A Human Robot Interaction Interface for Uncalibrated Visual Servoing

Laura Petrich

***Abstract* -- Substantial research has been published in Human Robot Interaction (HRI), tracking, and visual servoing over the past decade. We present an intuitive system based on the user-centered design for setting and executing visual tasks. Our HRI system expands on the work by Gridseth et *al*. [1], incorporating visual task specification [2], registration based tracking [3] and uncalibrated image-based visual servoing [4].**

I. INTRODUCTION

Real world robotic applications require robotic systems to adapt and work reliably in unstructured and dynamic environments. A user-centered design is central for robotic software to help bridge the gap between research and everyday real world applications. Visual servoing (VS) - also known as vision-based robot control - uses visual feedback to form an error function that drives a robot control law. The primary objective in VS is to minimize this error function. Dodds et al. in [2] present a concise and provably complete task specification language. Moreover, they have derived a related family of image-based task encodings used to monitor task performance in relation to the formed error function. Gridseth et al. shows that a practical, and usable VS system must address four general areas: Human Robot Interaction, multi-view geometry, tracking, and visual servoing [5]. This study integrated the HRI and multi-view geometric task constraint work done by Gridseth et al. [1], the Modular Tracking Framework (MTF) for registration based tracking work of Singh and Jagersand [3], and the Uncalibrated Visual Servoing (UVS) work of Ramirez and Jagersand [4]. This produced an intuitive, touchscreen interface based on the user-centered design approach for setting and executing geometric tasks. The primary contributions of this project are:

- an intuitive interface that guides the user through the task specification process,
- complete MTF integration with our HRI interface system,
- autonomous tracking and visual servoing coupled with semi-autonomous task specification to reduce cognitive strain on the user.

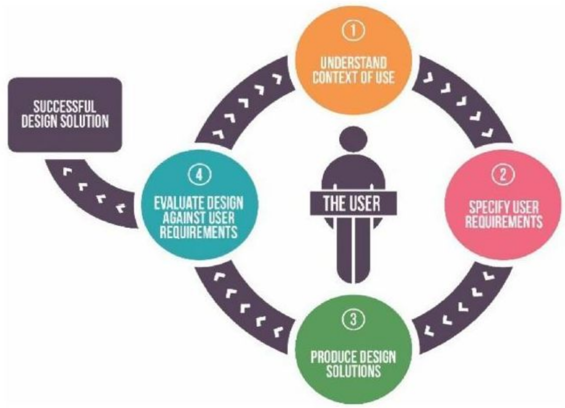

Fig 1. The User-Centered Design Approach [6]

II. VISUAL SERVOING

One of the earliest papers in which visual servoing is described dates back to 1979 [7]. Model-based classical VS approaches provides knowledge about the model and/or system parameters are known beforehand. Three broad classifications exist for VS control techniques: image - based, position/pose - based, or a hybrid approach. Image Based Visual Servoing (IBVS) often depends on a parametric model of the image Jacobian, which needs to be derived analytically in advance for each feature type. This requires the camera intrinsics to be known, and possibly also the 3D parameters (ie. depth for point features) [8]. Position Based Visual Servoing (PBVS) requires a calibrated camera with known intrinsics, as well as a known geometric model of 3D features to estimate the relative camera pose, with respect to the desired object [9]. Given that real-world environments are dynamic and unstructured, model-based approaches to VS are impractical. Uncalibrated Visual Servoing (UVS), presented by Jagersand et *al*. [10], is a viable approach for robot control in such real-world environments.

Uncalibrated visual servoing applies IBVS, without a-priori models or camera calibration, to formulate an error function that drives a robot control law. This control law in UVS is defined entirely in the image space, V, and gives a mapping directly from this space to the robot joint space. Let $q \in R^n$ be the configuration of a robot with n joints, and $s \in R^m$ represent m visual features. The primary objective of VS is to minimize an image error, denoted:

$$e(t) = s - s^* \qquad (1)$$

where s^* is the desired goal configuration of the m visual features in $s \in R^m$. This then relates the image space, s, and robot joints, q, with a visual-motor function $F : R^n \rightarrow R^m$, denoted:

$$s = F(q) \qquad (2)$$

Taking the derivative, and using a discrete time approximation we can formulate a proportional control law as:

$$\dot{q} = -\lambda \widehat{J}^{\dagger} e \tag{3}$$

where $\dot{q}$ denotes joint velocities, λ is the step size of robot motion per iteration, and $\widehat{J}^{\dagger}$ is the Moore-Penrose pseudoinverse of the Jacobian matrix. Here, the Jacobian matrix, J, is dependant on the number of degrees of freedom being controlled and the tracked features, and is defined as:

$$J = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{q})}{\partial q_1} & \cdots & \frac{\partial f_1(\boldsymbol{q})}{\partial q_m} \\ \vdots & & \vdots \\ \frac{\partial f_k(\boldsymbol{q})}{\partial q_1} & \cdots & \frac{\partial f_k(\boldsymbol{q})}{\partial q_m} \end{bmatrix} \tag{4}$$

where q represents a vector of joint angles, and f, a vector of image point features. In UVS, the Jacobian must be estimated before error minimization can begin. To find this initial estimate, small perturbations of each joint are performed, and rate of change in error, with respect to the joint motions, are estimated to produce an error vector. Using this error vector, we can estimate the partial derivative with respect to each joint:

$$\hat{J} = \left[\begin{bmatrix} \vdots \\ \frac{\Delta \boldsymbol{e}_{q_1}}{\Delta q_1} \\ \vdots \end{bmatrix} \cdots \begin{bmatrix} \vdots \\ \frac{\Delta \boldsymbol{e}_{q_m}}{\Delta q_m} \\ \vdots \end{bmatrix} \right] \tag{5}$$

with Δq_i denoting finite increments in the motion of joint i performed to prompt the error change Δe_{q_i}. With an initial Jacobian estimate, the visual error can be minimized by iteratively solving the control law (3). The Jacobian is updated in each iteration using a rank 1 updating formula in the Broyden hierarchy [16], detailed as:

$$\hat{J}_{k+1} = \hat{J}_k + \alpha \frac{(\Delta \boldsymbol{e} - \hat{J}_k \Delta \boldsymbol{q}) \Delta \boldsymbol{q}^T}{\Delta \boldsymbol{q}^T \Delta \boldsymbol{q}} \tag{6}$$

with α representing the applied learning rate.

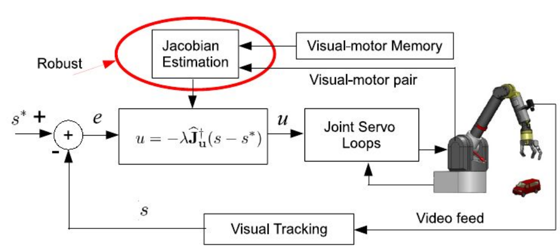

Fig 2. System block diagram proposed by Shademan et *al.* [11].

III. VISUAL TASK CONSTRAINTS

A visual task can be defined as the objective of bringing the pose and/or position of the robot end effector to a target position in the robots workspace, W. Let $\{f_1, f_2, ..., f_n\}$ represent the list of all observable feature points in image space, V. Dodds et al. [2], represent a *task function*, T, from the feature space, $F := V^n$, to the codomain $\{0, 1\}$. The task function, T, is defined as:

$$T(f) = 0 \tag{7}$$

The codomain represents whether the task specification was satisfied or not. If (7) holds, the task is satisfied at f.

Dodds et *al.* also introduce *task decidability* and *task encoding*. A task, T, is deemed *decidable*, if, given the camera configuration, it can be determined whether T was satisfied at f. Formally, this is denoted through the concept of an *encoded task function*:

$$E_T(f) = 0 \tag{8}$$

where each different geometric error constraint (eg. point-to-point), has a corresponding encoding to be used as the error function (1) minimized during visual servoing. The authors go on to prove that any point-to-point task is decidable on a family of injective two-camera models. Furthermore, they state that complex tasks can be decomposed into simpler ones that are provably decidable using operators that preserve task decidability. These decomposed tasks can be stacked and used simultaneously resulting in the encoded task function:

$$E(\mathbf{y}) = \begin{pmatrix} E_1(\mathbf{y}) \\ E_2(\mathbf{y}) \\ \vdots \\ E_k(\mathbf{y}) \end{pmatrix} \tag{9}$$

where $\mathbf{y}$ denotes the image measurement corresponding with the geometric error constraint [1].

