Bigimbaz:
un logiciel de recherche en indexation d'images

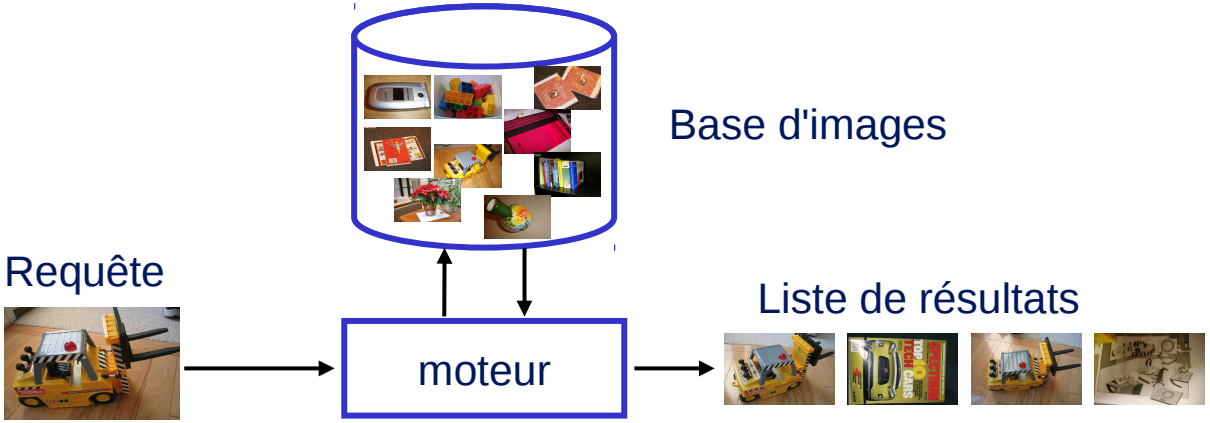
Matthijs Douze,

SED Grenoble

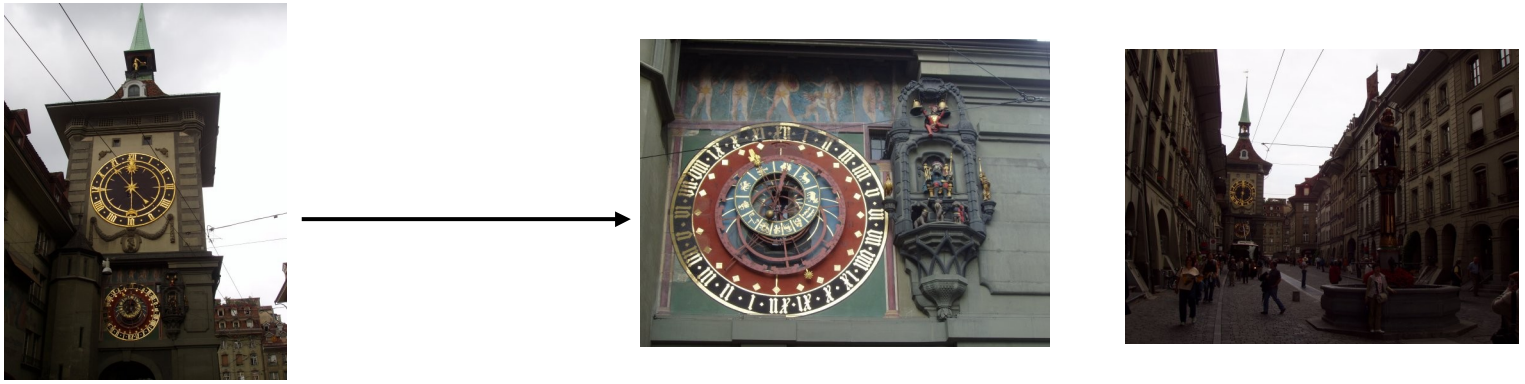
(travail effectué dans l'EPI LEAR)

Inria

Thème de recherche: l'indexation d'images



- Identification d'images similaires:
  - différents points de vue, images déformées...



