Human Interaction and Emerging Technologies (IHIET-AI 2024), Vol. 120, 2024, 250–259
https://doi.org/10.54941/ahfe1004575

Automatic Transcription of the Methods-Time Measurement MTM-1 Motions in VR

Valentina Gorobets[1], Roman Billeter[1], Rolf Adelsberger[2], and Andreas Kunz[1]

[1]Swiss Federal Institute of Technology, Zurich, 8092, Switzerland
[2]Sensoryx, Zurich, 8005, Switzerland

ABSTRACT

Our paper presents an approach to automatically detect and transcribe basic human motions in VR by means of the Methods-Time Measurement (MTM) system. MTM is a predetermined motion time system that consists of a list of predefined basic motions and the mean time values corresponding to those motions. This system is used to analyze manual workplaces. Currently, the MTM analysis is conducted manually. The working process that needs to be analyzed is video captured and further analyzed by dividing it into a sequence of basic MTM motions. There are various MTM systems that differ by their granularity level, such as MTM-1, MTM-2, MTM-UAS, etc. We propose and evaluate an approach of the automatic transcription of the MTM-1 basic motions. For our research, we use Unity software to create the virtual environment (VE) and interactions within it. Additionally, we use the HTC Vive tracking system and Sensoryx VRfree data glove that enable body- and hand-tracking. Our automatic transcription algorithm employs four decision trees that run simultaneously, each dedicated to transcribing hand, arm, body, and leg motions in real time. To assess our algorithm, we conducted a user study with 33 participants.

Keywords: Virtual reality, Methods-time measurement, Motion recognition, Process optimization

INTRODUCTION

Due to the increased availability of Virtual Reality (VR) technologies, they become more popular in industrial settings. This is driven by the increasing digitalization in industry, recently boosted by the “Industry 4.0” initiatives. With this, new approaches such as using different variations of digital twins (Jones, 2020) appear, that were not used previously in the industry.

For these digital twins, VR serves as a great visualization tool, since the spatial representation of objects significantly increases the sense of presence and immersion in comparison to the normal screen (Shu, 2019). Additionally, it enables intuitive interactions with the virtual objects, as well as easy tracking data acquisition. Therefore, VR is predestined for assessment purposes, such as the evaluation of workplaces. Measuring the movements of a person at a virtual workplace could help to optimize its layout prior to the physical implementation.

Methods-Time Measurement

Methods-Time Measurement (MTM) is a predetermined motion time system used to analyze manual processes. It was first introduced by L.H. Maynard & L.B. Stegemerten (Maynard, 1948). The MTM system analyses human motions by dividing them into a sequence of so-called basic motions, that are defined in this system. Every basic motion had a predetermined time value. Those time values are measured in so-called Time Measurement Units (TMUs).

There are various MTM systems, which differ by their granularity level. MTM-1 is considered to be the most detailed system, followed by the MTM-2, MTU-UAS, and MTM-MEK systems. Table 1 presents the list of categorized basic MTM-1 motions. The disadvantage of more detailed systems is the time and effort to perform MTM analysis.

Table 1. List of MTM-1 basic motions.

Upper Body Motions	**Arm Motions**	Reach	Move	Crank	Turn	
	Hand Motions	Grasp	Position	Disengage	Release	Apply pressure
Lower Body Motions	**Body Motions**	Sit	Bend	Kneel one knee	Kneel both knees	
	Leg gestures	Step	Side step	Turn step	Leg motion	Foot motion
	Eye Motions	Eye focus	Eye travel			

All MTM systems suffer from either requiring an already existing workplace, or a mock-up of it resembled by cardboard. A worker's movements are then video recorded and manually analyzed by multiple MTM experts, which is a time-consuming and error-prone process. Consequentially, automating this analysis will reduce effort and costs.

The most common use case for MTM systems is the ergonomics improvement of the workplace (Laring, 2002). Additionally, MTM can be used for the process planning (Morlock, 2017), as well as training progress tracking (Muller, 2016).

