



**HAL**  
open science

## **$\mu$ Poly : une Boîte à Outils pour Concevoir l'Interaction par Microgeste**

Adrien Chaffangeon Caillet, Aurélien Conil, Alix Goguey, Laurence Nigay

### ► To cite this version:

Adrien Chaffangeon Caillet, Aurélien Conil, Alix Goguey, Laurence Nigay.  $\mu$ Poly : une Boîte à Outils pour Concevoir l'Interaction par Microgeste. IHM'24 - 35e Conférence Internationale Francophone sur l'Interaction Humain-Machine, AFIHM; Sorbonne Université, Mar 2024, Paris, France. hal-04499957

**HAL Id: hal-04499957**

**<https://hal.science/hal-04499957>**

Submitted on 11 Mar 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# **μPoly : une Boîte à Outils pour Concevoir l'Interaction par Microgeste**

## **μPoly: a Toolkit to Design Microgesture Interaction**

ADRIEN CHAFFANGEON CAILLET\*, Université Grenoble Alpes, LIG, CNRS, France

AURÉLIEN CONIL\*, Université Grenoble Alpes, LIG, CNRS, France

ALIX GOGUEY, Université Grenoble Alpes, LIG, CNRS, France

LAURENCE NIGAY, Université Grenoble Alpes, LIG, CNRS, France

De nombreux systèmes de reconnaissance de microgestes ont été proposés. Ces systèmes diffèrent par leurs formes, leurs types de capteurs et leurs algorithmes de reconnaissance. Cependant, en l'absence d'un standard d'événement de type microgeste et d'une boîte à outils pour gérer l'interaction par microgestes, il est difficile pour un concepteur d'interaction de tester facilement et rapidement différents systèmes de reconnaissance et ensembles de microgestes. Nous proposons μPoly, une boîte à outils basée sur μGlyph, une notation pour décrire les microgestes, pour répondre à ce problème.

Numerous microgesture recognition systems have been proposed. These systems differ in shape, sensor types and recognition algorithms. However, in the absence of a microgesture event standard and a toolbox for microgesture interaction, it is difficult for an interaction designer to easily and quickly test different recognition systems and microgesture sets. We propose μPoly, a toolbox based on μGlyph, a notation for describing microgestures, to address this problem.

Mots Clés et Phrases Supplémentaires: Boîte à outils, Microgeste

Additional Key Words and Phrases: Toolkit, Microgesture

### **Reference:**

Adrien Chaffangeon Caillet, Aurélien Conil, Alix Goguey, and Laurence Nigay. 2024. μPoly : une Boîte à Outils pour Concevoir l'Interaction par Microgeste. *IHM '24 : Actes étendus de la 35<sup>e</sup> conférence Francophone sur l'Interaction Humain-Machine, March 25–29, 2024, Paris, France*

## **1 INTRODUCTION**

Ces dernières années, plusieurs systèmes ont été conçus pour détecter et reconnaître les mouvements rapides et subtils des doigts, appelés microgestes. Grâce à ces nouveaux systèmes, de nouvelles techniques d'interaction basées sur les microgestes ont été conçues et testées [4, 7, 13]. En particulier, les microgestes ont été étudiés pour leur capacité à être toujours disponibles, même en tenant ou non un objet, afin d'interagir dans divers contextes (par exemple, la réalité mixte, le cockpit, les voitures). Les systèmes de reconnaissance étant de plus en plus nombreux, il peut être difficile de trouver et de tester le meilleur système de reconnaissance pour un ensemble particulier de microgestes dans un contexte particulier.

Ces systèmes sont composés d'un dispositif de captation, qui se présente sous différentes formes (par exemple, gants [13], bagues [1], bracelets [9]) et contient différents capteurs (e.g., pression [13], centrale à inertie [11], caméra [12]), et d'un algorithme de reconnaissance associé (par exemple, ad-hoc [13], SVM [9], Forêt d'arbres décisionnels [11]). Chaque système a ses avantages et ses inconvénients. Par exemple, un système de suivi des mouvements (*motion capture*) permet un suivi précis des parties de la main, permettant de reconnaître des ensembles variés de microgestes. Cependant, il implique un volume de captation contrôlé et fixe, difficilement généralisable à tous les contextes. A l'opposé, une

---

\*. Both authors contributed equally to this research.

baguette semble plus adaptée à une utilisation ubiquitaire. Cependant, sa forme contraint les types et nombre de capteurs utilisables, ce qui implique généralement un ensemble restreint de microgestes. Par conséquent, un concepteur, ou un chercheur, souhaitant concevoir des interactions avec des microgestes, est amené à tester différents systèmes de reconnaissance avant de trouver le système de reconnaissance adéquat à l'ensemble de microgestes et au contexte considérés.