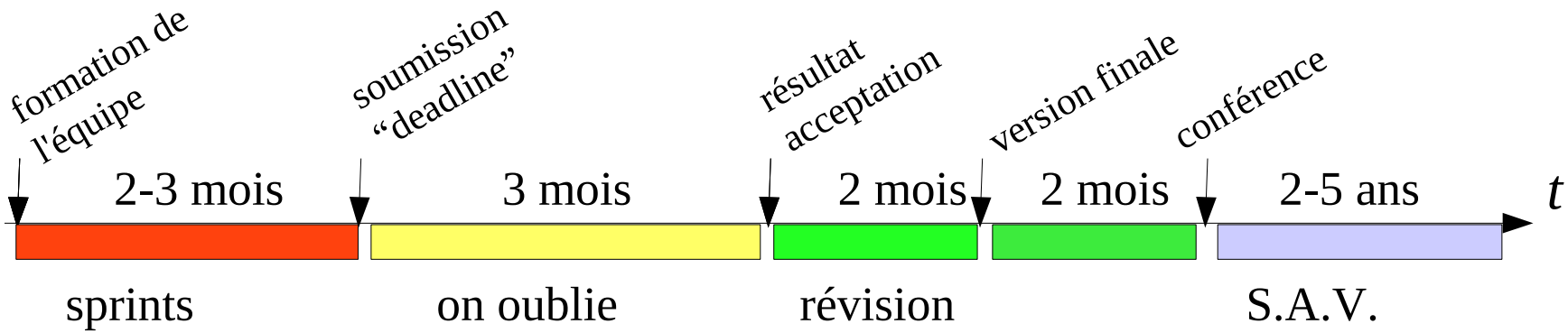
- Enjeux: pertinence du résultat, vitesse de recherche, taille de la base, etc.

Contexte de recherche

point de vue chercheur

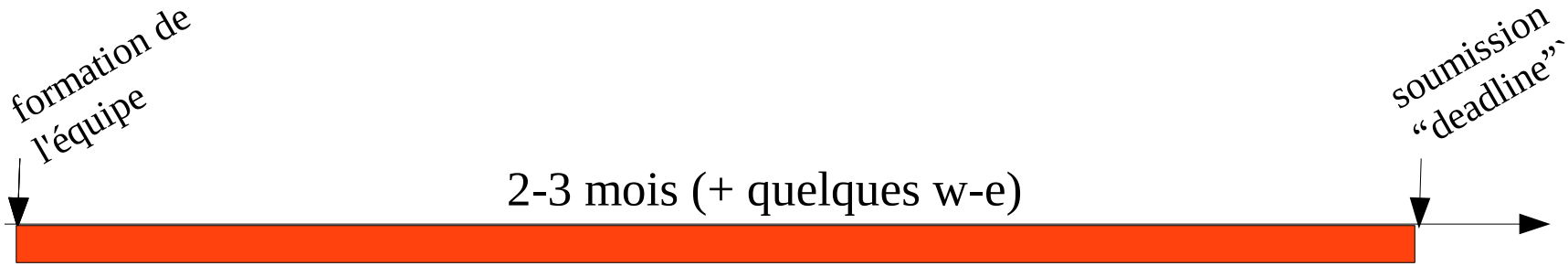
Contexte scientifique

- 2 conférences / an
  - grande communauté : 1500 papiers / conférence, 20-25% acceptés
  - timeline



  - journaux: même genre, en beaucoup plus long...
- Communauté très expérimentale
  - évaluation systématique, protocoles standard, concours
  - papiers théoriques sans expériences concluantes
  - prime à la simplicité : méthode tordue
  - reproductibilité : description incomplète