RELATED WORK

One of the possible approaches suggests using additional sensors and RFID tags (Fantoni, 2020). However, their method measures times at certain positions instead of detecting basic motions. Another research proposes an approach utilizing a single camera with the convolutional neural network to detect MTM-1 hand motions (Riedel, 2021). Due to the limited field of view of the employed camera, their work detects hand motions only, while all other body movements are not evaluated. Both works have in common that the users' actions were only performed in real environments.

Previous research (Gorobets, 2021) confirms the feasibility of conducting an MTM analysis in VR. It compares results obtained by a direct motion observation and MTM-2 analysis in identical setups in VR and reality. It concludes that TMU times obtained from MTM-2 analysis in VR correspond to the values obtained by conducting MTM-2 analysis in reality.

Some researchers (Bellarbi, 2019; Andreopoulos, 2022) proposed their approaches to detect MTM-2 basic motions in VR. However, automatic transcription of the MTM-1 basic motions in VR has not yet been investigated.

Research Gap

In our research, we concentrate on enhancing the virtual MTM-1 by introducing an automatic transcription approach. This approach is beneficial for non-existing workplaces, as it allows to avoid building a physical workplace or mock-up of it, but utilizes its virtual model instead.

METHODOLOGY

Setup

Hardware: To visualize an interactable virtual environment (VE), we used the HTC Vive Pro system. It consists of one head-mounted display (HMD), three trackers, and two base stations. To track hand gestures, we used the VRfree data gloves provided by Sensoryx. This system consists of a head module that is attached to the HMD and the data gloves. A head module tracks the wrist position. To track finger positions, inertial measurement units (IMUs) in the gloves are used. For tracking participants' motions we used the setup described as follows. Head motions are tracked using the HTC Vive Pro HMD. Hand motions are tracked with the VRfree glove. Lower body motions are tracked with three HTC Vive Trackers: one placed on the hip of the user and two placed on the feet.

Software: For creating the VE and interactions within, we use Unity 3D version 2021.3.13f1. Additionally, we use the Sensoryx SDK to enable integration of the VRfree gloves in Unity 3D. To create a realistic human avatar, we use the Make Human modelling software. Additionally, we use the VRIK Unity asset that animates the avatar based on the inversed kinematics principle.

VE design: The virtual workplace consists of several objects on a table that need to be grasped, assembled, and placed again back on the table, while the user is seated. The workplace integrated also other motions like cranking and foot motion. Other areas of the workplace also required step motions and kneeling. Due to the lack of an eye tracker and the low TMU contribution of eye movements, we did not integrate this MTM-1 class into our evaluation procedure. The overall layout for the seated operations is shown in Fig. 1.

Experiment design: To test that our algorithm transcribes possible MTM-1 motions, we developed a VE that has various tasks to perform. For this, we split up the study tasks into two groups: actions preformed while the users were seated, and actions preformed while the users in a standing position. The seated area aims to concentrate on the upper body motions and foot motions, while the standing position aims to automatically recognize other actions like step and kneel.

User study procedure: The user study starts with a brief introduction of the experiment goals. It is followed by the task explanation as well as familiarization with the VR equipment. Lastly, a participant completes the task in VR.

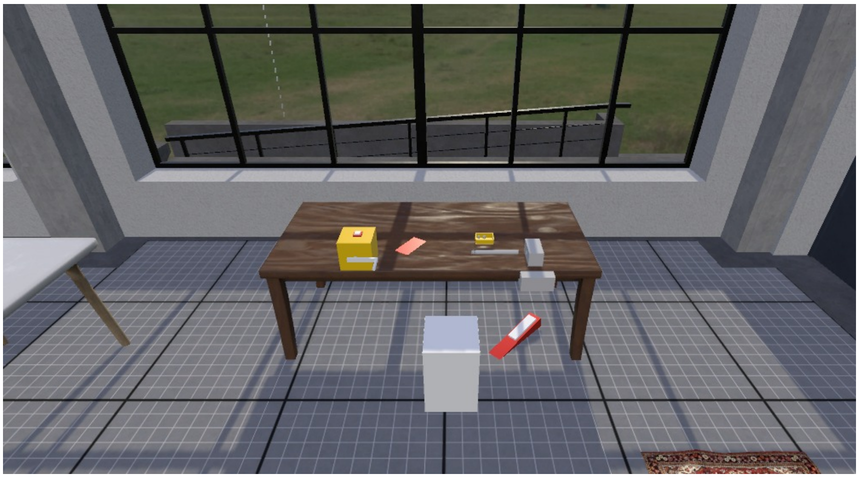

Figure 1: VE design, seated area. The cube represents a chair that participants are sitting on.