IV. HUMAN ROBOT INTERACTION

Substantial real-world applications of robotics have been achieved in controlled environments, such as factories, where they can be preprogrammed to accomplish specific tasks accurately, reliably, and autonomously. However, in unstructured and dynamic environments, such as personal homes, hospitals and non-assembly based businesses, having a robot that is able to continuously adapt accordingly is of high value. Subsequently, this led to an increase in Human Robot Interaction (HRI) [12][13][14] research over the past decade. HRI research focuses on understanding, designing, and evaluating robotic systems to work alongside humans, with varying degrees of autonomy [15].

Human Robot Interaction research is dichotomously categorized into remote interaction, and proximate interaction. The former occurs when the human and robot are not located in the same physical region; this separation could be due to design or necessity, such as with the Mars Rovers. The latter category occurs when the human and robot are co-located, such as applications in assistive/service robots. Within these two general categories, real-world applications can be further segregated into mobility (ie. teleoperation), physical manipulation (ie. telemanipulation), or social interaction where humans and robots connect through companionship [14].

Assistive robotics is a branch of HRI focused on the development of robotic aids to support independent living for people with chronic or degenerative limitations in motor and/or cognitive abilities. Statistics Canada states that over 2.4 million people living with disabilities receive regular assistance with at least one Activity of Daily Living (ADL) on a daily basis, with immediate family members most commonly identified as the primary caregiver. It has been shown that the number of ADLs requiring assistance is correlated with the severity of the disability [18]. Over the past decade there has been an increasing demand for healthcare services due to the rising elderly and disabled population; assistive robots can help bridge this gap by alleviating the labour burden for healthcare specialists and caregivers.

V. TRACKING

Precise and reliable visual servoing is dependant on the underlying tracking software implemented. Visual servoing relies on visually tracking the set task feature points that compose the error function, $e(t) = s - s^*$, to be minimized. Abhineet Singh and Martin Jagersand [3] released a Modular Tracking Framework (MTF), for registration based tracking (RBT), specifically targeted at robotics applications. The authors show that a tracking task can be decomposed into three modules: the optimization approach, the similarity metric, f, and the warping function, w. The Modular Tracking Framework categorizes these as the Search Method (SM), Appearance Model (AM), and State Space Model (SSM), respectively, which can each be specified at run-time, making the framework ideal for dynamically customizing to real-world applications.

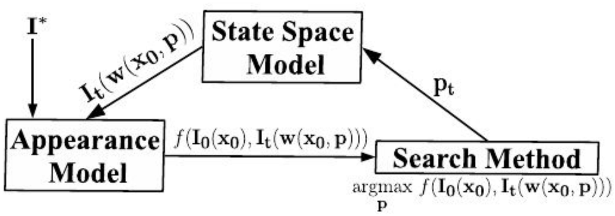

Fig. 3: Modular breakdown of a registration based tracker in the MTF framework [3]

The goal of registration based trackers is to find the optimal parameters, $p_t = [\ p_{st}, p_{at}\ ]$, which maximize the similarity metric, f, between the target patch $I^* = I_0(x)$ and the warped image patch $I^c = I_t(w(x, p_t))$, denoted:

$$\mathbf{p_t} = \underset{\mathbf{p}}{\text{argmax}}\ f(\mathbf{I}^*, \mathbf{I_t}(\mathbf{w}(\mathbf{x}, \mathbf{p}))) \qquad (10)$$

VI. OUR HUMAN ROBOT INTERACTION SYSTEM

We present a HRI system for semi-autonomous control by extending and simplifying the ViTa system created by Gridseth et *al.* [1]. This system keeps the human-in-the-loop by having the user set and publish geometric task constraints consistent with the concepts discussed in section III. Uncalibrated visual servoing and tracking are performed autonomously, thereby improving accuracy and repeatability, as well as reducing the cognitive load on the user, and the execution time. The Robot Operating System (ROS) was used to facilitate system integration between the user interface, tracking framework, and visual servoing. Our HRI system utilizes five separate nodes: the user interface, tracking, error control, error grouper, and uncalibrated visual servoing, which communicate vital information and handle distinct steps in the HRI system. Furthermore, it can be run as either a single camera system, or stereo vision system, depending on user preference and desired tasks.

The first and primary node is the User Interface (UI). The UI is a touchscreen device by which the user interacts with the system through task setting, decision making, and also oversee the entire process. Upon activation, the user is prompted to choose the type of task they would like to set. The user is then guided through each step of the process with simple instructions and prompted questions. Once the user is satisfied with the set tasks, they have the option of sending the tasks to be autonomously completed. Throughout the process, the user has the choice of resetting any current tasks being set, or wiping all tasks from system memory. The UI node is primarily written in python, utilizing the QT libraries integrated with ROS (RQT). Each subpart of the HRI system is presented to the user on this interface in a way designed to be simple to use as well as be aesthetically pleasing.