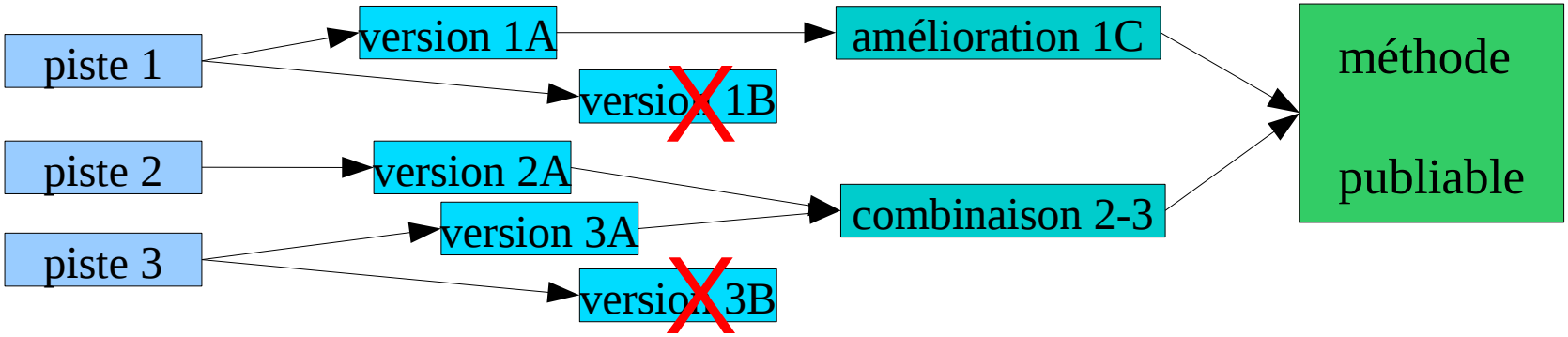
Le papier



- Au départ
  - une équipe : chercheur / doctorant / post-doc / ...
  - un domaine : indexation de vidéos (par ex.)
  - une baseline : résultat à améliorer
  - des idées : hypothèses, intuitions, pistes théoriques
- Sprints
- Écriture du papier (1-2 semaines)
- **L'ingénieur a sa place**
  - réflexes des ingénieurs / besoins des scientifiques
  - ingéniérie invisible (pas dans le papier, pas dans les réunions)

Phase de prototypage

- Exploration en largeur d'abord

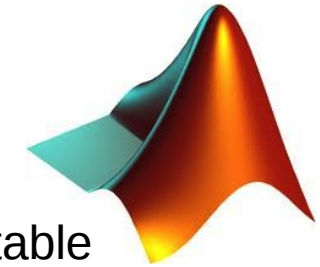


- Cycle implémentation – évaluation court
  - 1-3 jours
  - évaluation paramétrique
  - 60-95 % des pistes ne marchent pas : stop early
- Type de développement logiciel :
  - pas du code poubelle !

Important :
- vitesse de développement
- vitesse d'exécution
- ajout de fonctionnalité
- traçabilité

Inutile :
- portabilité
- documentation
- sécurité
- généricité

Approche classique : Matlab



- Idéal pour non-informaticiens
  - Langage ultra-simple, pas ambigü, stable
  - efficace quand on sait optimiser
  - riche librairie standard : visu, image, statistique...
  - bonne interface vers C (mex)
  - bonne alternative libre (octave)
- Standard :
  - le scientifique protoype en Matlab → publication
    - Distribution d'un script minimal
  - transfert industriel / démonstrateur...
    - un ingénieur arrive par derrière et le traduit en C++
- Limites de Matlab:
  - langage pauvre (structures de données, modularité)
  - problèmes de montée en charge : exploration paramétrique sur cluster...
  - il existe des rustines...
  - vendor lock-in
- Autre approche classique : le framework C++

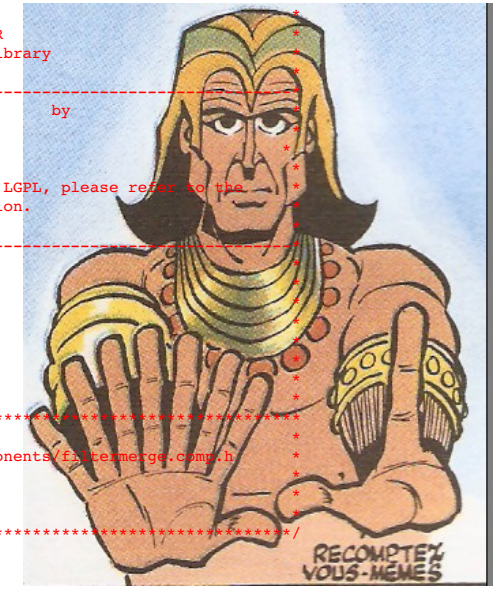
Bigimbaz

Présentation

- Historique :
  - serveur web de démo (2007)
  - Évolution → plateforme expérimentale
  - Administration : tandem chercheur (H. Jégou) + ingé (moi) -- gforge
  - + 7 contributeurs, ~25 utilisateurs
- Architecture :
  - Code de haut niveau, appel en ligne de commande : Python
  - Routines critiques : C (wrapping swig)
- Bénéfice :
  - Prototypage aussi rapide que Matlab
    + modulaire, démos, campagnes d'expériences
  - Moins lourd qu'un framework C++
- Priorité : **moins de code**
  - Python compact
  - Suppression agressive de code mort (contributeurs partis, algos peu performants, API complexes)