Automatic Transcription

To deliver the MTM-1 basic motions from the motions sequence performed by a user, we decided to use a decision-tree approach. The benefit in comparison to machine learning is that it does not require preliminary training of the network. Moreover, the decision tree approach is highly advantageous in the context of MTM-1 motions, considering to the pre-defined nature of these movements. Additionally, in VR initial information about the environment and virtual objects is available, and it reduces the complexity associated with detecting motions.

Our decision-tree approach consists of *four* decision trees that run simultaneously for (i) arm motions, (ii) hand motions, (iii) body motions, and (iv) leg gestures. Each of them transcribes one subgroup of the MTM-1 basic motions (see Table 1). Below, we will describe the working principle of each of them.

Upper Body Motions Recognition

This part consists of two decision trees: one for hand motions and one for arm motions.

The **hand motions** decision tree is triggered when an object has been grasped or released. Therefore, the trigger for the start of the hand motion decision tree is the beginning or end of touching a virtual object. Every moment of time, our algorithm checks whether a virtual object was touched or not. We observed that hand motions appear in the following sequence (see Fig. 2).

The **arm motions** decision tree is triggered by the end of the *Grasp* or *Release* hand motion. As arm motions precede hand motions, we use the pre-recorded tracking information in order to deliver correct transcription. Firstly, we check whether the hand action that triggered this decision tree is *Release*. If yes, then a preceding arm motion is either *Crank,* or *Move*. If the released virtual object had a label *crank*, we transcribe performed arm motion as *Crank*. If it was a different object, we transcribe *Move* motion. If the hand action that triggered the arm decision tree was not *Release*, it means that it was triggered by the transcribed hand motion *Grasp*. Therefore, the preceding arm motion was *Reach*.

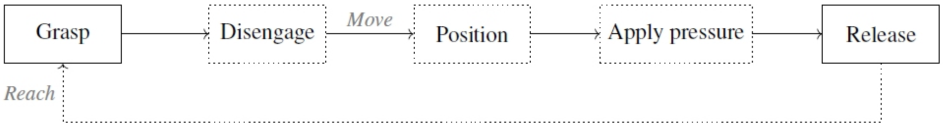

Figure 2: Hand motions cycle in MTM-1. The dashed motions are optional in a cycle. The names on the arrows represent the arm motions that make a transition between corresponding hand motions.

The *Turn* motion does not intuitively fit into the arm motions category. Its definition is the turning of the wrist during a reach or move motion. As *Turn* motion is always performed during *Reach* or *Move* arm motions, we integrated it into the arm motion decision tree, as it cannot be considered as an independent hand motion.

Lower Body Motions Recognition

This part consists of two decision trees: one for general body motions and one for leg gestures.

The **body motions** decision tree is triggered when a head position (h_{head}) is lower than the predefined threshold T_{Bend}. Similarly to the hand motions, we observed the sequential nature of the body motions (see Figure 3). Starting from the neutral standing position, lowering the head identifies the beginning of the *Bending* motion. Additionally, if the hip position is also getting below a threshold value T_{Sit}, we are transcribing the performed action as *Sit*. To transcribe *Kneeling* motions, we are considering the relative angle (α_{foot}) between the trackers that are located on the feet and the floor. After detection of one of the above-mentioned motions, participants return to their neutral standing position by performing an *Arise* motion. *Arise* motion is detected using the h_{head} and T_{Arise} threshold: $T_{Arise} < h_{head}$, $T_{Bend} < T_{Arise}$.

