Protocol Evaluation for TTool

Summary

---

0 What is TTool

TTool is an open-source AR 3D pose detector specifically designed for tool heads employed in woodworking operations with power tools. It is a multi-step algorithm composed by:
- i) an AI-powered classifier to identify the current tool
- ii) a UI for the user to guess and adjust the initial approximated global pose of the tool head
- iii) a pose refiner
- iv) visual feedback from the user to confirm the good alignment of the tool head and its model

We decided to evaluate TTool in real workshop fabrication scenarios with a selected population of users and physical tools since the algorithm is highly influenced by a human-machine interaction component and specific fabrication-related conditions.

1 Objective

The goal of the evaluation is to assess the following:
- 1) the error for the estimated pose by TTool (position + rotation)
- 2) the efficiency in tracking the pose measured in operational time
- 3) the usability and user appreciation of the proposed UI for the human-machine interaction component of the developed algorithm

2 Evaluation methodology

To test the accuracy we will ask users to find the pose by using the TTool pipeline. Next, as a ground truth, the user will insert the tool in a mold as a perfect negative of the tool. The user will place the physical tool head to perfectly fit the cavity. This will allow us to export the current validated pose of the tool and compare it with the digital twin of the mold in the same coordinate space.

In addition, we are particularly interested in understanding the time efficiency and learning curves associated with the different tools available in TTool. Users will go through multiple sessions using various tools, and we'll capture how long each tool takes to complete tasks. This allows us to not only assess the performance of each tool but also to gauge whether users find it increasingly easier to use TTool over time.

We aim to evaluate the UI's usability and the overall satisfaction of users. Our evaluation strategy utilizes a usability questionnaire. The questionnaire covers the three main TTool's algorithm steps (Select Tool, Input Tool, Final Tool Pose Validation), and a survey on which physical tool was the easiest to perform TTool (Overview of the Tools). Aside from the questionnaire, we also gather open comments during user feedback sessions to capture more detailed perspectives and insights. Additionally, our assessment includes an AB testing strategy for Tool Selection, comparing both Machine Learning (ML) and Non-ML interfaces. Participants will engage with the UI, answer the questionnaire, and share their thoughts in the interviews. Our results will comprise quantitative ratings from the Likert scale, qualitative feedback from interviews, and findings from the AB test comparison of ML and Non-ML tool selection methods.

To resume the evaluation protocol:

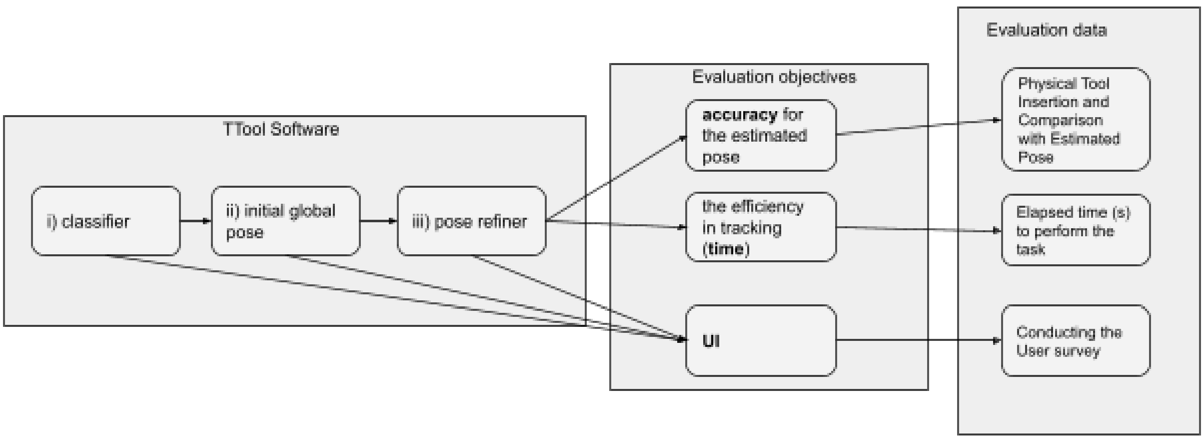

Fig.1 - Overview of the objectives and methodology for the evaluation protocol of TTool.

In the following chapters, we will define further the methodology.

2.0 Designated Interface

A touch 7inch display mounted on the tools has been selected as the main interface. This is an important limitation of the scope among all the possible interfaces that could have been used in the presented study. Although other head-mounted interfaces can be employed we decided to limit the scope to the current interface for the following reasons:

- i) more unusual interfaces, although more ergonomic (e.g. Hololens) they might require a higher digital literacy entry level to be used.
- ii) TTool is developed as a C++ console app and API for UNIX based systems. Developing and evaluating the software on a x64 machine and a connected touch display shorten considerably the research times.
- iii) a more ergonomic headset or phone require the use of a proprietary software (e.g. Unity) and some considerable time spend in code wrapping (e.g. C#, Java, etc) from the source code written in C++. A touch display allows us to evaluate the software while preserving the interoperability given by the low-level nature of TTool's source code (i.e. C++).