RGPP du code administratif

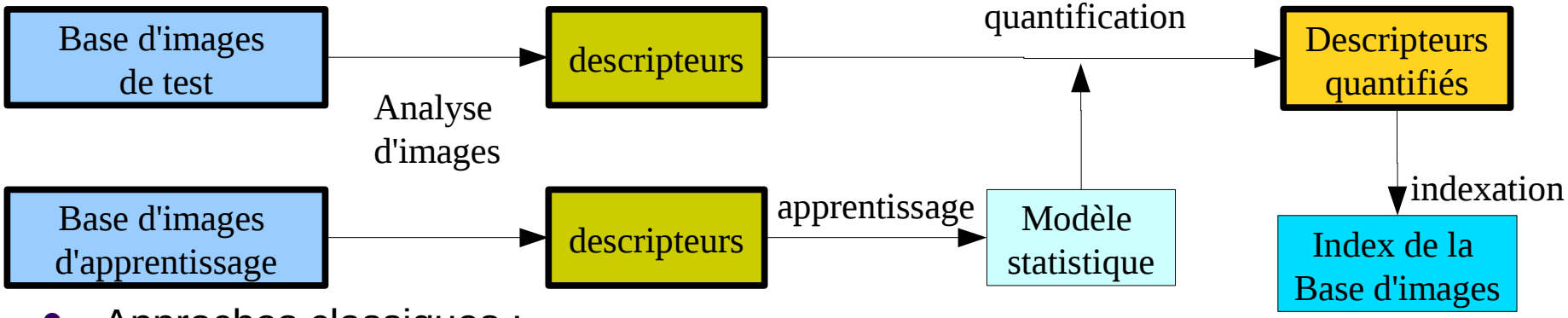
- Code administratif = ce qui n'existe pas en Matlab
  - Code qui ne fait rien
  - Valeur nulle pour la recherche
- Vision syntaxique (vert = code utile)
  - Cartouche d'en-tête (voir svn)
  - Includes, namespaces
    - Redondance nom de fichier / namespace / nom de la classe / commentaire
  - Commentaires copié-collés
  - Templates, typedefs, public/private, etc.
  - Code copié-collé
- Inutile donc nuisible
  - Perte de temps (écriture, lecture, compilation)
  - Perte de focus



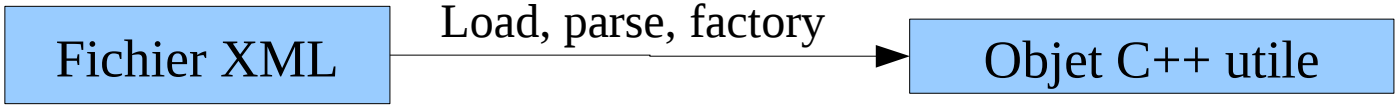

```
/******* COPYRIGHT ************************************************
 *                                                                *
 *                          FlowVR                                *
 *                    Application Library                         *
 *                                                                *
 *----------------------------------------------------------------*
 * COPYRIGHT (C) 2003-2011                  by                    *
 * INRIA                                                          *
 * ALL RIGHTS RESERVED.                                           *
 *                                                                *
 * This source is covered by the GNU LGPL, please refer to the    *
 * COPYING file for further information.                          *
 *                                                                *
 *----------------------------------------------------------------*
 *                                                                *
 *  Original Contributors:                                        *
 *    Jeremie Allard,                                             *
 *    Thomas Arcila,                                              *
 *    Jean-Denis Lesage.                                          *
 *    Clement Menier,                                             *
 *    Bruno Raffin                                                *
 *                                                                *
 ******************************************************************
 *                                                                *
 *  File : ./include/flowvr/app/components/filtermerge.comp.h     *
 *                                                                *
 *  Contact : Jean-Denis Lesage                                   *
 *                                                                *
 ******************************************************************/

#ifndef __FLOWVR_APP_FILTERMERGE__
#define __FLOWVR_APP_FILTERMERGE__

#include "flowvr/app/core/component.h"

/**
 * \file filtermerge.comp.h
 * \brief Merge filter
 */

namespace flowvr { namespace app {

    /**
     * \class FilterMerge
     * \brief Merge filter: composite shell  encapsulating a FilterMergePrimitive
     */
    class FilterMerge : public Composite
    {
        public :
            /**
             * \brief constructor with an id
             */
            FilterMerge(const std::string& id_) : Composite(id_, "Shell")
            {
                setInfo("A shell around a filtermerge component with one input port  mas
filtermerge");
                addPort("in", INPUT,FULL);
                addPort("out", OUTPUT,FULL);
            }

            /**
             * \brief add a FilterMergePrimitive with nb inputs
             *  nb defined according to primitive objects connected to the in port of Fi
             */
            virtual void execute();

            virtual Component* create() const   {       return new FilterMerge(getId());

    };

};};
#endif //__FILTERMERGE
111
```