En l'absence de normalisation des systèmes de reconnaissance et des événements de type microgeste, ce processus d'exploration du système de reconnaissance adéquat aux besoins peut rapidement devenir difficile et chronophage. Il est donc important de proposer un outil permettant de facilement définir un ensemble de microgestes à reconnaître ainsi que de prototyper et essayer rapidement différents systèmes de reconnaissance. Ce sont des éléments clés pour faire progresser le domaine de l'interaction par microgestes [8].

Récemment, Chaffangeon Caillet et al. ont introduit la première notation pour les microgestes :  $\mu$ Glyphe [5]. Cette notation permet de décrire un microgeste à différents niveaux d'abstraction. Au plus bas niveau,  $\mu$ Glyphe est basé sur un ensemble d'événements dérivés de la biomécanique de la main. Dans cette démonstration, nous présentons  $\mu$ Poly, une boîte à outils basée sur  $\mu$ Glyphe.  $\mu$ Poly reçoit en entrée des microgestes élémentaires, c.-à-d. des microgestes composés d'un seul événement de bas niveau sous leur forme  $\mu$ Glyphe.  $\mu$ Poly utilise, ensuite, ces microgestes élémentaires pour parcourir des machines à états représentant des descriptions  $\mu$ Glyphe de microgestes plus complexes c.-à-d. composés de plusieurs événements de bas niveau. Enfin,  $\mu$ Poly utilise un système de souscription d'événements pour envoyer les microgestes reconnus aux applications clientes abonnées.

Nous identifions trois avantages de  $\mu$ Poly pour un concepteur souhaitant utiliser les microgestes. **1/** En imposant une norme ( $\mu$ Glyphe) sur le format des événements microgestes,  $\mu$ Poly permet de tester facilement et de manière transparente différents systèmes de reconnaissance. **2/** Avec l'utilisation d'événements  $\mu$ Glyphe élémentaires et de machines à états,  $\mu$ Poly permet une distinction claire entre le système de reconnaissance et les ensembles de microgestes à reconnaître, et ainsi offre une plus grande flexibilité pour tester différents ensembles de microgestes avec le même système. **3/**  $\mu$ Poly intègre une bibliothèque de microgestes et d'applications exemples.

## 2 PRINCIPAUX COMPOSANTS DE $\mu$ POLY

$\mu$ Poly est une boîte à outils pour les microgestes programmée avec NodeJS et VueJS.

### 2.1 Événements de bas niveau

$\mu$ Poly prend en entrée des microgestes élémentaires pour permettre une plus grande flexibilité lors de la phase de reconnaissance de microgestes composés. En effet, les systèmes de reconnaissance de microgestes existants sont entraînés pour reconnaître des microgestes composés, mais rarement des microgestes élémentaires. Toutefois, cela offrirait une plus grande flexibilité, à l'instar des événements souris offrant l'accès aux événements composés (p.ex. click) et élémentaires (p.ex. down, move, up). Cette approche est particulièrement intéressante car de nombreux microgestes composés partagent les mêmes microgestes élémentaires [5, annexes]. Par exemple, l'ensemble :  $\downarrow$ ,  $\uparrow$ ,  $\leftarrow$ ,  $\rightarrow$ , permet de construire au moins 10 microgestes de la littérature :  $\downarrow$ [6],  $\uparrow$ [15],  $\downarrow$ : $\uparrow$ [10],  $\leftarrow$ [10],  $\rightarrow$ [10],  $\leftarrow$ : $\downarrow$ [14],  $\downarrow$ : $\uparrow$ [14],  $\leftarrow$ : $\uparrow$ [14],  $\downarrow$ : $\leftarrow$ [13], et  $\downarrow$ : $\rightarrow$ : $\uparrow$ [13].

### 2.2 Ajouter un nouveau système de reconnaissance

Pour ajouter un système de reconnaissance à  $\mu$ Poly, le concepteur du système doit créer une classe *pilote* dédiée. Cette classe, dont la structure est fournie, transforme les données de sortie du système en une structure de données

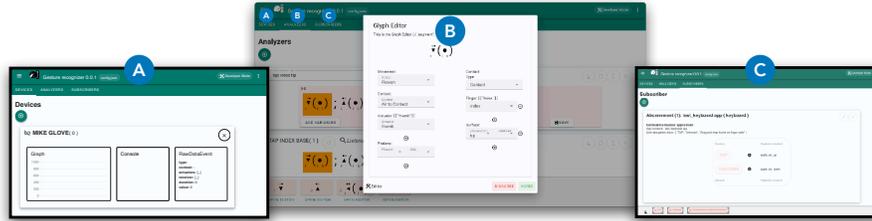


FIG. 1 – Capture d’écran de μPoly. A) Panneau pour sélectionner les systèmes de reconnaissance. B) Panneau pour ajouter des microgestes à reconnaître. C) Panneau pour ajouter les applications clientes.

représentant des microgestes élémentaires. Cette classe spécifie également la manière dont le système communique avec μPoly (serial port, rawHID ou Bluetooth). Les détails sur la création d’une classe *pilote* sont disponibles sur [GitLab \[2\]](#). Un utilisateur peut ensuite sélectionner un système de reconnaissance dans l’interface de μPoly (fig.1A).