2.2 Selected tools

We limited the scope of the employed tools for the study. The selected range should showcase and represent the majority of the tools employed in an ordinary wood-working operation.
The toolset for participants includes 7 elements:

- a) Auger Drill Bits (of two varying diameters: 20 and 34 mm)
- b) Brad Point Drill Bit
- c) Self Feeding Bit
- d) Twist Drill Bit
- e) CircularSaw
- f) ChainSaw

The varying diameters in the point a) in the selected tools are introduced to test the system under the same typology of tools but with slightly different dimensions.

2.3 Population

We've chosen participants with a minimum of 3 years' experience in both the digital and construction fields. As digital experience we intend anyone that has interacted with a touch display. For the construction field we limited the criteria to qualify all persons which main occupation involve practical and physical tasks related to the construction industry (e.g. carpentry, metal working, masonry, etc)

2.4 Accuracy of the Estimated Pose

To assess the accuracy and duration required for TTool, we designed a mold with pre-fabricated cuts and holes (by CNC), serving as our ground truth. Users will be directed to position the tool within the designated cut or hole. This procedure allows us to export the current validated pose of the tool and compare it with the digital twin of the mold in the same coordinate space.

type of operations:
- For the cutting process, we intend to examine angle variations, specifically at 60 and 90 degrees, using two distinct saw types (circular saw and chainsaw).

- Regarding hole drilling, our primary emphasis is on exploring various drill types with differing diameters and lengths, hence the angle in which the drill bit is inserted in the panel has no impact.

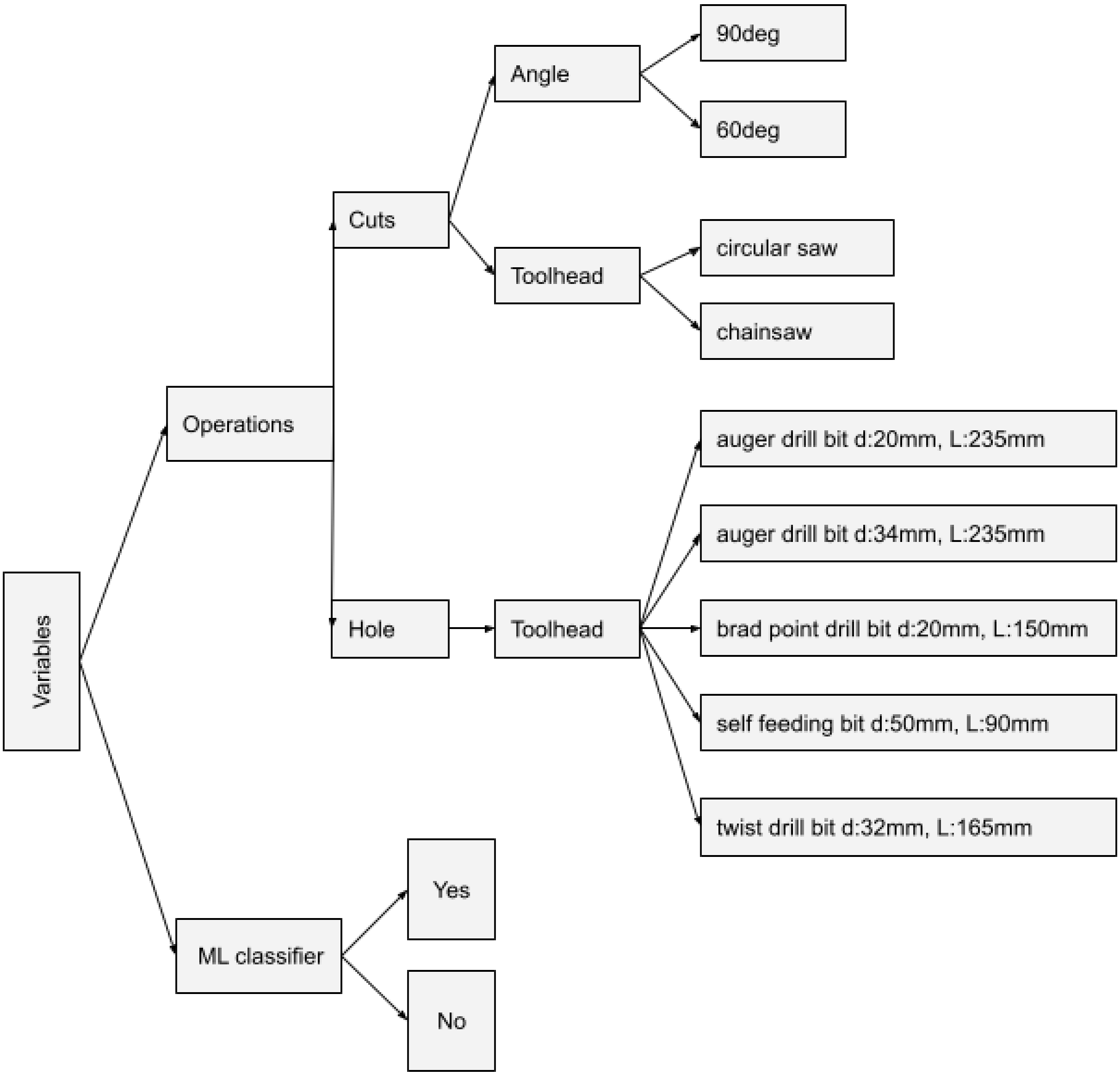

Fig.1 - Scheme of all variations of parameters selected for the study.

