Web of Tactile Things: Towards an Open and Standardized Platform for Tactile Things via the W3C Web of Things

Van Cu PHAM[1)⋆[0000−0001−8320−6001]], Quan Khanh LUU[1)⋆[0000−0002−8544−7260]], Tuan Tai NGUYEN[1)⋆[0000−0002−2417−4756]], Nhan Huu NGUYEN[1)⋆[0000−0002−9309−7336]], Yasuo TAN[1)[]], and Van Anh HO[1),2)⋆⋆[0000−0002−4519−0448]]

1) Japan Advanced Institute of Science and Technology, Ishikawa 923-1211, Japan
2) Japan Science and Technology Agency, PRESTO, Kawaguchi, Saitama, Japan
{cupham, quan-luu, tuan-nguyen, nnnhan, ytan, van-ho}@jaist.ac.jp

Abstract. Toward the development of the Tactile Internet beyond the 5G era and the recently-introduced *Metaverse*, a need for standardized platforms to exchange haptics information for human and cyber-physical systems is emerging. This paper introduces our attempt for an open and standardized platform for tactile things, namely Web of Tactile Thing (WoTT). The WoTT extends the W3C Web of Things (WoT) to exchange haptic information from tactile sensing devices to cross-domain services via the W3C WoT abstractions. This paper proposes (i) haptic vocabularies to generate the WoT Thing Description for vision-based tactile sensing devices, as well as (ii) mechanisms to connect, update, and exchange tactile information efficiently. To prove the feasibility of the proposed solution, a proof of concept includes (i) a tactile device to produce tactile information and (ii) a WoT client that consumes proposed vocabularies to create a digital twin of the physical device, has been implemented. The validity and interoperability of our proposed solution have been verified by abilities to reproduce the digital twin and reflect touch events timely and correctly via our extension.

Keywords: Web of Tactile · Tactile Internet · Tactile Interoperability.

1 Introduction

In the new normal of tele-working, online class, users are separated by distance and only can interact through computers' screen or virtual space. Current technology allows transmission of audio-visual information through the internet with reasonable latency. However, for the ultimate experience in cyber-physical system, another perceptions' immersion are considered crucial. Of human's five senses, touch can bring in detailed feedback on the mechanical properties of

⋆ Authors are equally contributed
⋆⋆ Corresponding author

the surrounding thanks to dense distribution of mechanoreceptors underneath the skin, benefiting a wide range of human's functions, such as object grasping/manipulation or locomotion [7]. Touch also helps humans perceive the roughness of a surface, the sharpness of an edge, temperature of an object without even looking at them. Especially, human skin is considered as a social mean, since emotion/feeling can be conveyed through touch primitives over small or large scale of skin, benefiting non-verbal communication among humans [15]. Nonetheless, the immersion of touch in the cyber-physical system is still limited, due to the lacking of universal tangible devices that has high spatio-temporal specs. Current tactile (or touch) sensing devices employed a variety of principles for transducing mechanical stimuli into measurable electrical parameters, such as resistance, capacitance, optical one [19]. Most of the devices are designed for small scales with flat, rigid surfaces, such as robotic fingers' pad, smart phone's touch screen, and so on. In fact, humans have skin covering the whole body, acting as largest organ with tactile sensation and softness. Therefore, a tactile sensing device, which can act as an input interface, should be scalable with continuous soft skin for enhancing reception of physical contact stimuli [16]. Also, in this era, such devices are expected to be connected through the internet, allowing transmission of tactile data with low latency, and exploited/accessed easily by applications, just like how audio-visual perceptions have played its role up to now.

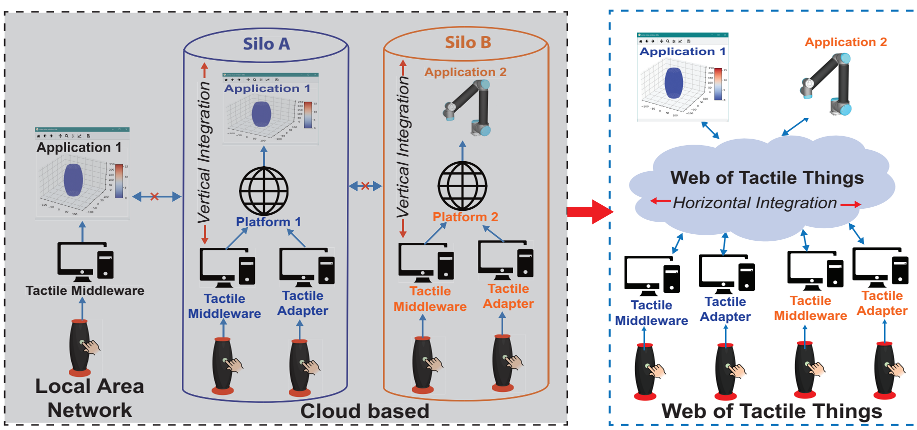

Fig. 1. Web of Tactile Things (WoTT): Breaking incompatible silos for Tactile Things towards the implementation of the Tactile Internet. Our goal is to utilize the horizontal integration at the platform level to counter the fragmentation for the Tactile Internet

In this paper, we propose an extension to the W3C Web of Things (WoT) that supports tactile sensing devices to pave the way for an open and standardized platform, named *Web of Tactile Things (WoTT)*, for interconnecting tactile devices, exposing tactile data to applications or services via the Web paradigm.