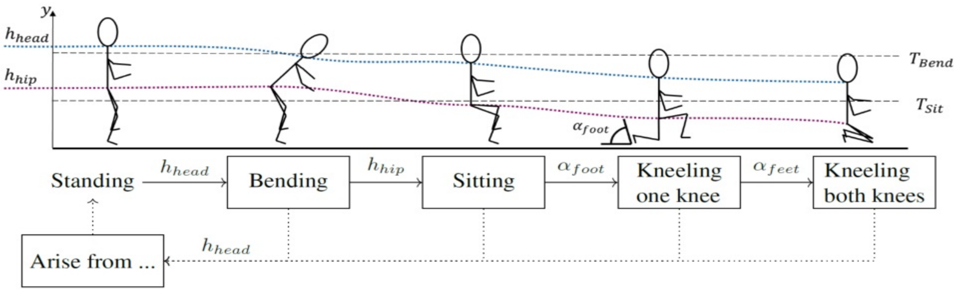

Figure 3: Body motions. The body motions are sequential and tracker metrics are used to detect the motions.

Threshold definition:

The thresholds T_{Bend} and T_{Sit} were defined empirically and are both 20 cm lower than the height of the head and hips in the neutral *Standing* position:

$$T_{Bend} = h_{head,standing} - 20 \text{ cm}, T_{Sit} = h_{hip,standing} - 20 \text{ cm}.$$

The threshold for kneeling angle T_{Kneel} is also obtained empirically and is 50°.

The T_{Arise} threshold should be set below $h_{head,standing}$ to ensure it is achieved, and above T_{Bend} to avoid multiple detection of a *Bending* motion. Empirically we set T_{Arise} threshold at the midpoint between these two values:

$$T_{Arise} = h_{head,standing} - 10 \text{ cm}.$$

Leg gestures consist of two subgroups: different *Step* variations and *Foot* and *Leg motions*. An example of the second subgroup is pressing a pedal. These two subgroups can be very similar and traditionally are differentiated by the estimate of the MTM expert. Due to this similarity, we decided to distinguish between those two subgroups by making an assumption that *Foot* and *Leg motions* happen only while seated.

The *Leg gesture* decision tree operates continuously throughout the entire algorithm's runtime, without being reliant on specific triggering events. It uses velocity of the foot (v_{foot}) to detect leg gestures.

Figure 4 depicts the logic of the hysteresis loop for the detection of the foot moving to avoid false *Leg gesture* detection. We empirically define two thresholds for v_{foot}:

- T_{in} defines the start of the leg gesture;
- T_{out} defines the end of the leg gesture.

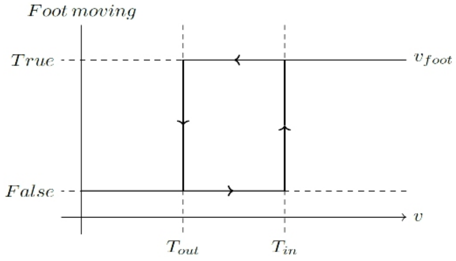

Figure 4: Hysteresis for detecting leg gestures.

These thresholds are used to filter out the noise caused by the tracking system. As it is common for hysteresis loops, these thresholds should be clearly separated ($T_{out} < T_{in}$) to avoid repeated detection of the same leg gesture. When the v_{foot} rises, it follows the lower path. If it exceeds $T_{in,}$ the foot is considered to be moving. If, after that, v_{foot} lowers again, it follows the upper path, and if it drops below T_{out}, the foot stopped moving, and we consider it as the end of the leg gesture.

Threshold definition: We define T_{in} and T_{out} separately for detecting *Step* ($T_{in,step}$, $T_{out,step}$) and *Foot* and *Leg motions* ($T_{in,foot/leg}$, $T_{out,foot/leg}$). However, we use the same procedure to define the thresholds for both subgroups. We record and analyse the velocity of the foot of a test user performing either *Step* or *Foot* or *Leg motions*.

Figure 5 illustrates the test user's foot velocity performing three steps over time. Based on our empirical analysis, we define two thresholds for detecting *Step* motions: $T_{in,step} = 0.3$ m/s, $T_{out,step} = 0.05$ m/s. Similarly, we define the *Foot* and *Leg motion* thresholds: $T_{in,foot/leg} = 0.1$ m/s, $T_{out,foot/leg} = 0.05$ m/s.