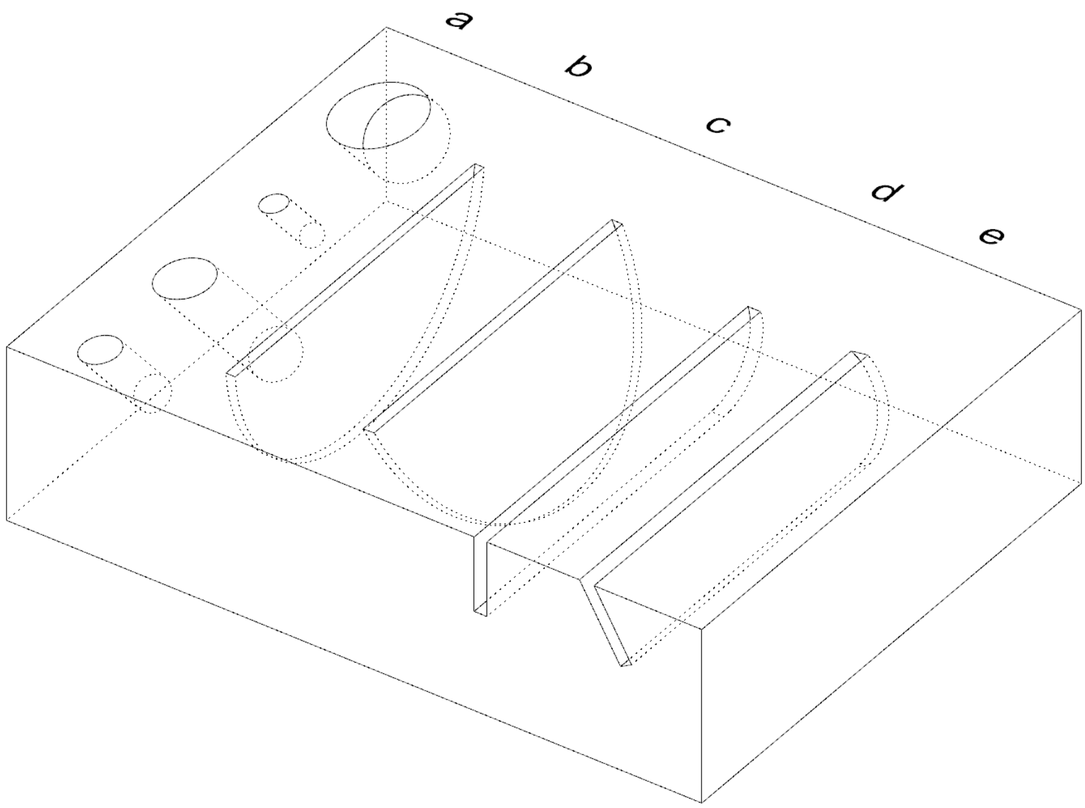

Fig.2 - An example of the ground truth mold where the user will insert the tool effectors once the TTool pose detection pipeline is completed: a) an example of the holes in the plate (given the chosen drill bits to test they will be more), b) 90 deg slot for circular saw, c) 60 deg slot for circular saw, d) 90 deg slot for chain saw, e) 60 deg slot for chain saw .

2.5 Time Efficiency

For time efficiency evaluation, we have multiple objectives:

- to measure the time efficiency associated with each individual tool within TTool;
- to gauge the impact of Machine Learning (ML) on the process;
- to investigate whether users experience a learning curve that makes it easier to use TTool over time.

Time metrics will be gathered in the following manner:

- Time tracking begins the moment a user is asked to start TTool and select a given tool—either manually or with ML assistance.
- Time tracking concludes when the final tool pose is validated and saved by the user (i.e. when inserted in the mold and the data exported).

The time duration for each user session, from tool selection to final pose validation, is measured in seconds.

2.6 UI Usability Assessment

In this chapter, we address the UI usability within the context of our evaluation.

The setup replicates a real-world environment with cuts and holes. Participants will stand with a physical tool in hand, complemented by the AC system.

Before they began, a quick 5-minute guide was provided to familiarize them with the process.

The 3 main operations the users will effectuate during the TTool are the following:

- Select Tool: users choose their desired tool. We've split the UI into two: one supported by machine learning which automatically identifies the tool, and a non-ML one where users make the choice themselves.
- Input Pose: once the tool is chosen, either through ML or manually, users must adjust its pose. This is because the AR tool's initial position might will notmatch the actual physical tool. By using the control buttons, users can make minor alignments to aid the refining process. The user will identify its level of satisfaction with the initial pose to step into the next pose refining phase.