This WoTT is supposed to bring the simplicity, openness, and success of today Web towards the Tactile Internet. As the very first step, we particularly focused on supporting vision-based tactile sensing devices previously reported in [5],[18], that have high spatial resolution with low sampling rate. Such devices are highly scalable, and pleasure to physical contact, thanks to its inherent soft, continuous skin.

Main contributions of this work include:

- Proposing an **information model** to support semantic interoperability for tactile sensing devices.
- Proposing a **platform** for tactile devices and services via the semantic abstraction of the W3C WoT.
- Implementing a **use-case** for the W3C WoT towards the development of the Tactile Internet.

The rest of this paper is organized as follows:

- In **Section** 2, preliminary concepts of the W3C WoT, tactile internet, and our vision-based tactile sensing devices will be introduced.
- In **Section** 3, design concepts, physical touch processing, and the mapping from physical elements into the information model are explained in detail.
- In **Section** 4, the experiment includes (i) the tactile sensing device (hardware), (ii) the platform, and (iii) an application to build the digital twin of the hardware, will be described.
- In **Section** 5, evaluations and results will be discussed.
- Finally, this work is summarized in **Section** 6.

2 Basic Concepts

2.1 Tactile Internet: How to avoid the current IoT fragmentations?

Tactile Internet (TI) is believed to be a *revolutionary level of development for society, economics and culture* [14]. TI intents further to improve the Human-Cyberspace interactions and will be a important factor towards the "Metaverse". Nevertheless, the transmission of haptic information is challenging as stated in [4]. Towards the TI, we can learn the lesson related to interoperability issues of the current IoT to *break silos* caused by the vertical integration (See **Figure** 1). In the early state of the TI, an open and standardized platform that interconnect devices, services, vendors, and, developers,etc., should be considered.

2.2 W3C Web of Thing

The WoT is backed by the W3C consortium and is currently applied to multiple IoT areas such as industrial [12], smart buildings [6], smart cities, retails, healthcare, etc., so that multiple vendors and ecosystems can be interoperable via its semantic abstractions.

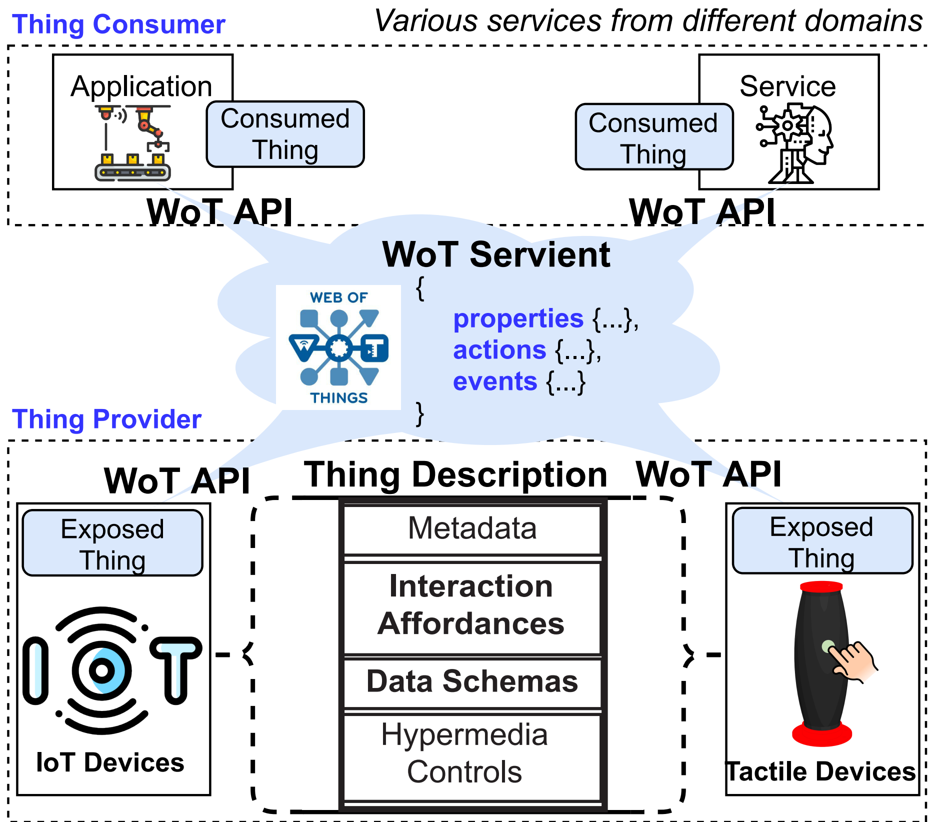

Fig. 2. Overview of the W3C Web of Things. Thing Description, which supports semantic abstractions for interactions between things(devices, services) within WoT ecosystem utilizing WoT APIs via the WoT Servient, is the heart of the WoT.

