The MacGyverBot: Tool Creation by Autonomous Agents

Lakshmi Nair
Georgia Institute of Technology
lnair3@gatech.edu

Jonathan Balloch
Georgia Institute of Technology
balloch@gatech.edu

Sonia Chernova
Georgia Institute of Technology
chernova@cc.gatech.edu

***Abstract*—This work explores the problem of Macgyvering in robotic systems. "MacGyvering" is defined as creating or repairing something in an inventive or improvised way by utilizing objects that are available at hand. Macgyvering poses a significant challenge since it requires robots that can simultaneously reason about the environment, objects, task goals and the robot's own capabilities. In this paper, we propose 6 levels of Macgyvering that outline a spectrum of complexity within this problem domain. We further discuss a computational framework for Level 1 of the spectrum which involves the creation of tools from available object parts. We implement our approach on a robotic platform and present our preliminary results.**

I. Introduction

Intelligence is often best expressed through ingenious problem solving, which has been a challenge for robots, but a skill that humans depend on, especially in high-stress or time constrained scenarios. The Apollo 13 incident of 1970 provides numerous examples of how human resourcefulness and creativity helped save the lives of the 3 astronauts on board [9]. To combat the increasing carbon dioxide levels within the cabin, the crew fashioned a contraption for filtering carbon dioxide out of a sock, a plastic bag, book covers and duct tape, knowing only the properties of the available parts and the goal. Our work envisions a robot that is co-located with humans like these astronauts, capable of providing assistance with similarly imaginative solutions. In our work, we explore the problem of robot tool creation for the purpose of problem solving, or *Macgyvering*. While tool-use is a widely explored problem within robotics, tool creation has received much less attention. To the best of our knowledge, this is the first work that proposes a computational framework for tool Macgyvering with its implementation on a robotic platform.

Tools are defined as objects that extend the physical influence of the agent or considered as extensions of the agent [18]. Existing work in Psychology has shown that tool creation emerges relatively late in children as compared to tool-use [6]. Their work posits that the actual tool creation in children is preceded by the step of analyzing the problem and imagining the tool suitable for the task. On this basis, we propose the following high-level steps involved in Macgyvering:

- **Problem Identification:** This involves identifying the problem and the desired effect the agent would like to accomplish - for example, a screw needs to be tightened;
- **Reference Tool Identification:** Identifying a suitable tool for accomplishing the desired effect - for example, screwdriver;
- **Tool Macgyvering:** If the reference tool is unavailable, a tool that accomplishes the same effect needs to be created from available parts
- **Tool Evaluation:** This involves attempting to solve the problem with the Macgyvered tool and in case of failures, iterating over the process to create the next best tool.

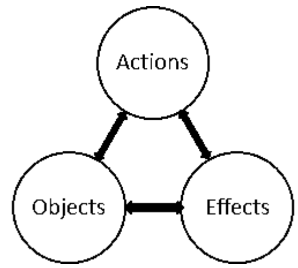


Fig. 1: Affordances described as a relation between objects, actions and effects.

The steps detailed above indicate the importance of affordances in Macgyvering. Affordances are the action possibilities available to the agent for a given object [16]; affordances function by priming specific actions for the user by virtue of the object's physical properties. We seek to combine objects and create tools that have certain affordances/produce certain effects. The Macgyvered tool can act as a replacement for a known item (eg. Macgyver a hammering object if hammer is unavailable) or can be an entirely new combination of items (eg. the contraption used on Apollo 13). Combinations of such items can be decomposed into their individual parts, each acting as a replacement for a known item eg. spacesuit tubes serve as replacement for pipes. In this manner, the latter problem can be decomposed to that of combining replacements for known items.

In the following sections, we discuss the different levels of Macgyvering and position our work with respect to existing literature. We then provide a detailed discussion of our approach and conclude with the results and future directions for our work.

II. Levels of Macgyvering

Affordances are often computationally defined as a relationship between objects, actions and effects [20] (Figure 1). We

Retained knowledge	No MacGyver	Level 0	Level 1	Level 2	Level 3
Object	Reference object	Object substitution	Object construction	Object substitution	Object construction
Action	✓	✓	✓	Action adaptation	Action adaptation
Effect	✓	✓	✓	✓	✓

TABLE I: Different Levels of Macgyvering - check marks indicate the knowledge retained across the spectrum

use this definition as a basis for introducing a spectrum that captures the varying degrees of Macgyvering complexity.