Moins de code administratif : exemple des configurations

- Gestion des configurations
  - passé en paramètre à un programme
  - des paramètres, un graphe, des listes de fichiers, etc.
- Pipeline dans Bigimbaz :



- Approches classiques :
  - Matlab : tout en dur...
  - Framework C++ :



  - Bcp de code, duplication de déclarations (DTD, code de parsing, classe)
  - On n'arrive plus à bouger le code : Arthrose du pipeline !

Une configuration dans Bigimbaz

- Solution : utiliser Python comme fichier de configuration
  - une configuration = 1 instance de classe
  - chargement dynamique de la classe
- Config Python :
  - Nom
  - Paramètres
  - Fichier unique
  - Mapping numéro vers nom de fichier
- instance passée en paramètre à
  - `compute_descriptors, train_quantizer, ...`
- Guérit l'arthrose du pipeline!

```
class NisterConfig(Config):

    def __init__(self):
        Config.__init__(self)
        self.basedir = '/scratch2/bigimbaz/datas
        self.nimg = 10200
        self.det_prog = 'obsidian'
        self.nb_centroids = 20000
        self.invfilename= self.basedir + 'ivfgeo
        self.binsign_computer =
"/scratch2/bigimbaz/dataset/flickr/clust/params_

    def img_filename(self,n):
        return self.basedir + 'jpg/ukbench%05d.j

    def desc_filename(self,n):
        return self.basedir + 'siftgeo/hesaff_no

    def vw_filename(self,n):
        return self.basedir + 'vwsgeo/hesaff_nor

class TrainConfig(Config):
  .....
```

Chercheur = late adopter technologique

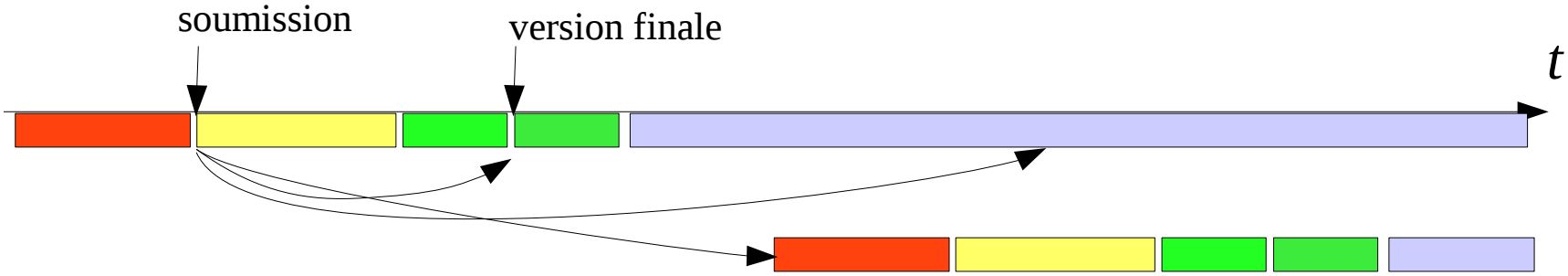
- Choix technologiques conservatifs
  - Valeur en recherche de l'apprentissage de techno = 0
  - Vieux = fiable, maintenu, pérenne, connu
  - Primitif = moins de niveaux d'abstraction
    - de compilation = contrôle précis, facile à débugger
    - de librairies = pas de surcouche, de dépendances