.

As illustrated in **Figure** 2, *things* in the WoT are abstractions of physical or virtual entities and they are accessible via a software stack that implements WoT building blocks, named *WoT Servient*. The most important part, which is the semantic enabler of the WoT, is the *Thing Description* (TD). In general, a TD instance defines following components:

- *Metadata*: A human-readable text to describe about the *thing* such as *@context, title, description, etc.*
- *Interaction Affordances*: to describe **what** kind of interactions a *thing* provides, and **how** to utilize those interactions. There are three following types of *Interaction Affordances*:
  - *PropertyAffordance*: could be utilized for getting or setting parameters of *things*. For example, ***operationStatus*** is a property that indicates current status of a thing, or switches current status to a desired value.

  - *ActionAffordance*: to describe actions (physical, time-consuming processes) that a thing provides. For example, ***toggle*** is an action that requires a process to achieve it(check the current status, then toggle it).
  - *EventAffordance*: to describe a push notification mechanism that sends notifications, events, value streams to subscribers asynchronously. For example, ***touchDetected*** is an event that notifies subscribers whenever a human is in the sensor range.
- *Data Schemas*: to define data formats of exchanged data. Data schemas of the TD are a subset of the JSON Schema [3]. The currently supported data schemas include: **array**, **boolean**, **number**, **integer**, **string**, **object**, and **null** data types.
- *Hypermedia Controls*: to define mapping between supported affordances into target protocol forms and data payload so that **Thing Consumers** and **Thing Providers** are able to interact using a consistent interaction model.

The W3C WoT also defines the *Scripting API* [1] to ease the development of WoT applications. Main tasks when defining TD(s) include (i) defining affordances (properties, actions, events), (ii) defining data types for corresponding affodances, and (iii) configuring security schemes as well as specifying target binding protocols.

2.3 Tactile Sensing Devices in the Web of Tactile Thing

Nowadays, comprehensive understanding of human behaviours and external environment is the main goal toward the realization of intimate human-machine interactions and tele-operation services. Of numerous sensing modalities, touch, recently, has been attracted many attentions, due to its intrinsic ability to reveal high-fidelity feedback on direct physical interactions with environment [1]. Smart devices embedded with soft artificial skins, here refers to as *tactile sensing devices* are the key enablers for collecting rich tactile information, that can reach human capabilities of sensing physical contact forces and pressures on the entire body. We envisage that tactile sensing devices, in the next few years, will exist in every consumer products, ranging from large-scale pieces of furniture (bed, sofa), to compact wearable devices [17], and even to tactile robotics [2]. Thus, how to enable the seamless interoperability of the diverse tactile devices over the Web of Thing is a crucial question that needs to be addressed. Toward this goal, based on the prestigious standardization of the W3C WoT, we propose the abstraction for properties and functions of a wide-range of tactile sensors (regardless of sensor scales, and sensing principles), especially ones embedded with soft/compliant artificial skins, so as to enhance the interconnection between tactile devices and possibly the virtual world, recently known as Metaverse.

3 Web of Tactile Thing

3.1 Physical Tactile Sensor

[1] https://www.w3.org/TR/wot-scripting-api/

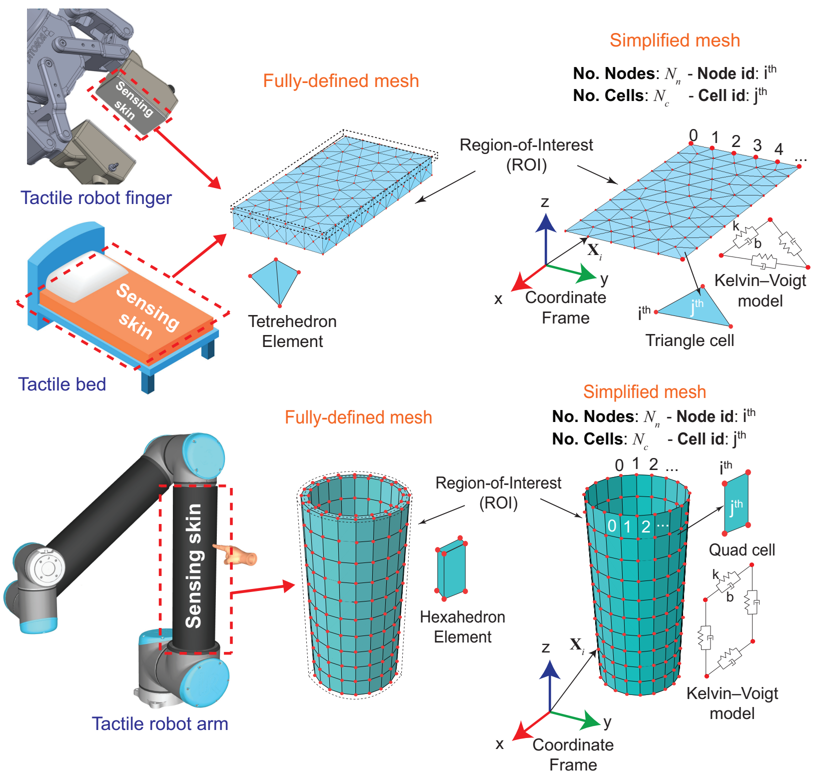

Fig. 3. Topologically meshes are considered as geometrical representation of the diversity tactile skins, which also abstracts the signal transduction method to only focus on the inference of semantic information based on the node displacements.

