

Augmentation de documents et d'applications par analyse de pixels

Sujet Master 2

Encadrants: Géry Casiez (gery.casiez@univ-lille1.fr), Sylvain Malacria (sylvain.malacria@inria.fr) et Edward Lank (lank@uwaterloo.ca)

Contexte

De nombreux documents (par exemple pdf) présentent du contenu qu'il n'est pas possible de modifier facilement sans posséder les sources. Il en est de même pour une grande majorité des des applications que nous utilisons tous les jours qui présentent des interfaces impossibles à personnaliser. En effet, la plupart des widgets sont prédéfinis par les concepteurs de l'application, que ce soit pour les items d'une barre d'outils ou les éléments d'un menu.

L'objectif général de ce projet est d'enrichir des documents existants ou de modifier l'interface d'applications existantes, sans modification de ces documents ou applications. Le principe est de capturer en temps réel le contenu affiché à l'écran et d'y appliquer des algorithmes de vision par ordinateur pour localiser et augmenter certaines parties de l'interface.

Un premier scénario d'application est le suivant : un utilisateur A a mis en ligne un document pdf qui présente différentes figures dont un graphique de type histogramme. Ce même utilisateur a également créé, mis en ligne, et associé à la figure une version interactive de ce graphique (en utilisant D3.js par exemple) qui permet notamment de modifier les axes, échelles et les séries affichées. Un utilisateur B télécharge le document pdf mis en ligne par l'utilisateur A. Lors de l'affichage de la figure, le système détecte que l'utilisateur A a créé une version graphique interactive et la superpose automatiquement sur l'histogramme. L'utilisateur peut alors interagir avec la version interactive du graphique comme s'il avait toujours été intégré au document d'origine. Les opérations de défilement et zoom du document sont prises en compte par le système pour mettre à jour en temps réel le contenu à superposer sur le document pdf.

Un second scénario est le suivant : un utilisateur veut modifier le positionnement des éléments présents dans la barre d'outils d'un logiciel. Le système détecte automatiquement la barre d'outils quand la fenêtre correspondante du logiciel est affichée. Il superpose alors une autre barre d'outils définie par l'utilisateur. Lors de l'interaction avec la barre d'outils augmentée, les événements sont retransmis aux widgets de l'application pour obtenir le comportement désiré.

D'autres scénarios sont envisageables comme l'aide à l'utilisation de logiciels en superposant des éléments graphiques indiquant à l'utilisateur dans quel ordre utiliser les widgets de l'interface. D'autres possibilités sont l'intégration de sources d'informations diverses à toute application comme l'affichage de la météo du jour dans une application de lecture de courriels.

Ces travaux sont connexes à d'autres travaux publiés dans la littérature comme Métisse [1], Webstrates [2], WinCuts [3], Facade [4]. Les travaux les plus proches de la littérature sont ceux sur Prefab [5, 6, 7].

Le projet commencera par la réalisation d'un état de l'art des travaux proches de la littérature. Ensuite le principal objectif du projet sera de mettre en place un démonstrateur permettant de faire de la capture d'écran en temps réel, effectuer de la reconnaissance d'images à l'aide d'algorithmes de vision par ordinateur, intégrer du contenu interactif et faire de la redirection d'événements. Suivant l'état d'avancement du projet, le but final sera d'intégrer le travail réalisé dans un serveur X sous Linux et d'écrire une toolkit pour faciliter le développement d'applications basées sur l'analyse et la superposition de contenu interactif sur un écran virtuel de bureau.

Bibliographie

- [1]- Olivier Chapuis and Nicolas Roussel. 2005. Metisse is not a 3D desktop!. In Proceedings of the 18th annual ACM symposium on User interface software and technology (UIST '05).
- [2]- Clemens N. Klokrose, James R. Eagan, Siemen Baader, Wendy Mackay, and Michel Beaudouin-Lafon. 2015. *Webstrates: Shareable Dynamic Media*. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (UIST '15).
- [3]- Desney S. Tan, Brian Meyers, and Mary Czerwinski. 2004. WinCuts: manipulating arbitrary window regions for more effective use of screen space. In CHI '04 Extended Abstracts on Human Factors in Computing Systems (CHI EA '04).
- [4]- Wolfgang Stuerzlinger, Olivier Chapuis, Dusty Phillips, and Nicolas Roussel. 2006. User interface façades: towards fully adaptable user interfaces. In Proceedings of the 19th annual ACM symposium on User interface software and technology (UIST '06).
- [5]- Morgan Dixon and James Fogarty. 2010. Prefab: implementing advanced behaviors using pixel-based reverse engineering of interface structure. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10).
- [6]- Morgan Dixon, Gierad Laput, and James Fogarty. 2014. Pixel-based methods for widget state and style in a runtime implementation of sliding widgets. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14).
- [7]- Morgan Dixon, Alexander Nied, and James Fogarty. 2014. Prefab layers and prefab annotations: extensible pixel-based interpretation of graphical interfaces. In Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST '14).