We define a *reference tool* as the "imagined" tool deemed suitable for the task at hand. This is analogical to the initial steps of tool-making as described in the literature [6]. For example, screwdriver is the reference tool for tightening screws. No Macgyvering is required if the reference tool is directly available to the agent. In other cases, we introduce the following 6 levels of Macgyvering (Levels 0-3 also shown in Table I):

- **Level 0:** The task goal/sub-goal, the effect and the action for accomplishing it are all known, but the system needs to identify a substitute for the reference tool. Eg. Knife as substitute for screwdriver to "engage" and tighten a screw. This is a simple case of object substitution.
- **Level 1:** The task goal/sub-goal, the effect and the action for accomplishing it are all known, but there is no direct substitute available. Thus, the robot needs to construct a new tool that accomplishes the effect with the same action. Eg. A clothespin and a coin combined to function as a screwdriver to "engage" and tighten a screw.
- **Level 2:** The goal/sub-goal and the effect to be accomplished are same as above, but the action itself is different. Hence, a new action (action adaptation) *and* tool for accomplishing the desired effect must be identified. Eg. Pliers can be used to "grasp" and turn a screw instead of the "engage" and turn of a screwdriver. The effects of both actions are the same, namely, tightening the screw.
- **Level 3:** The goal/sub-goal and the effect to be accomplished are retained but with action adaptation. Additionally, a tool should be constructed to accomplish the desired effect. Eg. combine a spring and two sticks to function as pliers to "grasp" and turn screws. This is useful when the available objects cannot be combined to create a screwdriver-like tool.

The more challenging extensions of these levels are as follows:

- **Level 4:** The goal/sub-goal is same as previous levels, but the effect *and* action are different. This involves identifying a tool that accomplishes a new effect but the same goal/sub-goal. Eg. Using rope to attach parts rather than tightening screws.
- **Level 5:** The goal/sub-goal is retained but the effect and action towards accomplishing it are different. Additionally, a tool must be constructed. Eg. Creating a rope from strands of yarn.

Level 0 Macgyvering which is a case of object substitution, has been previously tackled in the robotics literature [1], [4]. In this paper, we focus on Level 1 Macgyvering which involves the construction of tools from individual parts.

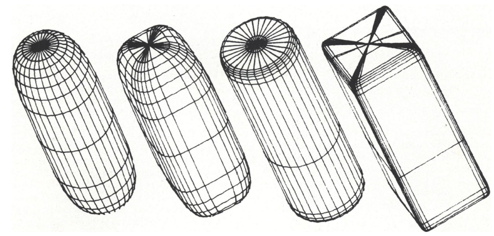


Fig. 2: Examples of Superquadrics: Superellipsoids with varying parameters - Figure from [5]

III. Related Work

A. Tool Creation in Animals

Tool-making and use in animals has been widely studied. Animals such as chimpanzees [8], bonobos [27] and birds such as rooks [7] and blue jays [18] have been shown to create and use simple tools to retrieve food. These are often cited as examples of higher order cognition. In [6], the authors identify two key aspects of tool-making: "tool manufacturing" and "tool innovation". Tool manufacturing refers to the physical transformation of the materials and is a necessary step in the tool-making process. Whereas tool innovation, which refers to imagining the type of tool suitable for the task, may not always be involved. Our approach which begins by "imagining" a reference tool followed by recreating it with objects that were not seen before, thus tackles the combined problem of tool innovation and tool manufacturing.

B. Object Substitution (Level 0 Macgyvering)

In this work we represent the tools using Superquadrics [5]. Superquadrics (SQ) refer to the family of geometric shapes that resemble quadrics but with the squaring operations replaced by arbitrary powers. Fig 2 shows examples of Superellipsoids, one class of SQ. Prior research has looked at modeling of objects using SQs [10] and the SQ representation of objects for pose recovery [13]. More recently, work by [1] has looked at SQ fitting of tools for tool substitution. We draw inspiration from their approach and use SQ fitting to represent the reference tool. However, rather than identifying substitute tools, we use SQs to guide the selection and attachment of geometrically appropriate pieces for creating the reference tool. Other approaches such as [24] have also tackled the problem of identifying uncommon uses of tools using part-based analysis.