Soft Sensor Skin Representation Geometric modeling of physical things (e.g, rigid object [10], fluid [8]) in 3-dimensional (3D) space has been growing crucial for a wide range of model-based applications such as graphics and robotics. Technically, every physically compliant object can be represented by a network of a huge number of small geometrical cells constructed from vertices, edges and surfaces as normally called topology or mesh. Material characteristics (skinMaterial TD property) are defined by the connectivity of this network. More specifically, the connection strength between a pair of vertices through an edge can be modeled by Kelvin-Voigt constitutive laws as demonstrated in **Figure** 3. In the landscape of soft sensing skins regardless of sensing principle, a single cell could be varied from very simple elements such as point, lines to complex shapes made of k-triangle cells (triangular or tetrahedron elements) and k-cube cells (quad or hexahedron meshes) as shown in **Figure** 3, that would facilitate finite element analysis of soft bodies. In this work, we abstract the use of a fully-defined geometrical model made of high-order elements (tetrahedron and hexahedron) for studying mechanical behaviors of the touchpad under physical stimulation, while focusing on the use of simplified meshes (formed by either triangular or quad elements) as a region of interest (ROI) of tactile sensing skins. These simplified meshes not only serve as the geometrical representation of diverse tactile skin shapes but also the outputs of high-level touch processing. This allocation is expected to reduce latency starting from encoding, transferring them within the cyber network and finally, decoding at the other end while still holding the feasibility and quality of the tactile information. For the tactile data model, any skin mesh could be defined as the connection of N_c skin cells (skinCells TD property), N_n skin nodes (skinNodes TD property) and their positions (nodeLocation TD property) with respect to the global coordinate system (sensorCoordinateFrame TD property).

Tactile Information Rich tactile information provided by tactile sensors, such as touch detection, contact intensity, and contact postural estimations can be reasonably inferred from deformation of soft sensor skins under interactions made with an external environment. By representing the sensing skin surface of a sensor by a mesh (see **Section** 3.1), we could measure the skin deformation by changes in the entire 3-D coordinates $\mathbf{X} \in \mathbb{R}^{3N}$ of N_n-mesh nodes with respective to its original state $\mathbf{X}_0$, in which the element $\mathbf{X}_i = P(i) \in \mathbb{R}^3$ being the 3-D position of node $i \in \mathcal{N}$ ($|\mathcal{N}| = N_n$). This skin deformation captured by node displacements $\mathbf{D} = \mathbf{X} - \mathbf{X}_0 \in \mathbb{R}^{3N}$ can then be processed to set off the following tactile events.

- **Action Detection** (actionDetection TD event): Action detection is the problem that recognizes which tactile primitive (*e.g.*, touch, slip, and so on) is acting on the tactile sensing devices. This problem can be formulated as a multiclass classification task where given the displacements at every nodes $\mathbf{D} \in \mathbb{R}^{3N}$, the type of tactile primitive detected is output class with the highest probability.
- **Skin Deformed Detection** (skinDeformedDetection TD event): Tactile devices could also provide the information on deformed regions/areas of the skin where physical contacts are made (contact posture). These areas of contacts are determined by the set of deformed nodes that are recognized by their indexes $\mathcal{I}_D = \{i \in \mathcal{I} \mid ||\mathbf{D}_i|| > \epsilon\}$. The contact posture represented by node indexes supposes to provide a more lightweight data packet than that identified by explicit 3-D coordinates of every deformed nodes.

3.2 Tactile Thing Description

Tactile Thing Description (Tactile TD) is proposed to specifically describe the properties and functions of tactile sensors by a standardized JSON-based information model, which is served as a novel protocol for haptics communication over the WoTT. We define the interaction affordances according to the physical properties and touch information that are purposefully provided by the tactile sensing devices.

Property Affordances The property affordance of the Tactile TD encapsulates all the physical attributes of the common vision-based tactile sensors, which especially focuses on the description of functional sensing elements characterized by the compliant/soft elastomeric skins. We propose the following Property Affordances to describe in detail the nature of sensing skins:

- **skinMaterial**: a read-only object datatype that describes the mechanical property of a sensitive skin through its material characteristic, in which **materialType** covers a brand of elastomeric materials commonly used to make artificial sensing skins (*e.g.*, Ecoflex^{TM}, Dragonskin^{TM}), and the information on the softness of material can also be accessed via **materialShoreHardness** string datatype that is possibly to include a variety of measurement scale (*e.g*, Shore 00, Shore A10, Shore D30).
- **skinShape**: that is an object-type property consists of **shapeType** and geometric dimensions specific to each type (*e.g.*, **lengthDimension**, **heightDimension**, **baseRadiusDimension**, and so on), which provide read-only information on the geometrical shape of a sensitive skin. This dataschema is initially employed to describe three popular sensor shapes, including rectangular prism (for flat sensing skin applicable to tactile bed, sofa, and robotic fingers), cylindrical and barrel shapes (for large-scale sensing skin).
- **skinNodes**: an observable property defining the skin surface nodes (see Sec. 3.1) that can represent the highly deformable states of an arbitrary soft sensing skin, extending beyond the basic geometrical shapes (**skinShape**) described above. This object-type property specifies the number of mesh nodes **numOfNodes** and an array of node objects *arrayOfNodes* which contains the identification (ID) number **nodeID** and 3D-coordinates of every nodes **nodeLocation** with respect to a Cartesian reference frame **referenceTo**.
- **skinCells**: a real-only property that represents the topological connectivity between the skin nodes (see Sec. 3.1), which is defined by an object-type dataschema that attributes the number of cells **numOfCells** and an array of cell objects **arrayOfCells** containing the cell ID number **cellID** and a list of node IDs **connectedNodes** that constitute a cell.
- **contactInformation**: this observable property offers comprehensive information related to physical contacts sensed by the sensor, which is the heart of numerous tactile interaction tasks and now is abstracted to propagate over the network of tactile things. **contactLocations** attribute lets the users know which part of the sensor being in contact, and how much the intensity and directional information of such the contacts can be accessed via **contactForces** data which represents the physical quantity of generalized acting forces with respect to the Cartesian space defined in **referenceTo**. Moreover, in order to afford the information on multiple contacts that can be retrieved from state-of-the-art large-scale tactile sensors (see **Section**

Hardware for example), we adopt WOT *ArraySchema* for accommodation of multiple contact forces and locations with the respective number of touch points given in **numOfContacts** attribute.

- **sensorCoordinateFrame**: a user-defined property describing a Cartesian coordinate system that is referenced by all the vector (directional) quantities defined in the Tactile TD, such as node location, contact forces and contact locations. The location of coordinate origin **originLocation** can be defined at feature points within the sensor (e.g., center of gravity, top left corner, and so on), while the direction of three orthogonal axes **axisOrientation** should comply with the right-hand rule, following the common standard of x-axis pointing forward, y-axis left, and z-axis up in relation to the sensor body.

Event Affordance We also attempt to propose abstractions for touch information processing based on the WoT *Event Affordance*, which automatically notifies users of purposeful tactile events with lightweight data packet that can be triggered from general-purpose appliances embedded with tactile sensing skins.

- **actionDetection**: an event, emitting signal defined by the WoT *String schema* string datatype, informs users whatever action of tactile primitives (e.g., "touch", "slip", and so on) acting on the tactile sensing device in a wide range of scenarios. For example, a touch action made on the skin surface of the targeted tactile devices can notify thing consumers of diversity kinds of events, such as when a person lies on bed, sits on sofa, or once an object has been grasped by a robotic finger. On the other hand, slip action can be triggered once a slip or stroke detected on the sensing skin surface, which is one of the most crucial tactile information for robotics grasping, haptics feedback, and for feeling a gentle stroke action on the skin surface as well.
- **skinDeformedDetection**: an event, pushed when there is a change in the skin state, notifies the deformed region of the skin, which is represented by a set of deformed node objects **arrayOfDeformedNodes** each of which contains the intensity of deformation (contact depth) and the ID number of the node being deformed (see **Section** 3.1). The fine deformed shape of the sensing skin obtained from this event benefits applications requiring accurate digital reproduction of such tactile devices (*e.g.*, remote skills transfer) or ones involved to posture analysis, like how a lying posture would affect sleeping quality.

4 Experiment

We public the implementation of the client and server in **Section. 4** as an open-source solution via Github [2]. Our experiment was conducted based on this open-source solution.

[2] https://github.com/Ho-lab-jaist/WoTT

4.1 Soft Vision-based Tactile Sensing Device

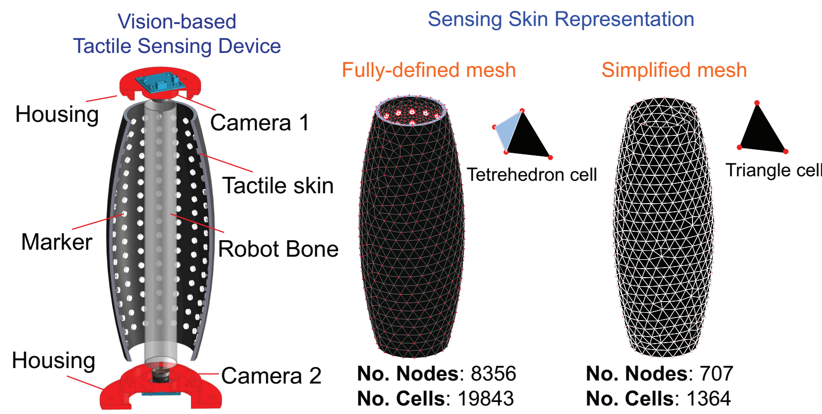

Fig. 4. Structural design of the vision-based tactile sensing device presented in [5], [18] and its presentation in the form of finite element models (skin mesh) which can be defined by skin nodes and cells in the simplified form.