The second node is a group of nodes that runs through the ROS mtf_bridge package provided by MTF [3]. This group of nodes handles camera initialization, image buffers, and registration based tracking. Once tasks are set and the user chooses to publish them, these tracking nodes handle all tracker updates, resets and deletions. The tracking nodes are written in C++ in order to fully utilize the MTF libraries which are written in the same language.

The third node runs the error control processes, where encoded task functions are set, and corresponding errors published Fig. 5: ROS nodes in our HRI system real-time as tracker updates are received from the tracking node. These errors are sent to the fourth node where they are grouped according to error dimensions, and stacked to form the global encoded task function (9). The error control node is implemented in C++ in order to utilize the Eigen library for fast and reliable calculations [17]. The final node handles task execution and error minimization through ROS-UVS provided by Oscar Ramirez and Martin Jagersand [4] .

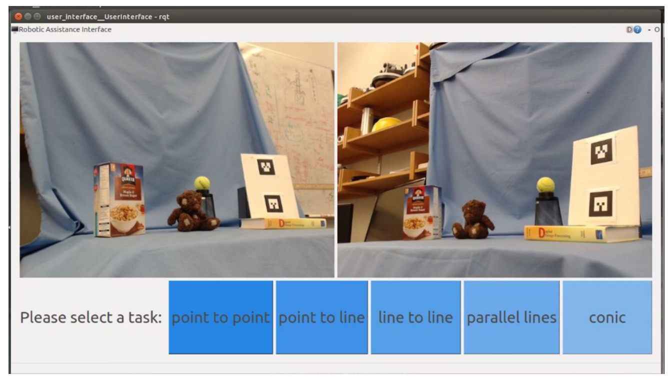

Fig. 4: Our user interface

VII. TASKS AND ERROR CONTROL

In this section we give a mathematical representation of each available task in our HRI system in accordance with the terminology laid out in section III.

Point-to-point task

Let T_{pp} be the task function defined on $F_{pp} := V^2$ by the rule:

$\{f_1, f_2\} \mapsto 0$ if f_1 and f_2 are the same point in 3D projective space

$\{f_1, f_2\} \mapsto 1$ otherwise.

Let E_{pp} be the encoded task function defined on T_{pp}, denoted:

$$E_{pp}(f_1, f_2) = f_2 - f_1 \quad (11)$$

where E_{pp} returns the 2D error vector to be minimized.

Point-to-line task

Let T_{pl} be the task function defined on $F_{pl} := V^3$ by the rule:

$\{f_1, f_2, f_3\} \mapsto 0$ if f_1 lies on the line created by f_2 and f_3

$\{f_1, f_2, f_3\} \mapsto 1$ otherwise.

Let E_{pl} be the encoded task function defined on T_{pl}, denoted:

$$E_{pl}(f_1, f_2, f_3) = f_1 \bullet (f_2 \times f_3) \quad (12)$$

where E_{pl} returns the 1D error vector to be minimized.

Line-to-line task

Let T_{ll} be the task function defined on $F_{ll} := V^4$ by the rule:

$\{f_1, f_2, f_3, f_4\} \mapsto 0$ if f_1 and f_2 lie on the line created by f_3 and f_4

$\{f_1, f_2, f_3, f_4\} \mapsto 1$ otherwise.

Let E_{ll} be the encoded task function defined on T_{ll}, denoted:

$$E_{ll}(f_1, f_2, f_3, f_4) = [f_1 \bullet (f_3 \times f_4), \; f_2 \bullet (f_3 \times f_4)] \quad (13)$$

where E_{ll} returns the 2D error vector to be minimized.

Parallel lines task

Let T_{par} be the task function defined on $F_{par} := V^4$ by the rule:

$\{f_1, f_2, f_3, f_4\} \mapsto 0$ if the line created by f_1 and f_2 is parallel with the line created by f_3 and f_4

$\{f_1, f_2, f_3, f_4\} \mapsto 1$ otherwise.

Let E_{par} be the encoded task function defined on T_{par}, denoted:

$$E_{par}(f_1, f_2, f_3, f_4) = (f_1 \times f_2) \times (f_3 \times f_4) \quad (14)$$

where E_{par} returns the 1D error vector to be minimized.