C. Macgyvering in Robots

While there hasn't been tremendous work within the area of Robotic Macgyvering, some existing research has looked at the use of environmental objects for solving tasks [25, 19]. Their work involved reasoning about the use of environmental objects as simple machines eg. levers. More recently, Choi et al. [11] proposed a framework for creating and using hybrid

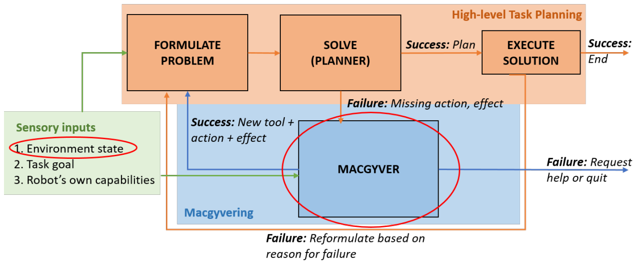


Fig. 3: Proposed Computational Model for a Macgyvering System: The highlighted aspects of this model are the focus of this paper. The Macgyvering module in this work implements Level 1 Macgyvering

tools within the cognitive architecture ICARUS. While the high-level goal of our respective works coincide, we take a more vision-based perspective to tackling the problem and present our framework on a robotic platform. However, we believe that a combination of the two approaches may be essential for a good MacgyverBot.

Additionally, recent work has also proposed the "Macgyver Test" for assessing the resourcefulness of robots [23]. This test proposes to assess the creativity of robots in problem solving, Macgyvering being a key example of human creativity.

IV. Overview of the High-Level Pipeline

In this section, we discuss the overview of our framework to highlight how Macgyvering fits in with the bigger picture of task planning. Shown in Figure 3 is our high-level process flow consisting of 3 main subsystems. Highlighted in red are the parts discussed in this paper. The 3 subsystems are:

- **Sensory Inputs**: This subsystem collects sensory inputs including information about the environment (objects and robot states), the task goal and the robot's own capabilities.
- **High-Level Task Planning**: This subsystem deals with the high-level task planning. This involves formulating the planning problem which may involve generation of domain and problem definitions followed by attempting to solve the problem. There are several existing approaches to high-level task planning ([26], [21]) that can be applied here. In cases where certain objects are required to complete the task, eg. the hammer action requires an appropriate object, their absence causes the solver to fail. Identifying the cause of failure in terms of missing action/effect can be achieved using approaches such as [17] and [28] on plan repair. This information guides the Macgyvering process.
- **Macgyvering**: This subsystem is activated in the case of a solver failure. The Macgyvering module (Figure 4) reasons about the missing action/effect and identifies if any prior known tools allow the desired action/effect to be accomplished (i.e, has the desired affordance). In such cases, the module proceeds to Macgyver the object. Once the tool is created, the planning domain can be updated with the information and the Planning system can attempt to replan for the task.

We explain our high level framework in the context of an example. The robot wishes to escape a building by constructing a staircase to reach an elevated area leading to the exit. The task planner, takes in all the relevant information and attempts to plan for the escape. However, the planning fails since the robot is unable to reach the elevated area. This missing action or effect is passed into the Macgyvering module. The module then outputs a solution that involves constructing a staircase using blocks of wood available to the robot. After constructing the staircase, the updated information allows the planner to re-plan and accomplish the task.

This paper focuses on how the Macgyvering module accomplishes the tool creation. Our future work looks at integrating it with the remainder of the high level framework including reasoning about planning failures and subsequent re-formulation of the problem.

V. Macgyvering Module: Level 1 Macgyvering

In this section, we discuss our implementation of Level 1 Macgyvering. Inspired by existing literature on tool-making, there are two key aspects incorporated into our system:

- **Reference Tool**: As described in [6], tool-making involves the prior step of imagining the appropriate tool for a task and we incorporate this idea using the reference tool that guides the Macgyvering process.
- **Generate and Test Paradigm**: Tool-making in children develops through trial-and-error approaches before the age of 7, after which they are able to construct tools from anticipation [6]. Hence, Maccgyvering can be considered an iterative process involving generation and testing of different solutions, starting with the best evaluated solutions. Such approaches are often cited in the Computational Creativity literature [3], wherein the system first generates an artifact and then evaluates its "goodness" via an iterative process that yields a satisfactory artifact. We use a similar approach where the robot tests different possibilities and iterates on the design if needed.