The general structure of a soft vision-based tactile sensor consists of a flexible touch pad (or normally called tactile skin) whose visual cues on it will be revealed by an internal illumination system. When an object comes in contact with the sensitive skin, optical sensor in camera will capture resulted skin's global deformation. Then, upon the variation in light property of the visual cues, an appropriate visual-to-tactile transduction protocol will be utilized. Detailed classification and review on such kind of tactile sensor in the literature up to date were extensively done in [13]. Here, we make use of our previous vision-based tactile sensor introduced in [5],[18], which record motion of markers (printed in the opposite site of the touching surface) during contact phase to infer tactile information, as a showcase for WoTT platform. In other words, we utilized markers as visual mediums to reflect displacement of a point cloud (topological nodes) representing the skin's deformation as well as corresponding mechanical behaviors (e.g, distributed forces) to produce tactile information. Such mechanism allows us to obtain richer sensory feedback that are close to human tactile cognition. Note that the sensory data transduction method proposed in this article should not be limited to marker-based tactile sensors but other visual features such as reflective membrane [9] are also applicable. **Figure** 4 illustrates structural design of our hardware used in this article and its decomposition into different meshes. It comprises a barrel-shaped, black soft tactile skin containing a grid of white markers (16 rows x 16 columns) distributed across the inner surface. Two end of the skin will be fixed by two camera housing bases which are connected with each other through a transparent tube as an artificial bone. Each end accommodates a fish-eye lens CMOS camera (ELP USBFHD01M-L180: resolution 640 x 480 pixels, frame rate 120 fps, field of view 180°) and an array of LEDs to enhance the visibility of markers to optical systems. Fabrication and assembling process can be reviewed more adequately in [16].

Touch Information Processing Here, we demonstrate that a typical touch processing pipeline for the large-scale vision-based tactile sensor can be easily adapted to provide tactile information complying with the data model (Tactile TD) proposed in **Section** 3. The core problem is to extract the displacement of barrel skin nodes (see **Figure** 4) from the cues of marker-featured tactile images. We refined a data-driven method presented in [5] for estimating such node displacements, instead of just markers' deviations, from a pair of tactile images. The dataset was collected based on the process, named fully autonomous generation of displacement and tactile picture (FADP) [5] and consisted of $11,650$ pairs of tactile images (input data) and corresponding nodes' displacement data (output labels), which was split in 8:2 for train and validation dataset. The model was trained for 50 epochs using Stochastic Gradient Descent (SGD) algorithm, with MSE (mean squared error) loss and constant learning rate of 0.01.

4.2 Server-side Setup and Implementation

We implement a WoT *Thing Producer* (server) which acts as a middle-ware to connect tactile sensing devices into the WoTT platform. This server exposes tactile information which is encoded into the tactile TD in **Section** 3.2 and is discoverable by W3C WoT clients. We employed *wotPY* [3] to support W3C WoT run-time, WoT Scripting API, and protocol bindings (*e.g.*, HTTP, Websocket).

In the initialization phase, physical properties of the tactile sensing device, such as skin nodes and skin cells, was retrieved from resource files and taken as input for producing WoT property affordances. For providing services involved to event affordances, the asynchronous IO programming paradigm was implemented to emit simultaneously events applicable to the vision-based sensor (*i.e.*, touchDetection, skinDeformed). Specifically, asynchronous functions was defined to periodically check ($\sim 50\,ms$) for physical touch actions and changes in the skin state, whose information is obtained by invoking the touch information processing routine of the sensor, and then the respective events are triggered and notify subscribers once conditions are satisfied.

4.3 Client-side Setup and Implementation

We designs a WoT client as an application that consumes the tactile TD to construct a three-dimensional digital twin of the physical tactile sensing device. To simultaneously receive data from the server and reproduce a replica of the device,

[3] https://github.com/agmangas/wot-py

which subscribes **skinDeformed** event, an asynchronous input/output processing was implemented in the client-side. This programming pattern allowed the client to wait for incoming information of deformation while independently maintain and update the 3D graph. When booting up the client, an initial digital twin of the sensor is created based on the locations of the skin nodes (**skinNodes** property) in the non-deformed state (decoded from tactile TD). After obtaining the displacement of deformed nodes (**arrayOfDeformedNodes** property), the counterparts in the 3D simulation would deform by the same amount. The region of skin deformation could also present a variation in color corresponding to the displacement intensity, giving a clearer illustration, as shown in **Figure** 5. The WOT client was deployed on machines connected to the server through local area network and Wi-Fi, the end-to-end data transmission time was then recorded and analyzed.

5 Evaluation

5.1 Tactile Thing Description Validity and Bench-marking

The W3C WoT Interest Group (IG) and Working Group(WG) maintain a playground [4] to validate TD(s). The auto-generated tactile TD by our server was confirmed to be a validated WoT TD and is able to use by any WoT client. Additionally, we utilized a bench-marking tool, named *testbench* [11], created by the Technical University of Munich to further verify the client-server interactions. As the result, we received a **100 %** passed result for every pre-configured test case. These results proved the correct operations and validity of our proposed solution.