Ceci...	...plutôt que
Python 2	Python 3
swig	cython
Makefile	Cmake
C	C++, Java
svn	git
Lapack interface Fortran	clapack
BaseHTTPServer	Apache + mod_python

- Ajouté à Bigimbaz car bénéfice significatif :
  - SSE
  - numpy (plutôt que opérateurs matrices in-house)
  - openmp (plutôt que pthread)

Traçabilité

- Expériences reproductibles

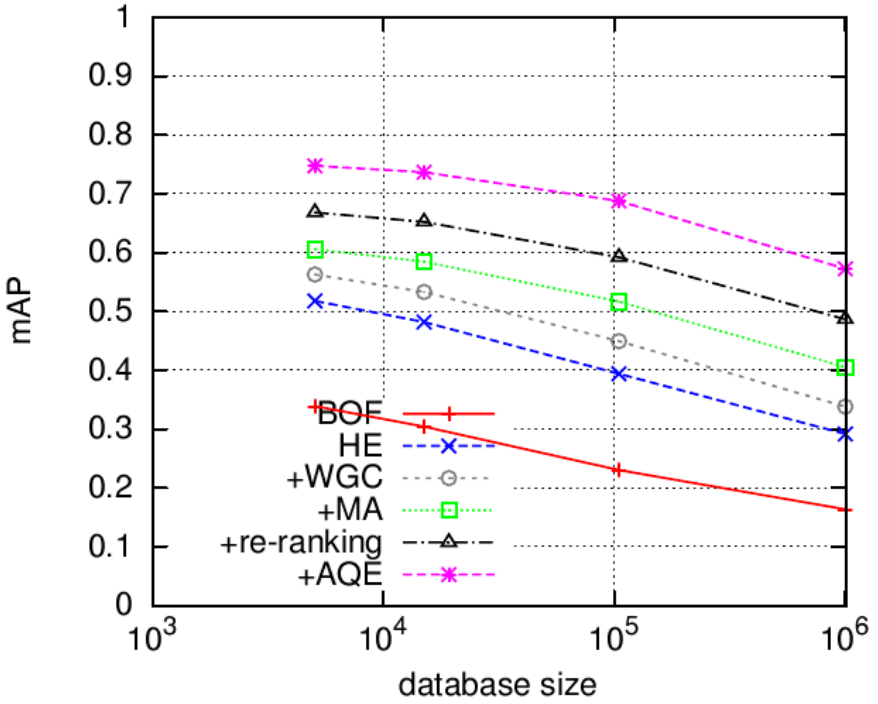


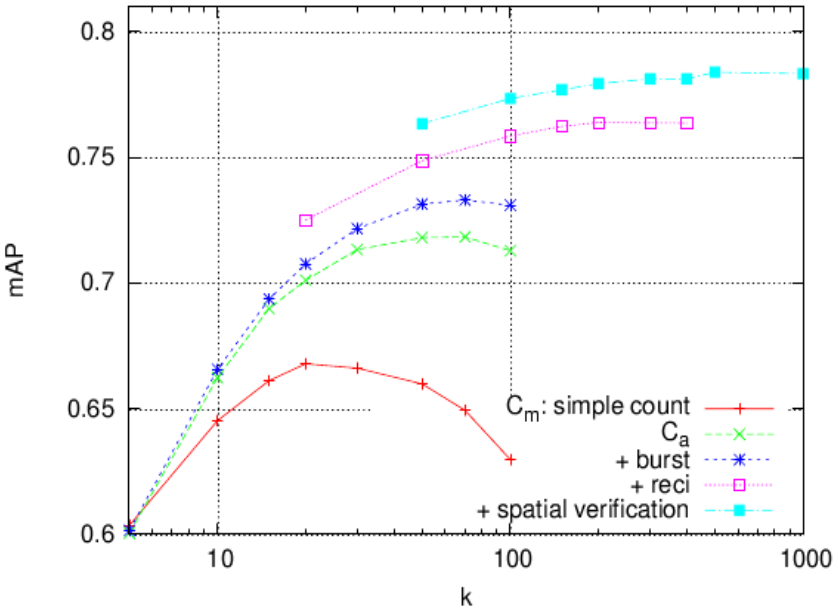
  - version finale / S.A.V. / baseline pour le travail suivant / publication du code
  - Pérennité du travail / non régression

- Organisation :
  - gestion de versions sur le code
  - environnement matériel / logiciel stable – ne pas effacer de données !
  - ligne de commande dans le source LaTeX
  - initialiser les générateurs aléatoires (multithread...)