Final Pose: users must examine the tool's pose before initiating the refiner. The TTool algorithm fine-tunes the alignment to match the physical tool more closely. Once satisfied, users should confirm and save the adjusted pose for future use.

For our UI usability assessment, we adopted a feedback approach inspired by the QUIS (Questionnaire for User Interface Satisfaction) format, using a 5-point Likert scale. After each practice session, users complete sections of the questionnaire covering Select Tool, Input Tool, and Final Tool Pose Validation. Once all rounds are finished, they provide feedback in the Overview of the Tools section.

The questionnaire provided to users is as follows::

	ML	Non-ML
SelectToolQuestionsML	How effectively did the design of the ML in UI minimize unnecessary steps or interactions during tool selection? 1 (Very Ineffective) / 2 / 3 / 4 / 5 (Very Effective) How would you rate the time efficiency of selecting tools using the ML-integrated UI? 1 (Very Time-Consuming) / 2 / 3 / 4 / 5 (Very Time-Efficient) How often would you choose to use the ML-integrated version of the UI for tool selection if you were to complete the same task regularly? Rate your preference: 1 (Always) / 2 (Mostly) / 3 (Sometimes) / 4 (Rarely) / 5 (Never)	Did you find the process of manually selecting tools straightforward or challenging compared to using the ML-integrated UI? 1 (More Challenging) / 2 / 3 / 4 / 5 (More Straightforward) How would you rate the time efficiency of manually selecting tools without the assistance of the ML model's classification? 1 (Very Time-Consuming) / 2 / 3 / 4 / 5 (Very Time-Efficient) How often would you choose to use the non-ML version of the UI for tool selection if you were to complete the same task regularly? Rate your preference: 1 (Always) / 2 (Mostly) / 3 (Sometimes) / 4 (Rarely) / 5 (Never)

lnputPoseQuestions	How easy was it for you to achieve the desired level of precision in aligning the AR objects with the real physical objects? Rate the ease of achieving precision 1 (Very Difficult) / 2 / 3 / 4 / 5 (Very Easy) How user-friendly and easy was the process of consistently adjusting the rotation? 1 (Very Difficult) / 2 / 3 / 4 / 5 (Very Easy) How user-friendly and easy was the process of consistently adjusting the translation? 1 (Very Difficult) / 2 / 3 / 4 / 5 (Very Easy) How did the time it took to align the AR objects with the real physical objects compare to your expectations? Rate the time comparison 1 (Much More Time-Consuming) / 2 / 3 / 4 / 5 (Much Quicker) Was it physically challenging for you to manipulate the tool while working with the AR system? Rate the physical challenge: 1 (Extremely Challenging) / 2 / 3 (Moderate) / 4 / 5 (Very Easy) How easy was it for you to maintain focus on the display and ensure the alignment of AR pose with real-world tool? Rate your ease of focus and alignment: 1 (Very Difficult) / 2 / 3 (Moderate) / 4 / 5 (Very Easy)
FinalPoseQuestions	How effective do you believe the refinement tool was in achieving the desired final pose of the AR objects? Rate the effectiveness of the refinement tool: 1 (Very Ineffective) / 2 / 3 / 4 / 5 (Very Effective) How often did you encounter challenges or uncertainty when deciding to start the refinement process for alignment enhancement? Rate the frequency of challenges: 1 (Very Rarely) / 2 / 3 / 4 / 5 (Very Often) How confident were you in determining when to stop using the refinement tool after achieving the desired alignment enhancement? Rate your confidence level: 1 (Not Confident at All) / 2 / 3 / 4 / 5 (Very Confident)
Tool Based	How easy was it to perform the task with the tool using TTool Overall, how would you rate your experience with an Auger Drill Bit (d:20mm, L:235mm) using the TTool? 1 (Very Negative) / 2 / 3 / 4 / 5 (Very Positive) Overall, how would you rate your experience with Auger Drill Bit

	(d:34mm, L:235mm) using the TTool? 1 (Very Negative) / 2 / 3 / 4 / 5 (Very Positive) Overall, how would you rate your experience with Brad Point Drill Bit (d:20mm, L:150mm) using the TTool? 1 (Very Negative) / 2 / 3 / 4 / 5 (Very Positive) Overall, how would you rate your experience with the Self Feeding Bit (d:50mm, L:90mm) using the TTool? 1 (Very Negative) / 2 / 3 / 4 / 5 (Very Positive) Overall, how would you rate your experience with the Twist Drill Bit (d:32mm, L:165mm) using the TTool? 1 (Very Negative) / 2 / 3 / 4 / 5 (Very Positive) Overall, how would you rate your experience with CircularSaw using the TTool? 1 (Very Negative) / 2 / 3 / 4 / 5 (Very Positive) Overall, how would you rate your experience with ChainSaw using the TTool? 1 (Very Negative) / 2 / 3 / 4 / 5 (Very Positive)

Fig.3 - UI Usability Questionnaire.