Macgyvering may involve reasoning about various aspects of objects such as geometric and material properties. In this paper we focus on reasoning using geometric properties of objects and formulate our problem as follows:

"Given a reference tool and a set of environmental objects, can the robot construct a substitute tool by reasoning using geometric properties?"

Figure 4 shows the complete Macgyvering module for tackling this problem. In addition to using SQ fitting to represent tools, our work involves reasoning about *attachment points*. These refer to points where objects can be attached to one another. In most cases of Macgyvering, humans often use Duct tape or glue to put together pieces. However, certain objects such as bottles may provide natural attachment points at the bottle opening where a suitable sized object can be inserted. In our work, we use magnets to emulate attachment points.

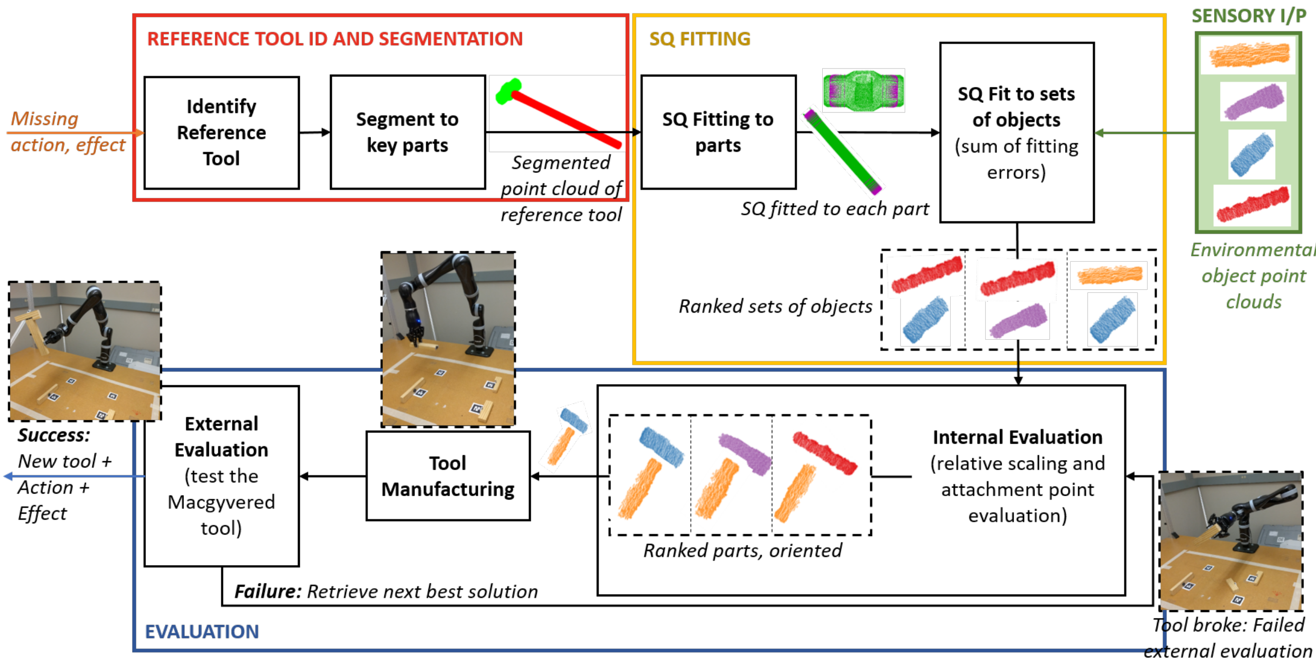


Fig. 4: Computational pipeline for the Level 1 Macgyvering module: The example shows construction of a hammer. Based on the missing action/effect, a reference tool is identified, followed by SQ fitting to object parts and finally the internal and external evaluation of the tool.

We now discuss the different steps involved in the Macgyvering module in the following sections.

A. *Reference Tool Identification and Segmentation*

The first step involves identifying a reference tool that helps accomplish the task. We currently provide this mapping from the desired affordance to the appropriate tool, although approaches such as AffordanceNet [12] can accomplish this. Once the reference tool is identified, we obtain its segmented point cloud representation. In our case, we directly use the ToolWeb dataset [2] which contains 116 pre-segmented point clouds of various household tools and objects. The segmented point cloud of the reference tool is relayed to the next step.