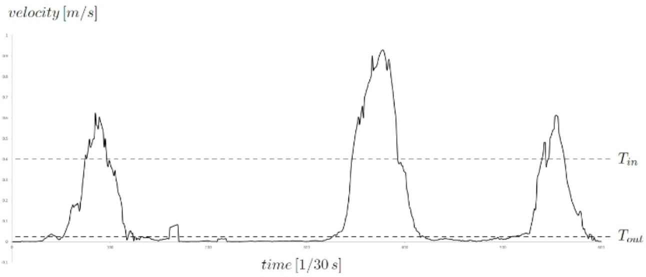

Figure 5: Velocity of the foot of a test user performing 3 steps over time.

Therefore, the leg gestures decision tree has the following logic. Firstly, we check whether a motion is performed seated or upright. If the motion was performed upright and we detect a foot moving, we transcribe one of the *Step* motions. If the foot's turning angle along the floor is between 45° and 90°, *Turn Step* is transcribed. Otherwise, if the foot travels less than 60% in the forward direction of the foot, it is considered a *Side Step*, and if not, normal *Step* is transcribed.

If the motion was performed seated and we detect a foot moving, we transcribe either *Foot* or *Leg Motion*. A *Leg Motion* is transcribed if the disposition of the feet is above 10cm, and a *Foot Motion* if it is below.

RESULTS AND DISCUSSION

To assess the results received by the proposed algorithm, we are using the following metrics. Firstly, every automatically delivered basic motion was labeled as True Positive (TP), False Positive (FP), or False Negative (FN). Positive stands for a transcribed motion and negative for the absence of a transcription. True stands for the correctly recognized motion by the algorithm and false for the incorrect recognition. Therefore:

- TP - The algorithm correctly transcribed a motion that was performed by the user.
- FP - The algorithm transcribed a motion that was not performed by the user.
- FN - The algorithm did not transcribe a motion even though the user performed one.

Based on those metrics, we additionally introduce *Precision* and *Recall*:

$$\text{Precision} = \frac{\Sigma \text{TP}}{\Sigma \text{FP} + \Sigma \text{TP}}; \ \text{Recall} = \frac{\Sigma \text{TP}}{\Sigma \text{FN} + \Sigma \text{TP}} \tag{1}$$

Precision is the ratio of true positive results to the total number of positive results. It measures the accuracy of the algorithm in identifying true positives. *Recall* is the ratio of true positive results to the total number of relevant results. It measures the completeness of the model in identifying all relevant results.

General Results

In the user study, 2846 motions were transcribed by the algorithm. 2670 of them are TPs, 176 are FPs, and 68 are FNs. The overall *Precision* and *Recall* of the transcription algorithm are: *Precision* = 0.938, *Recall* = 0.975.

Below, we give a summary of the results for all transcribed basic motions.

Table 2. Summary of the results for transcribed hand motions.

	Hand Motions				
	Grasp	Release	Position	Apply Pressure	Disengage
TP	384	380	60	31	29
FP	21	35	0	0	0
FN	3	5	2	0	2
Precision	0.948	0.916	1	1	1
Recall	0.992	0.987	0.968	1	0.935

Table 3. Summary of the results for transcribed arm motions.

	Arm Motions		
	Reach	Move	Crank
TP	314	342	34
FP	1	26	4
FN	3	4	0
Precision	0.997	0.929	0.895
Recall	0.991	0.988	1

Discussion

As it is seen from the general results, our algorithm has a good overall performance. The majority of TPs in *Grasp* (see Table 2) are caused by tracking problems. Additionally, they lead to TPs of *Release* motion, as *Grasp* and *Release* are interdependent motions.

Leg Motion and *Foot Motion* transcription (see Table 5) can be improved by additional trackers attached to each leg. As we only used two trackers on the feet, it is difficult to distinguish between those two motions. Additionally, redefining empirically found parameters for different step types can also improve precision and recall for different *Step* variations.

Table 4. Summary of the results for transcribed body motions.

