HAL
open science

# Visualisation de Traces d'Exécution et Limite de Perception

Rémy Dautriche

## To cite this version:

Rémy Dautriche. Visualisation de Traces d'Exécution et Limite de Perception. IHM'14, 26e conférence francophone sur l'Interaction Homme-Machine, Oct 2014, Lille, France. pp.69-70, 2014. hal-01089638

**HAL Id: hal-01089638**

**https://hal.science/hal-01089638v1**

Submitted on 2 Dec 2014

**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.

# Visualisation de Traces d'Exécution et Limite de Perception

Rémy Dautriche
UJF-Grenoble 1
Laboratoire d'Informatique de Grenoble
UMR 5217 F-38041 Grenoble, France
remy.dautriche@imag.fr

## I. Introduction

L'analyse de données temporelles est transversale à de nombreux domaines (e.g. médical, finance, météorlogie, industriel, ). Ces données sont aujourd'hui générées en grandes quantités par de multiples sources pouvant être de natures très hétérogènes.

Le but recherché en analysant des données temporelles est de découvrir un comportement permettant d'établir un modèle (e.g. météorlogique, finance) et d'identifier des comportements anormaux, ou des erreurs dans les données. L'identification d'*outliers* peut se faire à l'aide d'algorithmes de fouille de données ou avec des outils de visualisation.

Dans le cadre de ma thèse CIFRE, effectuée au sein du Laboratoire d'Informatique de Grenoble et de STMicroelectronics, je m'intéresse plus particulièrement aux traces générées durant l'exécution d'applications multimedia sur systèmes embarqués. Une trace d'exécution est typiquement composée d'évènements bas niveau qui ont eu lieu durant le décodage d'un flux multimédia (e.g. films, télévision). Chaque évènement est daté et est décrit par son type ainsi que par un nombre variable d'arguments dépendant du type d'évènement. Listing 1. montre un extrait de trace d'exécution. La première ligne représente un changement de contexte passant du processus `1023` au processus `4259`. La seconde ligne est une entrée de fonction se trouvant à l'adresse `0x434` et prenant un argument à l'adresse mémoire `0x98b`. La troisième ligne correspond à la sortie de cette fonction. La durée d'une exécution peut s'étendre sur plusieurs heures, générant de l'ordre de 100000 évènements par minute.

```
 ,
[...]
12.3243245      C       1023 4259
12.3455465      F       0x434 0x98b
12.3456546      f       0x434
[...]
```

Listing 1. Extrait d'une trace d'exécution

L'analyse des traces d'exécution permet le déboggage de l'application multimédia en cours de développement ainsi que d'apporter des optimisations en étudiant son comportement au cours de l'exécution. La complexité croissante des plateformes matérielles ainsi que des nouveaux formats d'encodage font croître la taille des traces, rendant le développement des applications multimédia difficile et coûteux. La méthode de travail actuelle utilisée par les développeurs est composée de deux étapes principales : (1) récolter des informations générales de haut niveau, comme par exemple la charge globale du système, puis (2) utiliser des outils très détaillés permettant une analyse très fine de la trace. De part la quantité d'information, l'analyse fine de la trace s'avère fastidieuse. La Figure 1 montre un problème fréquemment rencontré lorsque trop d'évènements sont affichés simultanément : un phénomène d'aliasing apparaît rendant la lecture impossible.

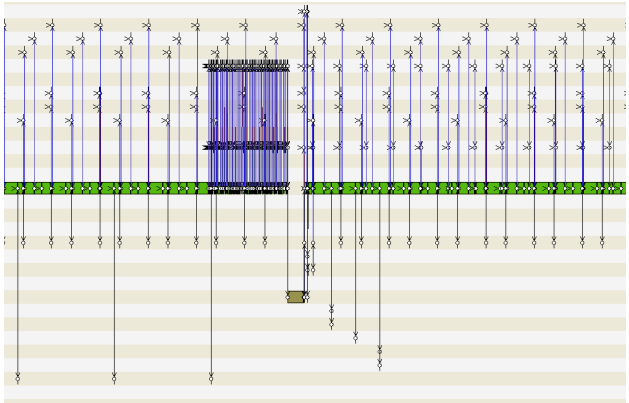

FIGURE 1. Diagramme de Gantt réprésentant les évènements entre les acteurs

Un des défis à relever est de proposer un outil fournissant une première vue de la trace d'exécution permettant d'observer des macro-comportements et de réaliser un filtrage des évènements avant de passer à une analyse fine. Les objectifs de cette visualisation sont multiples :

1) proposer une vue d'ensemble de l'exécution ;
2) identifier les motifs répétitifs dans la trace ; et
3) identifier les comportements anormaux qui se sont produits durant l'exécution.

A partir de ces informations, le développeur sera en mesure de filtrer les évènements, les acteurs et les zones temporelles, réduisant de façon significative la taille des données, avant de passer à une analyse approfondie des problèmes identifiés.