B. *Superquadric Fitting*

For SQ fitting, we use the approach detailed in [1]. We fit SQ separately to each of the reference tool segments using Levenberg-Marquardt optimization. The SQs are modeled using 13 parameters: 3 for scale in each dimension, 2 for shape variance, 3 for Euler angles, 2 for tapering parameters and 3 for the central point/mean. For optimization, we use the inside-outside function, $F(x, y, z; \lambda)$ where λ is the set of parameters. For points inside the SQ, $F < 1$; for points on the SQ, $F = 1$ and $F > 1$ for points outside the SQ. This results in the following minimization problem:

$$\arg\min_{\lambda} \sum_{i=1}^{N} (F(x, y, z; \lambda) - 1)^2$$

This optimization is repeated for different classes of SQs and the SQ with the lowest residual error for each part of the reference tool is chosen as the best representation for that part (superellipsoids are the best fit for both hammer handle and head as shown in Fig 4).

Following SQ fitting for the reference tool, our approach retrieves the point clouds corresponding to the environmental objects. We use RGBD sensors and perform plane subtraction and segmentation of the input point cloud to identify the point cloud segments corresponding to each object (Fig 4 Sensory I/P). Subsequently, it fits the classes of SQs representing the reference tool parts (superellipsoids in this case) to each of them. Each of the parts are evaluated for their match to each segment of the reference tool. For n objects and k tool segments, this leads to $k * n$ fittings. Hence, an implicit assumption of our approach is that the number of object parts required equals the number of reference tool segments. In the case of tools, $k = 2$ since most tools only have an action and a grasp part. We use O to denote the pairs/sets of objects being considered for each part of the tool, i.e, $O = o_1, o_2 ... o_k$. This leads to $k * \binom{n}{k}$ possibilities for O. The total residual fitting error associated with any given O is computed as sum of the residual errors of its components:

$$e_{residual}(O) = \sum_{i=1}^{k} res(o_i)$$

Where, $res(o_i)$ refers to the residual fitting error of object o_i. This computation is performed for all the possible sets of objects O and passed on for subsequent evaluation.

C. *Evaluation*

The evaluation module involves both intrinsic/internal and extrinsic/external evaluation. The considerations here are:

- Are the objects appropriately shaped?

- Are they proportional in scaling when compared to the reference tool proportions?
- Do they have attachment points facilitating the desired attachment?
- Does the tool work in the real world?

1) Intrinsic Evaluation: This evaluation deals with the first 3 considerations. The system first computes the relative scale ratios between different segments of the reference tool. A good set of candidate parts should have a similar scale ratio. For eg, the handle of a hammer may be twice as long as the hammer head. To encode this aspect, we compute the ratio of the scales for pairs of reference tool segments along each dimension and sort it, resulting in a 3D vector denoted by $rel(ref)$). Sorting removes dependency of the ratio on the specific dimension. We repeat the process with the objects $o_i \in O$ to obtain $rel(O)$ and compute the norm of the difference between $rel(ref)$ and $rel(O)$:

$$e_{ratio}(O) = \|rel(ref) - rel(O)\|$$

Next, the system retrieves the relative SQ orientations between the different segments of the reference tool. The system also retrieves the positions of the attachment points of the candidate point clouds (indicated by AR tags) relative to their mean/centroid. Each object point cloud in the set O is oriented according to the relative orientations of the reference tool parts (Fig 4 Internal Evaluation). The system then checks for the distance of the closest attachment point from the point cloud attachment location, denoted by $AttDist(o_i)$. This indicates a measure of whether there are attachment points near the desired attachment location:

$$e_{att}(O) = \sum_{i=1}^{k} AttDist(o_i)$$

In effect, this checks if any attachment points on the objects facilitate their desired attachment and score them accordingly. The final total error for each set of parts is computed as the sum total of all the different errors:

$$e_{final}(O) = e_{residual}(O) + e_{att}(O) + e_{ratio}(O)$$

The set of pieces with the lowest e_{final} is chosen as the best set of parts to combine along with their corresponding attachment points.

While the system makes the best judgment based on the reference tool, the results may be prone to errors resulting in incorrect part selection. Additionally, the physical object attachments may be weak and cannot be encoded in the intrinsic evaluation. Thus, it would be beneficial to have the robot evaluate the output by physically constructing and attempting to use the tool in the real world.