VIII. CONCLUSION

This work amalgamates selected contributions made in Human Robot Interaction [1], registration based tracking [3], task specification [2], and uncalibrated visual servoing [4], into a user-centered interface for task execution. Through the use of a semi-autonomous HRI system, the user is able to remain in full control and oversee each step of the process, while the cognitive load is reduced in comparison with solely teleoperated systems.

IX. FUTURE WORK

In our future work, incorporating predefined tasks as an additional layer of the interface could yield pertinent results. This would involve a thorough study of the activities of daily living to determine which tasks are of high value. Subsequently, these tasks would need to be defined in a way that still allows for task execution in a dynamic, unstructured environment. This additional layer would provide a way for the user to select and execute highly-repetitive daily tasks with minimal effort.

X. REFERENCES

[1] Mona Gridseth, Oscar Ramirez, Camilo Perez Quintero, and Martin Jagersand. Vita: Visual task specification interface for manipulation with uncalibrated visual servoing. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3434-3440. IEEE, 2016.

[2] Z. Dodds, G. D. Hager, A. S. Morse and J. P. Hespanha, "Task specification and monitoring for uncalibrated hand/eye coordination," *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, Detroit, MI, 1999, pp. 1607-1613 vol.2.

[3] A. Singh and M. Jagersand. Modular Tracking Framework: A Unified Approach to Registration based Tracking. *CoRR*, abs/1602.09130, 2016.

[4] O. A. Ramirez and M. Jagersand, "Practical Considerations of Uncalibrated Visual Servoing," *2016 13th Conference on Computer and Robot Vision (CRV)*, Victoria, BC, 2016, pp. 164-169.

[5] M. Gridseth, C. Perez Quintero, R. Tatsambon Fomena, O. Ramirez, M. Jägersand, "Bringing visual servoing into real world applications", *Human Robot Collaboration Workshop at RSS*, 2013.

[6] A. Vaezipour, A. Rakotonirainy, N. Haworth, P. Delhomme, 2017. Enhancing eco-safe driving behaviour through the use of in-vehicle human-machine interface: A qualitative study. *Transportation Research*

Part A: Policy and Practice. Vil. 100, pp. 247-263.

[7] Agin, G.J., "Real Time Control of a Robot with a Mobile Camera". Technical Note 179, SRI International, Feb. 1979.

[8] B. Espiau, F. Chaumette and P. Rives, "A new approach to visual servoing in robotics," in *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313-326, Jun 1992.

[9] W. J. Wilson, C. C. Williams Hulls and G. S. Bell, "Relative end-effector control using Cartesian position based visual servoing," in *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684-696, Oct 1996.

[10] A. m. Farahmand, A. Shademan and M. Jagersand, "Global visual-motor estimation for uncalibrated visual servoing," *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, 2007, pp. 1969-1974.

[11] A. Shademan, A. m. Farahmand and M. Jägersand, "Robust Jacobian estimation for uncalibrated visual servoing," *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK, 2010, pp. 5564-5569.

[12] C. P. Quintero, R. Tatsambon, M. Gridseth and M. Jägersand, "Visual pointing gestures for bi-directional human robot interaction in a pick-and-place task," *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Kobe, 2015, pp. 349-354.

[13] A. Edsinger and C. C. Kemp, "Human-Robot Interaction for Cooperative Manipulation: Handing Objects to One Another," *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, Jeju, 2007, pp. 1167-1172.

[14] Michael A. Goodrich; Alan C. Schultz, "Human-Robot Interaction: A Survey," in *Human-Robot Interaction:A Survey* , 1, Now Foundations and Trends, 2008.

[15] K. Stubbs, P. J. Hinds and D. Wettergreen, "Autonomy and Common Ground in Human-Robot Interaction: A Field Study," in *IEEE Intelligent Systems*, vol. 22, no. 2, pp. 42-50, March-April 2007.

[16] M. Jagersand, O. Fuentes and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," *Proceedings of International Conference on*

Robotics and Automation, Albuquerque, NM, 1997, pp. 2874-2880 vol.4.

[17] “Eigen: A C++ template library for linear algebra,” http://eigen.tuxfamily.org, accessed: 2018-05-01.

[18] “Living with disability series. Help with activities of daily living for people with a disability,” https://www.statcan.gc.ca/pub/11-008-x/2010002/article/11345-eng.htm#n1, accessed:2018-05-01.