FIGURE 2. Extrait d'une vue globale de trace d'exécution.

De nombreux travaux ont déjà porté sur la visualisation de multiples séries temporelles. KronoMiner [7], SignalLens [5] et BinX [1] permettent à l'utilisateur de choisir le niveau d'abstraction, définissant ainsi les différentes aggrégations temporelles. Horizon Graph [3], et plus récemment Interactive Horizon Graph [6], Ripple Graph [2] et Braided Graph [4] sont des visualisations compactes permettant l'affichage d'un nombre important de séries temporelles simultanément. Toutes ces techniques basent leurs aggrégations sur le choix de l'utilisateur ou sur l'espace d'affichage disponible mais aucune ne cherche à optimiser la quantité d'information affichée en fonction de l'espace disponible.

## II. APPROCHE

Nos premiers travaux ont porté sur l'étude du passage à l'échelle des techniques existantes [3, 6, 2]. La limite découverte à travers ces travaux et dorénavant admise par la communauté est un graphe de 24 pixels de haut avec un nombre maximal de variation d'une couleur égal à 5.

Nos premiers prototypes sont allés jusqu'à des graphes de 1 pixel de haut sur une résolution verticale de 1800 pixels, affichant autant de graphes (i.e. variables temporelles dans le jeu de données). Sous ces conditions, nous avons pu observer les limites de perception, notre système visuel n'arrivant pas à percevoir certaines informations affichées, ainsi perdues. La Figure 2 montre un extrait d'une vue globale d'exécution. Chaque graphe, de quelques pixels de haut, correspond à un acteur au cours de l'exécution. A cette échelle, il devient difficile de percevoir les différences de contraste entre chaque ligne. En revanche, le caractère répétitif et périodique d'une exécution de décodage reste clair.

Notre approche face à ce problème est de définir un modèle basé sur notre connaissance de la perception humain qui optimise la quantité d'information affichée en fonction de l'espace écran disponible. Plus précisément, de nombreux travaux ont été ménés sur la sensibilité de l'oeil humain, notamment sa capacité à différencier plusieurs niveaux de contraste en fonction de la fréquence spatiale. La figure 3 [1] montre la limite de sensibilité de notre système visuel.

Avec la généralisation des écrans haute résolution, les pixels sont devenus invisibles à l'œil nu. Les visualisations doivent

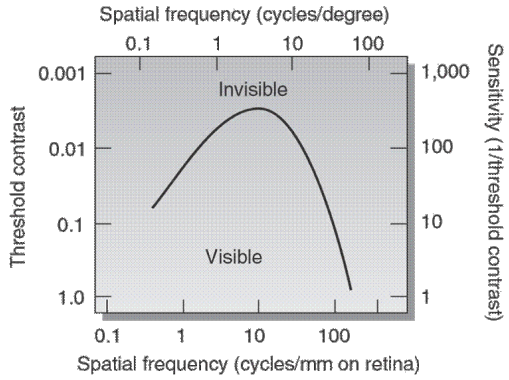


FIGURE 3. Sensibilité au contraste en fonction de la fréquence spatiale

dorénavant tenir compte de la limite physiologique de notre système visuel et ainsi maximiser la quantité d'information affichée en fonction de la quantité d'information perçue. De nouvelles contraintes pourront alors être générées par le modèle pouvant servir de point de départ pour les algorithmes d'aggrégation et à la découpe des données.

1. Source : http://www.cns.nyu.edu/~david/courses/perception/lecturenotes/channels/channels.html

## RÉFÉRENCES

[1] L. Berry and T. Munzner. Binx : Dynamic exploration of time series datasets accross aggregation levels. In *IEEE Symposium on Information Visualization, 2004*. IEEE Computer Society, 2004.

[2] M. Cho, B. Kim, H. J. Bae, and J. Seo. Stroscope : Multi-scale visualization of irregularly measured time-series data. In *IEEE Transactions on Visualization and Computer Graphics*, pages 808–821. IEEE Computer Society, 2014.

[3] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon : The effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1303–1312, New York, NY, USA, 2009. ACM.

[4] W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. In *IEEE Transactions on Visualization and Computer Graphics*, pages 927–934. IEEE Computer Society, 2010.

[5] R. Kincaid. Signallens : Focus+context applied to electronic time series. In *IEEE Transactions on Visualization and Computer Graphics*, pages 900–907. IEEE Computer Society, 2010.

[6] C. Perrin, F. Vernier, and J. D. Fekete. Interactive horizon graphs : Improving the compact visualization of multiple time series. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3217–3226, New York, NY, USA, 2013. ACM.

[7] J. Zhao, Chevalier F., and R. Balakrishnan. Kronominer : Using multi-foci navigation for the visual exploration of time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1737–1746, New York, NY, USA, 2011. ACM.