5.2 End-to-End Delay

The end-to-end delay was evaluated by an experiment of measuring the total time for processing, transmitting tactile data and visualizing a digital twin of the tactile sensing device via the WoTT platform. The necessary time of each step of a complete process is presented in **Figure** 5, where Δt_1 is the time for the touch signal to reach the server's computer, Δt_2 is the time needed for the transforming signal into tactile TD, Δt_3 is the network transmission time, Δt_4 is the period required for the client to consume the tactile TD, and Δt_5 is the time taken to visualize the digital twin.

The experiment was performed by a human touching the tactile sensing device's skin, the test run was repeated ten times and the average time of each step was recorded. The time needed to initialize the application on the client-side was also measured ten times and taken average. The experiments were conducted on three different devices, a PC and a Jetson Nano Developer Kit connected to the sever's machine through Ethernet, and a laptop communicating via Wi-Fi. According to the results shown in **Table** 1, the Jetson Nano Kit took significantly

[4] http://plugfest.thingweb.io/playground/

longer to accomplish the task, while the end-to-end delay for the PC and laptop are approximately the same. The difference was mainly due to the inferior computational capability, especially in the visualization step, of the Jetson Nano Kit. The total execution time for the PC and laptop are in acceptable range, which proves the feasibility of the WoTT platform in interconnecting tactile devices and applications.

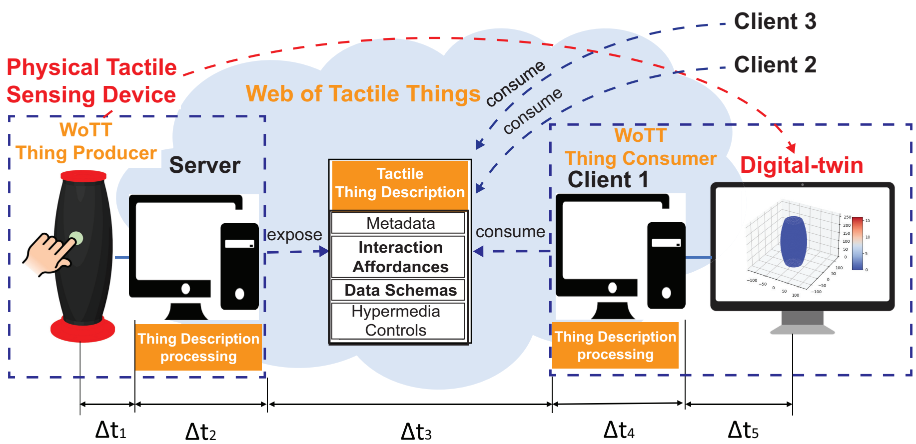

Fig. 5. An example use-case of the WoTT. A WoT client is able to create a digital twin of a tactile sensing device connected to the WoTT. End-to-End delay time of represent a touch interaction (by human) is visualized.

Table 1. End-to-end delay result of WoTT system.

Device	Init. time	Δt_1	Δt_2	Δt_3	Δt_4	Δt_5	End-to-end delay
PC	1.033 s	0.040 s	0.020 s	0.023 s	0.056 s	0.091 s	0.230 s
Jetson Nano	3.921 s	0.178 s	0.061 s	0.466 s	0.446 s	0.708 s	1.859 s
Laptop Wi-Fi	0.782 s	0.020 s	0.017 s	0.035 s	0.021 s	0.106 s	0.199 s

6 Concluding Remarks

Towards the development of the *Metaverse* and the Tactile Internet, interconnecting tactile things into cyberspace will be emerging. Furthermore, in order to counter the fragmentation caused by vertical integrations in the current IoT systems, we propose an idea on a platform for tactile things by utilizing the success of the Web paradigm. This paper introduces an extension for W3C WoT that supports tactile devices to create a platform (named WoTT) to interconnect tactile devices and applications via W3C WoT abstractions. We introduce a novel information model to map physical tactile sensing devices into digital forms and it can represent all characteristics and events (human interactions) of physical devices. The proposed information model was validated by tools provided by W3C WoT Working Group/Interest Group members. To prove the feasibility of our solution, we conduct an experiment that uses a physical tactile sensing device and a digital twin of it. We were able to reproduce the physical devices and reflect haptic information via the W3C WoT. The overhead added by the extension is relatively small compared to the benefit that allows Web developers to join and develop applications for the Tactile Internet. In the future, we plan to leverage this approach toward generalized ontology model for tactile sensing devices, implement data transmission with low latency, and evaluate on actual robotic applications using WoTT.

Acknowledgements This work was fully supported by JST Precursory Research for Embryonic Science and Technology PRESTO under Grant JPMJPR2038.