### 2.3 Ajouter un microgeste à reconnaître

Pour ajouter un microgeste à reconnaître, l’utilisateur spécifie la description μGlyph de ce microgeste, en utilisant un éditeur visuel pour la notation μGlyph (fig.1B). Pour faciliter l’utilisation de μPoly par les personnes non familiarisées avec μGlyph, μPoly propose une bibliothèque de microgestes courants et décrits par leur nom et leur image. D’autre part, μPoly est compatible avec les descriptions μGlyph de μDex [3], une base de données de plus de 100 microgestes issus de la littérature, comprenant une description textuelle et/ou une image du microgeste. Un utilisateur peut directement copier/coller la description de μDex dans l’éditeur de μPoly.

### 2.4 S’abonner à et recevoir un microgeste

L’abonnement à un microgeste de μPoly se fait via des *websocket*, déjà implémentées dans un grand nombre de langages de programmation. μPoly joue le rôle du serveur *websockets* et les applications tierces le rôle de client. Pour s’abonner à un microgeste, l’utilisateur déclare une nouvelle application dans l’interface μPoly (fig.1C) en spécifiant le port de la *websocket*, puis sélectionne les microgestes auxquels l’application s’abonne. Pour faciliter le développement, μPoly permet de simuler des microgestes indépendamment du système de reconnaissance.

## 3 CLASSES PILOTES ET APPLICATIONS TIERCES

μPoly propose des classes *pilotes* pour différents systèmes de reconnaissance et des application tierces exemples.

### 3.1 Exemples de Classes *pilotes*

**Gant capteurs industriels**, un système de reconnaissance utilisé dans [13]. Il contient 8 capteurs de pression et 2 de flexion, tous achetés dans le commerce. Il utilise un algorithme ad-hoc pour reconnaître l’ensemble des taps du pouce vers les phalanges de l’index et du majeur ainsi que l’ongle de l’annulaire.

**Gant capteurs velostat**, un système de reconnaissance utilisé dans [4]. Il contient 2 capteurs de pression, fait main avec du Velostat, et 1 capteur de flexion, acheté dans le commerce. Il utilise un algorithme ad-hoc pour reconnaître des taps du pouce vers les phalanges de l’index ainsi que si l’index est ouvert, entrouvert ou fermé.

**Gant capteur fil conducteur**, un système de reconnaissance utilisé dans [7]. Il contient 19 capteurs de contact, fait main avec du fil de couture conducteur. Il utilise un algorithme ad-hoc pour reconnaître des taps et double taps du pouce sur les phalanges des autres doigts ainsi que des swipes du pouces sur les autres doigts.

### 3.2 Exemples d'Applications tierces

**Raccourci clavier** Dans cette application, les microgestes émulent des touches ou combinaisons de touches clavier, par exemple, les flèches directionnelles pour contrôler une présentation.

**VLC** Via l'API de contrôle VLC, cette application associe des microgestes aux fonctions de contrôle du lecteur de vidéo (comme lecture, pause et volume).

## 4 CONCLUSION

Nous avons présenté  $\mu$ Poly, une boîte à outils pour le développement de techniques d'interaction basées sur les microgestes. Nous avons montré comment  $\mu$ Poly facilite la conception d'applications utilisant des microgestes en permettant : **1/** un changement rapide du système de reconnaissance des microgestes, **2/** une édition/modification facile de l'ensemble des microgestes à reconnaître.