2) Extrinsic Evaluation: Given the set of parts and their attachment points chosen by intrinsic evaluation, the robot manufactures the tool by joining the pieces. The robot then evaluates the constructed tool for its task suitability by applying the desired action on the tool eg. in the case of Macgyvering a hammer, the robot attempts to hammer with the new object. We assume that the robot is provided with the appropriate action trajectory for using the object although it can be learned through demonstrations [22] or even adapted from a source trajectory to fit the modified tool ([14], [15]). When performing the action with the constructed tool, if the tool breaks or fails to accomplish the task, the system identifies the next appropriate set of objects O based on e_{final} and creates a new tool for testing. Thus, the cycle between intrinsic and extrinsic evaluation allows the robot to explore and evaluate different tool-making possibilities.

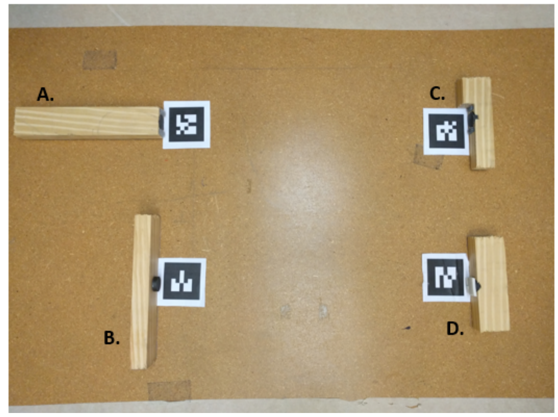
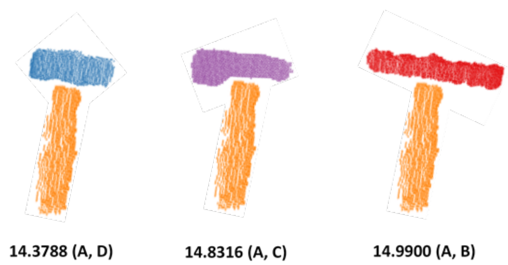
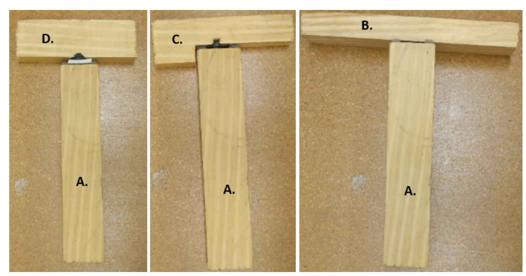


Fig. 5: Robot workspace showing the 4 different pieces (A. through D.) each with an attachment point. The AR tags are only used to identify the attachment point locations.



(a) The 3 best outputs of internal evaluation along with the corresponding parts and e_{final} values



(b) Corresponding actual hammer constructions

Fig. 6: Left to right are the highest to lowest ranked close matches for a hammer based on internal evaluation

VI. Results

In this section we demonstrate our framework on a robotic platform. We created an experimental setup using 4 different object parts each with an attachment point as shown in Fig 5. These parts allow for 12 different potential part combinations,