2.7 Matrix variations

By combining all the selected parameters for each objective of the evaluation we obtain a total of 18 cases to study per test cycle. Each test cycle should be repeated 2 times to test later the learning curve of the users and produce a wider dataset for this set. Hence, the *total operations to test per user will be 36*.

spéci men index	ML	toolhead itype	angle type
1	no	circular saw	90
2	no	circular saw	60
3	no	chain saw	90
4	no	chain saw	60
5	no	auger drill bit d:20mm, L:235mm	no

6	no	auger drill bit d:34mm, L:235mm	no
7	no	brad point drill bit d:20mm, L:150mm	no
8	no	self feeding bit d:50mm, L:90mm	no
9	no	twist drill bit d:32mm, L:165mm	no
10	yes	circular saw	90
11	yes	circular saw	60
12	yes	chain saw	90
13	yes	chain saw	65
14	yes	auger drill bit d:20mm, L:235mm	no
15	yes	auger drill bit d:34mm, L:235mm	no
16	yes	brad point drill bit d:20mm, L:150mm	no
17	yes	self feeding bit d:50mm, L:90mm	no
18	yes	twist drill bit d:32mm, L:165mm	no

Fig.4 - Matrix of all possible combinations to test for one testing cycle. This is repeated <u>twice</u> per user.

3 Evaluation DMP

In this section we go into details of the data collection, processing and representation phase.

3.1 Accuracy of the Estimated Pose

Using the AC (Augmented Carpentry), we can export the current validated pose of the tool and compare it with the digital twin of the mold in the same coordinate space (in AC).
This comparison is feasible because both the validated pose of the tool and the digital twin model possess reference points on the same coordinate system, such as axis start and end.

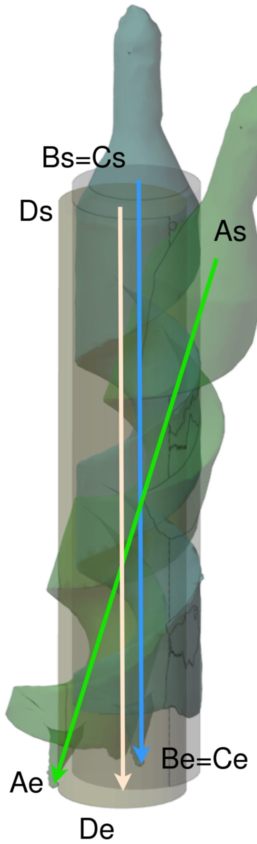

Fig.6 - Representation of the pose calculation error system: in grey, the axis of the physical hole of the plate with an exact corresponding; in blue, the axis of the physical tool head tightly inserted in the cavity; in yellow, the corresponding axis emplacement in the 3D space of the physical hole; in green, the tool head model aligned by TTool. Note the errors are exaggerated for representation purposes.

For chainsaw and circular saw we will use the same axis reference system with an analog comparison based on their control points.

Each vector consists of two points: a starting point 'A' and an ending point 'B'.
For evaluation, we are going to construct two such vectors—one representing the predicted pose (fig 6 vector A) and another for the ground truth based on the digital twin of the mold (fig 6 vector D). *The final error for the **position** will be calculated as the mean of the error for each axis between ground truth and estimated pose.*

For evaluating rotation disparity, we consider the vector as a directed line segment. Euler angles define the orientation of this vector in 3D space. These angles describe the orientation of the vector with respect to the fixed coordinate axes. We subsequently utilize these Euler angles (`$Predicted_Rotation, $GT_Rotation`) to determine the rotational disparity between the estimated pose and the ground truth pose. Both angles are measured in degrees and in the same rotation order. *The final error for the **rotation** will be calculated as the mean of the error for each axis between ground truth and estimated pose.*

The data for the pose calculation error will be graphically represented as follow:

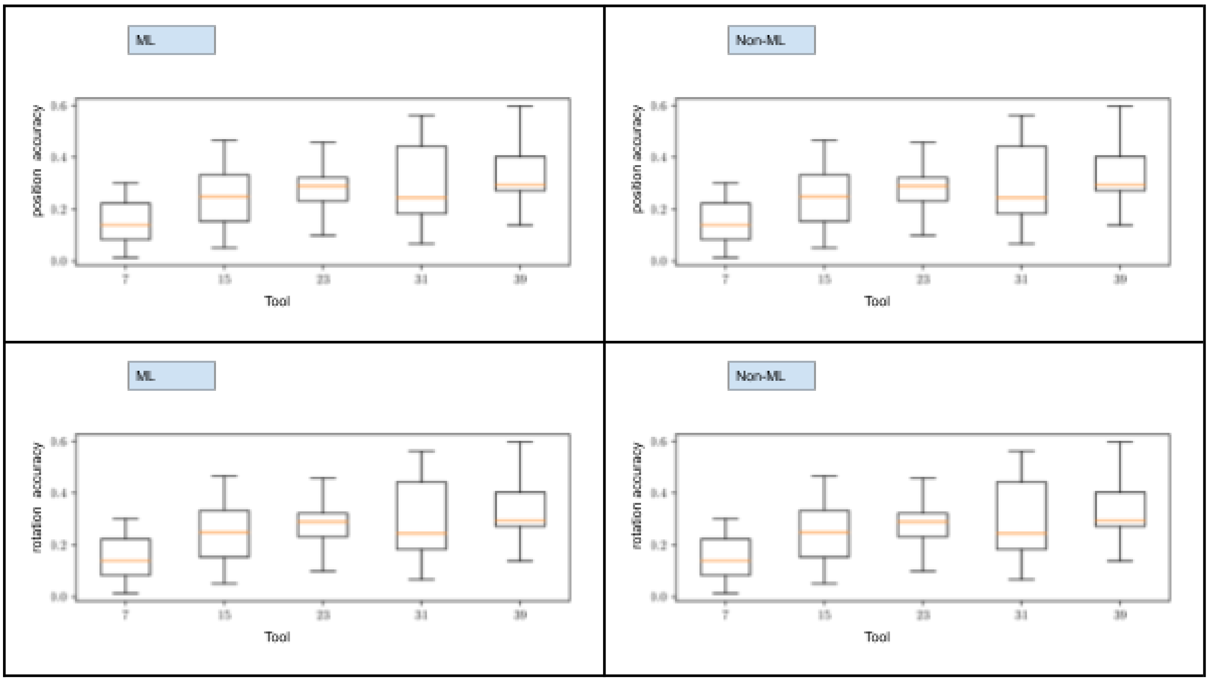

Fig.7 - Graphical representation for the pose error (translation + rotation) for both ML and non-ML version of TTool.

3.2 Time Efficiency

The output data for the time efficiency gauging:

`$startTrack`: This timestamp is recorded when the user starts interacting with each tool, thus initiating the time-tracking process.
`$endTrack`: This is captured when the user validates and saves the final pose of the tool, marking the conclusion of the time measurement for that specific tool and session.

The output of processed data is the following:

`$resultTrack` the time spent by user in that session calculated by subtracting the `$endTrack` from `$startTrack`.

The data will be graphically represented as follow:

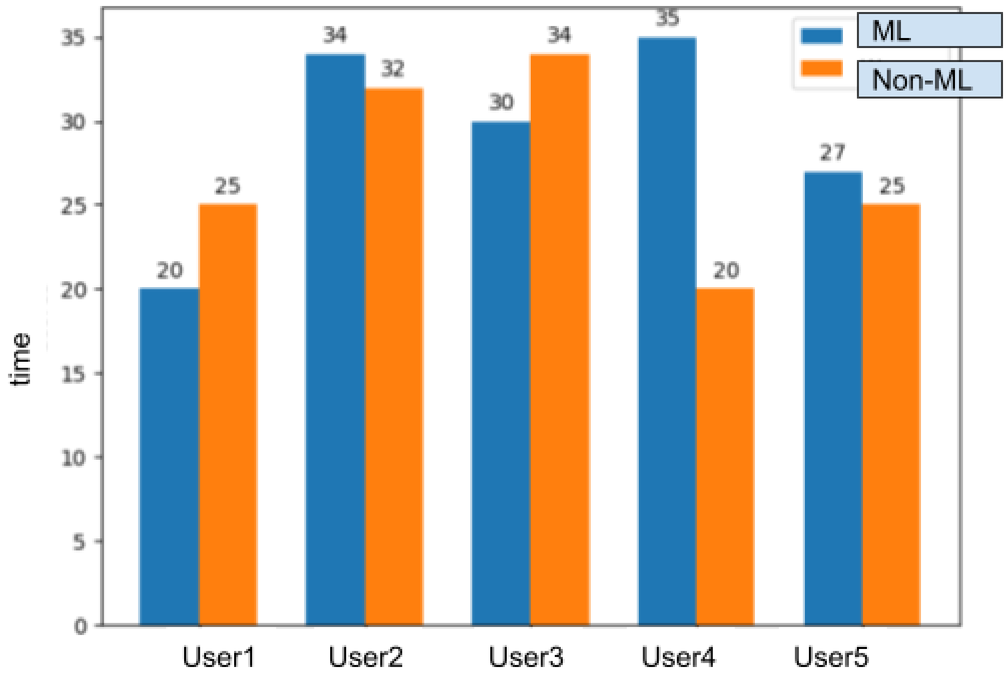

Fig.8 - Column bar graph for the representation of the time estimated to complete the task.

3.3 UI

In the evaluation of UI Usability, we use a Likert scale ranging from 'Strongly Disagree (1)' to 'Strongly Agree (5)'.
These qualitative responses are inherently mapped to numerical values, allowing for quantitative analysis.
For each question, we follow these steps:

- Calculate Mean Score: We assign numeric values to the Likert scale responses: Strongly Disagree (1), Disagree (2), Neutral (3), Agree (4), and Strongly Agree (5).
- Sum of Numeric Values: We add up the numeric values of all the responses for that question, denoted as `$Sum`.
- Total Number of Responses: We count the total number of responses for that question, represented as `$totalNum`.
- Mean Calculation for an individual question (`$MeanQuestion`): We calculate the mean score by dividing the sum of the numeric values (`$Sum`) by the total number of responses (`$totalNum`).
- Mean Calculation for each section (*ToolSelection, InputTool, ToolValidation*) in both ML and Non-ML: We calculate the mean for all questions within a section, we calculate the mean of these individual mean scores This section mean score represents the overall sentiment for that section.