Body Motions				
	Sit	Arise from Sit	Bend	Arise from Bend
TP	64	62	33	33
FP	0	0	0	0
FN	1	2	0	0
Precision	1	1	1	1
Recall	0.985	0.969	1	1
	Kneel one Knee	Arise from one Knee	Kneel both Knees	Arise from both Knees
TP	66	66	33	33
FP	0	0	0	0
FN	0	0	0	0
Precision	1	1	1	1
Recall	1	1	1	1

Table 5. Summary of the results for transcribed leg gestures.

Leg Gestures					
	Step	Side Step	Turn Step	Leg Motion	Foot Motion
TP	151	131	98	58	38
FP	16	20	0	15	7
FN	3	4	32	10	26
Precision	0.904	0.868	1	0.795	0.844
Recall	0.981	0.97	0.754	0.853	0.594

CONCLUSION

In this paper, we presented a decision tree approach to automatically transcribe MTM-1 basic motions in VR. We discussed all the potential benefits of using VR for the MTM analysis. We presented our user study procedure as well as a thorough explanation of the decision tree approach we proposed. We finalized our paper by presenting and discussing the results of our study.

For future work, the setup could be enhanced by additional trackers on the legs to improve *Foot* and *Leg motion* recognition, as well as an eye tracker to detect *Eye motions*. The algorithms could be improved by redefining thresholds for *Side step* and *Turn step* motion. Finally, the study procedure in terms of realistic tasks could be improved by using the algorithm in an already existing industrial workplace.

ACKNOWLEDGMENT

We would like to express our gratitude to Sensoryx for providing the tracking space and continuous support with the glove. Additionally, special thanks to Thomas Nescher for providing the virtual model of the industrial building.

REFERENCES

Andreopoulos, E., Gorobets, V. and Kunz, A. (2022), 'Automatic MTM-Transcription in Virtual Reality Using the Digital Twin of a Workplace', (preprint).

Bellarbi, A., Jessel, J.-P. and Dalto, L. D. (2019), 'Towards Method Time Measurement Identification Using Virtual Reality and Gesture Recognition', 2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), IEEE, 2019.

Fantoni, G., Al-Zubaidi, S. Q., Coli, E. and Mazzei, D. (2020), 'Automating the process of method-time-measurement', International Journal of Productivity and Performance Management 70(4), 958–982.

Gorobets, V., Holzwarth, V., Hirt, C., Jufer, N. and Kunz, A. (2021), 'A VR-based approach in conducting MTM for manual workplaces', The International Journal of Advanced Manufacturing Technology 117(7-8), 2501–2510.

Jones, D., Snider, C., Nassehi, A., Yon, J. and Hicks, B. (2020), 'Characterising the Digital Twin: A systematic literature review', CIRP Journal of Manufacturing Science and Technology 29, 36–52.

Laring, J., Forsman, M., Kadefors, R. and Örtengren, R. (2002), 'MTM-based ergonomic workload analysis', International Journal of Industrial Ergonomics 30(3), 135–148.

Maynard, H. B., Stegemerten, G. J., & Schwab, J. L. (1948). Methods-time measurement.

Morlock, F., Kreggenfeld, N., Louw, L., Kreimeier, D. and Kuhlenkötter, B. (2017), 'Teaching Methods-Time Measurement (MTM) for Workplace Design in Learning Factories', Procedia Manufacturing 9, 369–375.

Müller, B. C., Nguyen, T. D., Dang, Q.-V., Duc, B. M., Seliger, G., Krüger, J. and Kohl, H. (2016), 'Motion Tracking Applied in Assembly for Worker Training in different Locations', Procedia CIRP 48, 460–465.

Riedel, A., Brehm, N. and Pfeifroth, T. (2021), 'Hand Gesture Recognition of Methods-Time Measurement-1 Motions in Manual Assembly Tasks Using Graph Convolutional Networks', Applied Artificial Intelligence 36(1).

Shu, Y., Huang, Y.-Z., Chang, S.-H. and Chen, M.-Y. (2018), 'Do virtual reality head-mounted displays make a difference? A comparison of presence and self-efficacy between head-mounted displays and desktop computer-facilitated virtual environments', Virtual Reality 23(4), 437–446.