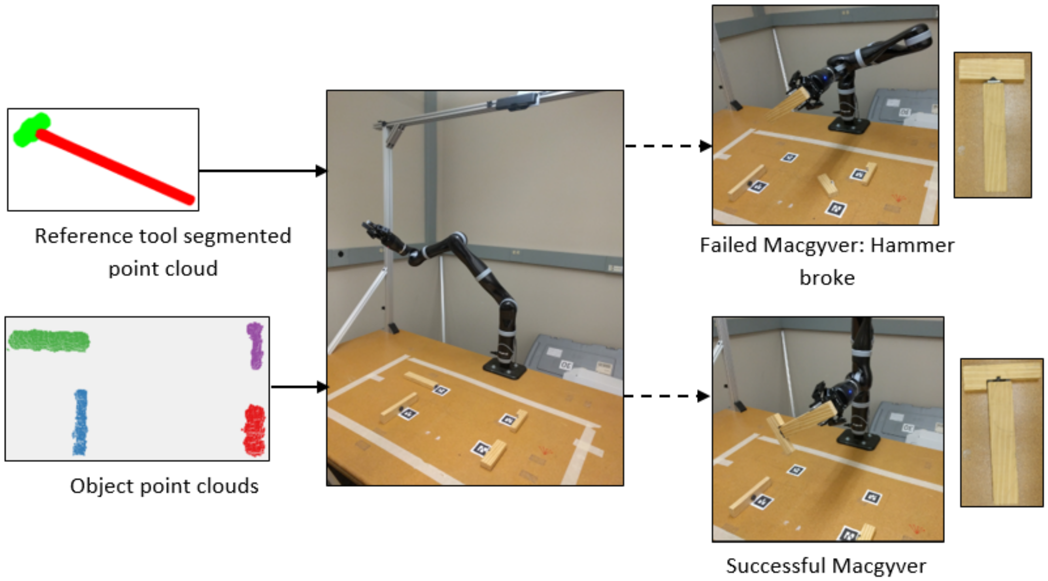


Fig. 7: Experimental setup of the robot: The reference tool point cloud was provided. The robot observed the objects in its environment (object point clouds) and constructed a hammer. The first construction failed (top right) and the robot created a different hammer which succeeded the hammer test (bottom right). The Macgyvered objects are shown alongside.

3 of which closely resemble a hammer. Fig 6 shows the ranked part combinations along with their corresponding e_{final} values. Of the 3, one closely fits both the shape and size but has a weak attachment point causing it to break when attempting to hammer (left); The middle pair is not a close geometric match for the hammer head since it has a protrusion on that piece, but the pair match closely in terms of relative ratio of scales and also have a strong attachment point; the third one does not closely match the size proportions and has a weak attachment. Hence, from a pure geometric perspective, the first hammer is a better fit, but with respect to task execution as well, the second hammer is most appropriate. Thus, as expected, internal evaluation identifies the former as the best candidate which is only pruned by the external evaluation of the tool.

Shown in Fig 7 (middle) is the complete experimental setup. The robot was provided with a segmented 3D scan of a hammer which served as the reference tool. After observing the different object point clouds, our algorithm identifies the best pair of objects to use. The system then relays this information to the robot which proceeds to connect the individual parts to create the hammer shown in Fig 6 (leftmost) using parts A and D. It then proceeds to test the Macgyvered hammer which breaks due to the weak attachment point Fig 7 (top right). The robot then creates the next best solution shown in Fig 6 (middle) using parts A and C which has a strong attachment point. Upon testing the hammer, it retains its stability and the robot is able to execute the hammering action (Fig 7 - bottom right). Once a successful solution is found, the robot can proceed with the remainder of the task.

VII. Conclusion and Future Work

In this paper, we've discussed the problem of Macgyvering in robots to increase their resourcefulness. We introduced a spectrum of complexity highlighting the different levels of Macgyvering and presented a computational framework that allows the robot to reason about and construct tools from individual pieces. While our work focused on tools, the approach is applicable to non-tool objects as well since the reference object used in Macgyvering does not have to be a tool. Similarly, testing such objects involve checking whether they accomplish the desired effect. For eg. a Macgyvered shelter should protect from the rain.

Macgyvering presents a challenging problem for robots and numerous issues remain to be addressed. Our future work aims to tackle some of the problems such as:

- Integrating the Macgyvering module with high-level task planning;
- Reasoning about the task goals and robot capabilities when creating new objects;
- Reasoning about other aspects of the domain and material properties of objects;
- Implementing the additional levels of Macgyvering and reasoning about the appropriate level for a given problem;

Acknowledgments

This work is supported in part by NSF IIS 1564080 and ONR N000141612835.

References

[1] Paulo Abelha, Frank Guerin, and Markus Schoeler. A model-based approach to finding substitute tools in 3d vision data. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2471–2478. IEEE, 2016.

[2] Paulo Abelha Ferreira and Frank Guerin. Learning how a tool affords by simulating 3d models from the web.

In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS 2017)*. IEEE Press, 2017.

[3] Margareta Ackerman, Ashok Goel, Colin G Johnson, Anna Jordanous, Carlos León, Rafael Pérez y Pérez, Hannu Toivonen, and Dan Ventura. Teaching computational creativity. In *Proceedings of the Eighth International Conference on Computational Creativity, ICCC, Atlanta*, 2017.

[4] Alejandro Agostini, Mohamad Javad Aein, Sandor Szedmak, Eren Erdal Aksoy, Justus Piater, and Florentin Würgütter. Using structural bootstrapping for object substitution in robotic executions of human-like manipulation tasks. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 6479–6486. IEEE, 2015.

[5] Alan H Barr. Superquadrics and angle-preserving transformations. *IEEE Computer graphics and Applications*, 1(1):11–23, 1981.

[6] Sarah R Beck, Ian A Apperly, Jackie Chappell, Carlie Guthrie, and Nicola Cutting. Making tools isnt childs play. *Cognition*, 119(2):301–306, 2011.