The output of processed data is the following:
`$ToolSelectionMeanML` :mean sentiment score for the "Tool Selection" section within the Machine Learning (ML) approach.
`$InputToolMeanML` :mean sentiment score for the "Input Tool" section within the Machine Learning (ML) approach.

$ToolValidationMeanML :mean sentiment score for the "Tool Validation" section within the Machine Learning (ML) approach.
$ToolSelectionMeanNonML :mean sentiment score for the "Tool Selection" section within the Non-Machine Learning (Non-ML) approach.
$InputToolMeanNonML :mean sentiment score for the "Input Tool" section within the Non-Machine Learning (Non-ML) approach.
$ToolValidationMeanNonML :mean sentiment score for the "Tool Validation" section within the Non-Machine Learning (Non-ML) approach.

We have obtained mean scores that reflect the average opinions or perceptions of participants for each section in both the Machine Learning (ML) and Non-Machine Learning (Non-ML) approaches. These scores will serve as valuable data points for our analysis.

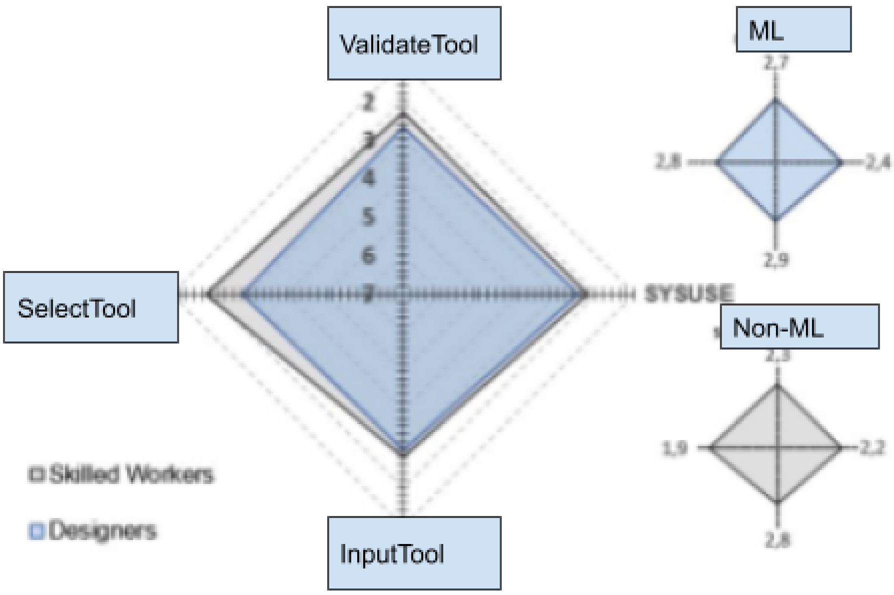

Fig.9 - Graphical representation for the UI usabilityresults.

---

4 Terminology

TTool = the 6DoF pose detector for tool heads
AC = the main C++ app for Augmented Carpentry
6DoF = the 6 degrees of freedom of the pose, in other words, the position and rotation

5 Annexes

Clarifications on the calculation of the error for the position and rotation-

Position:
After data collection, we have coordinates for points A and B. We are going to construct the vector.

Coordinates for A and B for **Predicted Pose** are as follows:

```
$Predicted_Point_A = ($Predicted_Point_AX, $Predicted_Point_AY,
$Predicted_Point_AZ)
$Predicted_Point_B = ($Predicted_Point_BX, $Predicted_Point_BY,
$Predicted_Point_BZ)
```

To create the (`$Predicted_Position`) vector, we subtract the coordinates of A from B:

```
$Predicted_Position = $Predicted_Point_B - $Predicted_Point_A
```

Or in component form:

```
$Predicted_Position_X=($Predicted_Point_BX - $Predicted_Point_AX)
$Predicted_Position_Y=($Predicted_Point_BY - $Predicted_Point_AY)
$Predicted_Position_Z=($Predicted_Point_BZ - $Predicted_Point_AZ)

$Predicted_Position =
($Predicted_Position_X,$Predicted_Position_Y,
$Predicted_Position_Z)
```

For the **Ground Truth**, we have::

```
$GT_Point_A = ($GT_Point_AX, $GT_Point_AY, $GT_Point_AZ)
$GT_Point_B = ($GT_Point_BX, $GT_Point_BY, $GT_Point_BZ)
```

To create the (`$GT_Position`) vector, we subtract the coordinates of A from B:

```
$GT_Position = $GT_Point_B - $GT_Point_A
```