References

1. Altun, K., MacLean, K.E.: Recognizing affect in human touch of a robot. Pattern Recognition Letters **66**, 31–40 (2015). https://doi.org/https://doi.org/10.1016/j.patrec.2014.10.016, https://www.sciencedirect.com/science/article/pii/S016786551400333X, pattern Recognition in Human Computer Interaction
2. Dahiya, R.S., Metta, G., Valle, M., Sandini, G.: Tactile sensing—from humans to humanoids. IEEE Transactions on Robotics **26**(1), 1–20 (2010). https://doi.org/10.1109/TRO.2009.2033627
3. Droettboom, M., et al.: Understanding json schema. Available on: https://json-schema.org/understanding-json-schema/UnderstandingJSONSchema.pdf (accessed on 23 November 2021) (2021)
4. Fanibhare, V., Sarkar, N.I., Al-Anbuky, A.: A survey of the tactile internet: Design issues and challenges, applications, and future directions. Electronics **10**(17) (2021). https://doi.org/10.3390/electronics10172171, https://www.mdpi.com/2079-9292/10/17/2171
5. Ho, V.A., Nakayama, S.: Iotouch: whole-body tactile sensing technology toward the tele-touch. Advanced Robotics **35**(11), 685–696 (2021). https://doi.org/10.1080/01691864.2021.1925588
6. Ibaseta, D., García, A., Álvarez, M., Garzón, B., Díez, F., Coca, P., Pero, C.D., Molleda, J.: Monitoring and control of energy consumption in buildings using wot: A novel approach for smart retrofit. Sustainable Cities and Society **65**, 102637 (2021). https://doi.org/https://doi.org/10.1016/j.scs.2020.102637, https://www.sciencedirect.com/science/article/pii/S2210670720308532

7. Johansson, R.S., Westling, G.: Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects. Experimental Brain Research **56**(3), 550–564 (1984). https://doi.org/10.1007/BF00237997, https://doi.org/10.1007/BF00237997
8. Kambe, T.: Introduction to a Geometrical Theory of Fluid Flows and Dynamical Systems, pp. 137–165. Springer Netherlands, Dordrecht (2001). https://doi.org/10.1007/978-94-010-0446-6_7, https://doi.org/10.1007/978-94-010-0446-6_7
9. Li, R., Platt, R., Yuan, W., ten Pas, A., Roscup, N., Srinivasan, M.A., Adelson, E.: Localization and manipulation of small parts using gelsight tactile sensing. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 3988–3993 (2014). https://doi.org/10.1109/IROS.2014.6943123
10. Rashid, M., Gullett, P.: On a finite element method with variable element topology. Computer Methods in Applied Mechanics and Engineering **190**(11), 1509–1527 (2000). https://doi.org/https://doi.org/10.1016/S0045-7825(00)00175-4, https://www.sciencedirect.com/science/article/pii/S0045782500001754
11. Schlott, V.E., Korkan, E., Kaebisch, S., Steinhorst, S.: W-ade: Timing performance benchmarking in web of things. In: Bielikova, M., Mikkonen, T., Pautasso, C. (eds.) Web Engineering. pp. 70–86. Springer International Publishing, Cham (2020)
12. Sciullo, L., Bhattacharjee, S., Kovatsch, M.: Bringing deterministic industrial networking to the w3c web of things with tsn and opc ua. In: Proceedings of the 10th International Conference on the Internet of Things. IoT '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3410992.3410997, https://doi.org/10.1145/3410992.3410997
13. Shah, U.H., Muthusamy, R., Gan, D., Zweiri, Y., Seneviratne, L.: On the Design and Development of Vision-based Tactile Sensors. Journal of Intelligent & Robotic Systems **102**(4), 82 (2021). https://doi.org/10.1007/s10846-021-01431-0, https://doi.org/10.1007/s10846-021-01431-0
14. Sharma, S.K., Woungang, I., Anpalagan, A., Chatzinotas, S.: Toward tactile internet in beyond 5g era: Recent advances, current issues, and future directions. IEEE Access **8**, 56948–56991 (2020). https://doi.org/10.1109/ACCESS.2020.2980369
15. Tjew A Sin, M., Koole, S.: That human touch that means so much: Exploring the tactile dimension of social life. In-Mind (Deutsche Ausgabe) **2013**(17), 1–17 (2013)
16. Van Duong, L., Ho, V.A.: Large-scale vision-based tactile sensing for robot links: Design, modeling, and evaluation. IEEE Transactions on Robotics **37**(2), 390–403 (2021). https://doi.org/10.1109/TRO.2020.3031251
17. Yang, T., Xie, D., Li, Z., Zhu, H.: Recent advances in wearable tactile sensors: Materials, sensing mechanisms, and device performance. Materials Science and Engineering: R: Reports **115**, 1–37 (2017). https://doi.org/https://doi.org/10.1016/j.mser.2017.02.001, https://www.sciencedirect.com/science/article/pii/S0927796X16301231
18. Yoshigi, S., Wang, J., Nakayama, S., Ho, V.A.: Deep learning-based whole-arm soft tactile sensation. In: 2020 3rd IEEE International Conference on Soft Robotics (RoboSoft). pp. 132–137 (2020). https://doi.org/10.1109/RoboSoft48309.2020.9116018
19. Zou, L., Ge, C., Wang, Z.J., Cretu, E., Li, X.: Novel tactile sensor technology and smart tactile sensing systems: A review. Sensors **17**(11) (2017). https://doi.org/10.3390/s17112653, https://www.mdpi.com/1424-8220/17/11/2653