[7] Christopher D Bird and Nathan J Emery. Insightful problem solving and creative tool modification by captive nontool-using rooks. *Proceedings of the National Academy of Sciences*, 106(25):10370–10375, 2009.

[8] Christophe Boesch and Hedwige Boesch. Tool use and tool making in wild chimpanzees. *Folia primatologica*, 54(1-2):86–99, 1990.

[9] Stephen Cass. Apollo 13, we have a solution. *IEEE Spectrum On-line, 04*, 1, 2005.

[10] Laurent Chevalier, Fabrice Jaillet, and Atilla Baskurt. Segmentation and superquadric modeling of 3d objects. 2003.

[11] Dongkyu Choi, Pat Langley, and Son Thanh To. Creating and using tools in a hybrid cognitive architecture. 2018.

[12] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *International Conference on Robotics and Automation (ICRA)*, 2018.

[13] Kester Duncan, Sudeep Sarkar, Redwan Alqasemi, and Rajiv Dubey. Multi-scale superquadric fitting for efficient shape and pose recovery of unknown objects. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4238–4243. IEEE, 2013.

[14] Tesca Fitzgerald, Ashok K Goel, and Andrea L Thomaz. Representing skill demonstrations for adaptation and transfer. In *AAAI Symposium on Knowledge, Skill, and Behavior Transfer in Autonomous Robots*, 2014.

[15] Pawel Gajewski, Paulo Ferreira, Georg Bartels, Chaozheng Wang, Frank Guerin, Bipin Indurkhya, Michael Beetz, and Bartlomiej Sniezynski. Adapting everyday manipulation skills to varied scenarios. *arXiv preprint arXiv:1803.02743*, 2018.

[16] James J Gibson. Perceiving, acting, and knowing: Toward an ecological psychology. *The Theory of Affordances*, pages 67–82, 1977.

[17] Kristian J Hammond. Explaining and repairing plans that fail. *Artificial intelligence*, 45(1-2):173–228, 1990.

[18] Thony B Jones and Alan C Kamil. Tool-making and tool-using in the northern blue jay. *Science*, 180(4090): 1076–1078, 1973.

[19] Martin Levihn and Mike Stilman. Using environment objects as tools: Unconventional door opening. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2502–2508. IEEE, 2014.

[20] Manuel Lopes, Francisco S Melo, and Luis Montesano. Affordance-based imitation learning in robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1015–1021. IEEE, 2007.

[21] Christopher Lörken and Joachim Hertzberg. Grounding planning operators by affordances. In *International Conference on Cognitive Systems (CogSys)*, pages 79–84. Citeseer, 2008.

[22] Muhammad Asif Rana, Mustafa Mukadam, S Reza Ahmadzadeh, Sonia Chernova, and Byron Boots. Towards robust skill generalization: Unifying learning from demonstration and motion planning. In *Conference on Robot Learning*, pages 109–118, 2017.

[23] Vasanth Sarathy and Matthias Scheutz. The macgyver test-a framework for evaluating machine resourcefulness and creative problem solving. *arXiv preprint arXiv:1704.08350*, 2017.

[24] Markus Schoeler and Florentin Wörgötter. Bootstrapping the semantics of tools: Affordance analysis of real world objects on a per-part basis. *IEEE Transactions on Cognitive and Developmental Systems*, 8(2):84–98, 2016.

[25] Mike Stilman, Munzir Zafar, Can Erdogan, Peng Hou, Saul Reynolds-Haertle, and Gregory Tracy. Robots using environment objects as tools the macgyver paradigm for mobile manipulation. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2568–2568. IEEE, 2014.

[26] Jaeyong Sung, Bart Selman, and Ashutosh Saxena. Synthesizing manipulation sequences for under-specified tasks using unrolled markov random fields. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2970–2977. IEEE, 2014.

[27] Nicholas Toth, Kathy D Schick, E Sue Savage-Rumbaugh, Rose A Sevcik, and Duane M Rumbaugh. Pan the tool-maker: investigations into the stone tool-making and tool-using capabilities of a bonobo (pan paniscus). *Journal of Archaeological Science*, 20(1):81–91, 1993.

[28] Roman Van Der Krogt and Mathijs De Weerdt. Plan repair as an extension of planning. In *ICAPS*, volume 5, pages 161–170, 2005.