Or in component form:

```
$GT_Position_X=($GT_Point_BX - $GT_Point_AX)
$GT_Position_Y=($GT_Point_BY - $GT_Point_AY)
$GT_Position_Z=($GT_Point_BZ - $GT_Point_AZ)
```

Thus,

```
$GT_Position = ($GT_Position_X,$GT_Position_Y, $GT_Position_Z)
```

Now, we compute the difference (`$Difference_Position`) between Predicted Pose vector and Ground Truth vector:

```
$Difference_Position = $Predicted_Position - $GT_Position
```

Or in component form:

```
$Difference_Position_X=($Predicted_Position_X - $GT_Position_X)
$Difference_Position_Y=($Predicted_Position_Y - $GT_Position_Y)
$Difference_Position_Z=($Predicted_Position_Z - $GT_Position_Z)
```

Thus,
`$Difference_Position = ($Difference_Position_X,$Difference_Position_Y,$Difference_Position_Z)`

The output of processed data is the following:
`$Difference_Position_X`: the difference for the translation X component
`$Difference_Position_Y`: the difference for the translation Y component
`$Difference_Position_Z`: the difference for the translation Z component

These outputs indicate how much the vectors deviate from each other in terms of position.

Position Accuracy Metric
The evaluation of the position accuracy of the estimated pose is expressed by the Euclidean distance.

Using `$Difference_Position_X, $Difference_Position_Y, $Difference_Position_Z` the differences in X, Y, and Z coordinates of ground truth and TTool pose estimation vectors , we compute the Euclidean distance.
We compute the position accuracy values, represented as `$AccuracyPosition`

`$AccuracyPosition =`
$$\sqrt{(\$Difference\_Position\_X)^2 + (\$Difference\_Position\_Y)^2 + (\$Difference\_Position\_Z)^2}$$

By aggregating the accuracy value, we compute the several values:
- mean accuracy
- minimum accuracy
- maximum accuracy

To compute the mean accuracy value, denoted as `$MeanAccuracyPosition`, respectively, across all tool and joinery positions for both the ML and Non-ML approaches.
$$\$MeanAccuracyPosition = \frac{(\Sigma(\$AccuracyPosition))}{N}$$

where, `$AccuracyPosition` is each individual AccuracyPosition, `N` is the total number of values.

To compute minimum accuracy:
`$MinAccuracyPosition = min(`
$\$AccuracyPosition_1$, $\$AccuracyPosition_2$,..., $\$AccuracyPosition_N$)

To compute maximum accuracy:
`$MaxAccuracyPosition = max(`
$\$AccuracyPosition_1$, $\$AccuracyPosition_2$,..., $\$AccuracyPosition_N$)

Rotation:
So we have
`$Predicted_Rotation = ($Predicted_Alpha, $Predicted_Beta, $Predicted_Gamma)`

and

`$GT_Rotation = ($GT_Alpha, $GT_Beta, $GT_Gamma)`

Subsequently, we calculate the absolute difference in Euler angles between the predicted pose and the ground truth as a measure of rotation disparity. The differences are computed as:

`$Difference_Rotation = | $Predicted_Rotation - $GT_Rotation |`

Or in component form:
```
$Difference_Rotation_Alpha= | $Predicted_Alpha - $GT_Alpha |
$Difference_Rotation_Beta = | $Predicted_Beta - $GT_Beta |
$Difference_Rotation_Gamma= | $Predicted_Gamma - $GT_Gamma |
```

Thus,
```
$Difference_Rotation =
($Difference_Rotation_Alpha,$Difference_Rotation_Beta,$Difference_
Rotation_Gamma)
```

Rotation Metric
The evaluation of the rotation accuracy of the estimated pose is expressed by the Euclidean distance.

Using `$Difference_Rotation_Alpha, $Difference_Rotation_Beta, $Difference_Rotation_Gamma` the differences in α, β and γ coordinates of ground truth and TTool pose estimation Euler angles , we compute the Euclidean distance.
We compute the rotation accuracy values, represented as `$AccuracyRotation`

`$AccuracyRotation =`

$$\sqrt{(\$Difference\_Rotation\_Alpha)^2 + (\$Difference\_Rotation\_Beta)^2 + (\$Difference\_Rotation\_Gamma)^2}$$

By aggregating the rotation accuracy value, we compute the several values:

- mean accuracy
- minimum accuracy
- maximum accuracy

To compute the mean rotation accuracy value, denoted as `$MeanAccuracyRotation`, respectively, across all tool and joinery positions for both the ML and Non-ML approaches.

$$\$MeanAccuracyRotation = \frac{(\Sigma(\$AccuracyRotation))}{N}$$

where, `$AccuracyRotation` is each individual rotation, `N` is the total number of values.

To compute minimum accuracy:
`$MinAccuracyRotation = min(`
$\$AccuracyRotation_1$, $\$AccuracyRotation_2$,..., $\$AccuracyRotation_N$)

To compute maximum accuracy:
`$MaxAccuracyRotation = max(`
$\$AccuracyRotation_1$, $\$AccuracyRotation_2$,..., $\$AccuracyRotation_N$)