## RÉFÉRENCES

- [1] Roger Boldu, Alexandru Dancu, Denys J.C. Matthies, Pablo Gallego Cascón, Shanaka Ransir, and Suranga Nanayakkara. 2018. Thumb-In-Motion : Evaluating Thumb-to-Ring Microgestures for Athletic Activity. In *Proceedings of the 2018 ACM Symposium on Spatial User Interaction* (Berlin, Germany) (SUI '18). Association for Computing Machinery, New York, NY, USA, 150–157. <https://doi.org/10.1145/3267782.3267796>
- [2] Adrien Chaffangeon Caillet. 2023. GitLab  $\mu$ Poly. <https://gricad-gitlab.univ-grenoble-alpes.fr/mic/gestureRecognizer>. [Online; accessed 12-January-2024].
- [3] Adrien Chaffangeon Caillet. 2023.  $\mu$ Dex. <https://microglyph.imag.fr/>. [Online; accessed 12-January-2024].
- [4] Adrien Chaffangeon Caillet, Alix Goguey, and Laurence Nigay. 2023. 3D Selection in Mixed Reality : Designing a Two-Phase Technique To Reduce Fatigue. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 800–809. <https://doi.org/10.1109/ISMAR59233.2023.00095>
- [5] Adrien Chaffangeon Caillet, Alix Goguey, and Laurence Nigay. 2023.  $\mu$ Glyph : A Microgesture Notation. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 269, 28 pages. <https://doi.org/10.1145/3544548.3580693>
- [6] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. 2016. User Elicitation on Single-Hand Microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (CHI '16). Association for Computing Machinery, New York, NY, USA, 3403–3414. <https://doi.org/10.1145/2858036.2858589>
- [7] Gauthier Robert Jean Faisandaz, Alix Goguey, Christophe Jouffrais, and Laurence Nigay. 2022. Keep in Touch : Combining Touch Interaction with Thumb-to-Finger  $\mu$ Gestures for People with Visual Impairment. In *Proceedings of the 2022 International Conference on Multimodal Interaction* (Bengaluru, India) (ICMI '22). Association for Computing Machinery, New York, NY, USA, 105–116. <https://doi.org/10.1145/3536221.3556589>
- [8] Brad Myers, Scott E. Hudson, and Randy Pausch. 2000. Past, Present, and Future of User Interface Software Tools. *ACM Trans. Comput.-Hum. Interact.* 7, 1 (mar 2000), 3–28. <https://doi.org/10.1145/344949.344959>
- [9] T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. 2009. Enabling Always-Available Input with Muscle-Computer Interfaces. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (Victoria, BC, Canada) (UIST '09). Association for Computing Machinery, New York, NY, USA, 167–176. <https://doi.org/10.1145/1622176.1622208>
- [10] Adwait Sharma, Michael A. Hedderich, Divyanshu Bhardwaj, Bruno Fruchard, Jess McIntosh, Aditya Shekhar Nittala, Dietrich Klakow, Daniel Ashbrook, and Jürgen Steimle. 2021. SoloFinger : Robust Microgestures While Grasping Everyday Objects. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 744, 15 pages. <https://doi.org/10.1145/3411764.3445197>
- [11] Adwait Sharma, Christina Salchow-Hömmen, Vimal Suresh Mollyn, Aditya Shekhar Nittala, Michael A. Hedderich, Marion Koelle, Thomas Seel, and Jürgen Steimle. 2023. SparseIMU : Computational Design of Sparse IMU Layouts for Sensing Fine-grained Finger Microgestures. *ACM Transactions on Computer-Human Interaction* 30, 3 (jun 2023), 1–40. <https://doi.org/10.1145/3569894>
- [12] Mohamed Soliman, Franziska Mueller, Lena Hegemann, Joan Sol Roo, Christian Theobalt, and Jürgen Steimle. 2018. FingerInput : Capturing Expressive Single-Hand Thumb-to-Finger Microgestures. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces* (Tokyo, Japan) (ISS '18). Association for Computing Machinery, New York, NY, USA, 177–187. <https://doi.org/10.1145/3279778.3279799>

- [13] Jérémy Wambecke, Alix Goguey, Laurence Nigay, Lauren Dargent, Daniel Hauret, Stéphanie Lafon, and Jean-Samuel Louis de Visme. 2021. M[Eye]Cro : Eye-Gaze+Microgestures for Multitasking and Interruptions. *Proc. ACM Hum.-Comput. Interact.* 5, EICS, Article 210 (may 2021), 22 pages. <https://doi.org/10.1145/3461732>
- [14] Katrin Wolf, Anja Naumann, Michael Rohs, and Jörg Müller. 2011. A taxonomy of microinteractions : Defining microgestures based on ergonomic and scenario-Dependent requirements. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6946 LNCS, PART 1 (2011), 559–575. [https://doi.org/10.1007/978-3-642-23774-4\\_45](https://doi.org/10.1007/978-3-642-23774-4_45)
- [15] Katrin Wolf, Robert Schleicher, Sven Kratz, and Michael Rohs. 2013. Tickle : A surface-independent interaction technique for grasp interfaces. *TEI 2013 - Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (2013), 185–192. <https://doi.org/10.1145/2460625.2460654>