT's QR code (Figure 6 (1-2)). This not only brings an interactive 3D virtual model into the AR scene (Figure 6 (3)), but also imports semantic information into the system, like IP address and interaction protocol. After the IoT registration, user can press its virtual icon to access its function list and select to add an Action Node (Figure 6 (4)), to the end of the TaskSequence. When a robot-IoT interaction Action Node is added, a green arrow path appears, pointing towards the IoT device as a visual indicator (Figure 6 (5)).

EventLine task visualization. While the AR view is good for spatial task visualization, it is constrained by the view of the display, which makes it difficult for user to perform global monitoring and manipulation of the entire task. To compensate for this, we introduce an abstract visualization of the task, called **EventLine**. The design of EventLine is inspired by the timeline concept used commonly in the animation industry. The difference being that, in our case, the task is governed by events, such as robot navigation and IoT interaction. As is illustrated in Figure 7 (1), the EventLine displays all the non-navigation Nodes as icons, and the user can access its details to review and edit (Figure 7 (3)). By dragging the handlebar, user can preview the task with a virtual robot (Figure 7 (2)). This is designed to simulate the robot path and avoid unexpected errors. When multiple task lines exist, only the currently selected task line will show its EventLine. User can switch the selected task line by tapping on it in the AR view. The selected task line will be highlighted with the white indicator flowing through it.

Insert. By dragging the EventLine handlebar, users can insert new Nodes into the designated position in the TaskSequence, which is illustrated by the position of the virtual robot (Figure 8 (1)). These Nodes are 1) non-robotic IoT Action Nodes, 2) Time Nodes, and 3) Logic Nodes. To insert an IoT function, the system provides the user with a list of all the IoT devices that are connected to the system with the embedded functions (Figure 8 (2)). To insert a Time Node, users either set a fixed wait time (Figure 8 (3)), or define a *wait...until* condition that is triggered by the IoT working status or sensing values. User can repeat the process and create composite AND/OR boolean conditions. In terms of the Logic Node, upon selecting, an alternative TaskSequence will be created and user will be asked to define the trigger condition (Figure 8 (4)). This allows users to define new task line that branches from the Logic Node position (Figure 8 (5)). When executing a task with multiple TaskSequences, the system will run from the default TaskSequence (the first created TaskSequence) and decides which TaskSequence to continue at an Logic Node, based on the condition check.

Edit. By utilizing the EventLine, V.Ra allows user to edit their authored task by looping, mirroring, or deleting part of the selected EventLine. The copy and mirror functions are designed to increase the authoring efficiency for scenarios like *repeat this floor sweeping path 10 times* (loop), or *go back to where you came from* (mirror). When accessing the Edit mode, two interactive markers will appear on

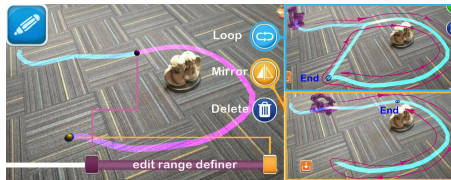


Figure 9: The Edit function for partially loop, mirror, or delete the authored task for the defined range.

the EventLine with the middle part highlighted. Users can drag the markers to define the edit range, and the corresponding part in the AR view will also be highlighted (Figure 9).

CONCLUSION

This paper has presented V.Ra, a spatially situated visual programming system for household robot task planning. We have adopted a workflow approach of one single AR-SLAM device for task authoring and robot execution, and presented the system design. In V.Ra, humans and smartthings enhance each other's capability within the fluidly connected ecology, such that spatially oriented collaborative tasks can be operated with lightweight system requirements. We believe that V.Ra opens an inspiring perspective for researchers to reconsider human's role in the coming era of Internet-of-Robotic-Things.

ACKNOWLEDGEMENT

This work was partially supported by the NSF under grants FW-HTF 1839971, IIS (NRI) 1637961 and IIP (PFI:BIC) 1632154. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

REFERENCES

- [1] Yuanzhi Cao, Zhuangying Xu, Terrell Glenn, Ke Huo, and Karthik Ramani. 2018. Ani-Bot: A Modular Robotics System Supporting Creation, Tweaking, and Usage with Mixed-Reality Interactions. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 419–428.
- [2] Matei Ciocarlie, Kaijen Hsiao, Adam Leeper, and David Gossow. 2012. Mobile manipulation through an assistive home robot. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 5313–5320.
- [3] Sunao Hashimoto, Akihiko Ishida, Masahiko Inami, and Takeo Igarashi. 2011. Touchme: An augmented reality based remote robot manipulation. In *21st Int. Conf. on Artificial Reality and Telexistence, Proc. of ICAT2011*.
- [4] Ke Huo, Yuanzhi Cao, Sang Ho Yoon, Zhuangying Xu, Guiming Chen, and Karthik Ramani. 2018. Scenariot: Spatially Mapping Smart Things Within Augmented Reality Scenes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 219.
- [5] Ke Huo, Tianyi Wang, Luis Paredes, Ana M Villanueva, Yuanzhi Cao, and Karthik Ramani. 2018. SynchronizAR: Instant Synchronization for Spontaneous and Spatial Collaborations in Augmented Reality. In *The 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, 19–30.
- [6] Shunichi Kasahara, Ryuma Niiyama, Valentin Heun, and Hiroshi Ishii. 2013. exTouch: spatially-aware embodied manipulation of actuated objects mediated by augmented reality. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*. ACM, 223–228.
- [7] Hai Nguyen, Matei Ciocarlie, Kaijen Hsiao, and Charles C Kemp. 2013. Ros commander (rosc): Behavior creation for home robots. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 467–474.
- [8] Daisuke Sakamoto, Yuta Sugiura, Masahiko Inami, and Takeo Igarashi. 2016. Graphical instruction for home robots. *Computer* 49, 7 (2016), 20–25.