Résultats

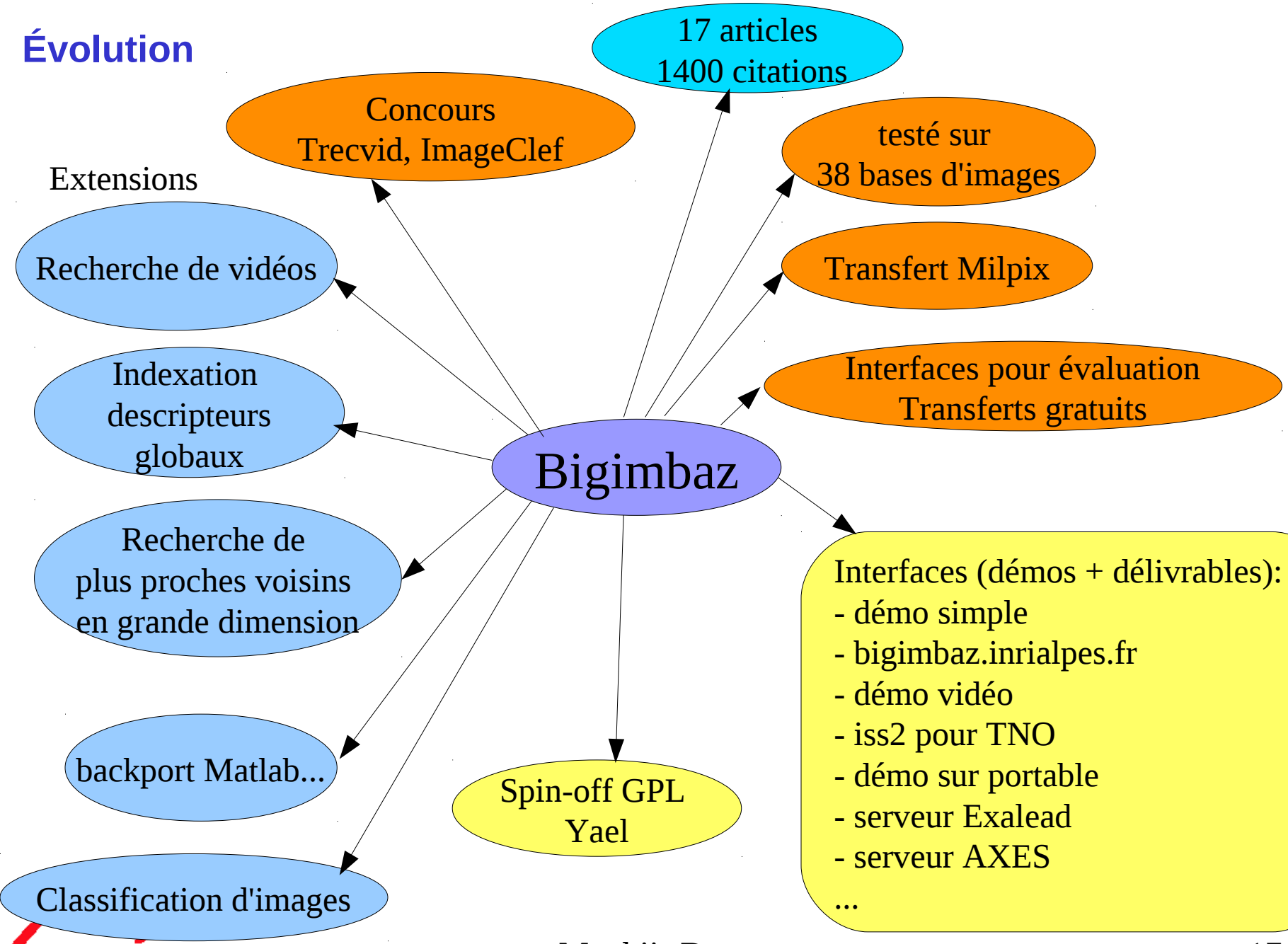
Papiers (17 articles, 1400 citations)





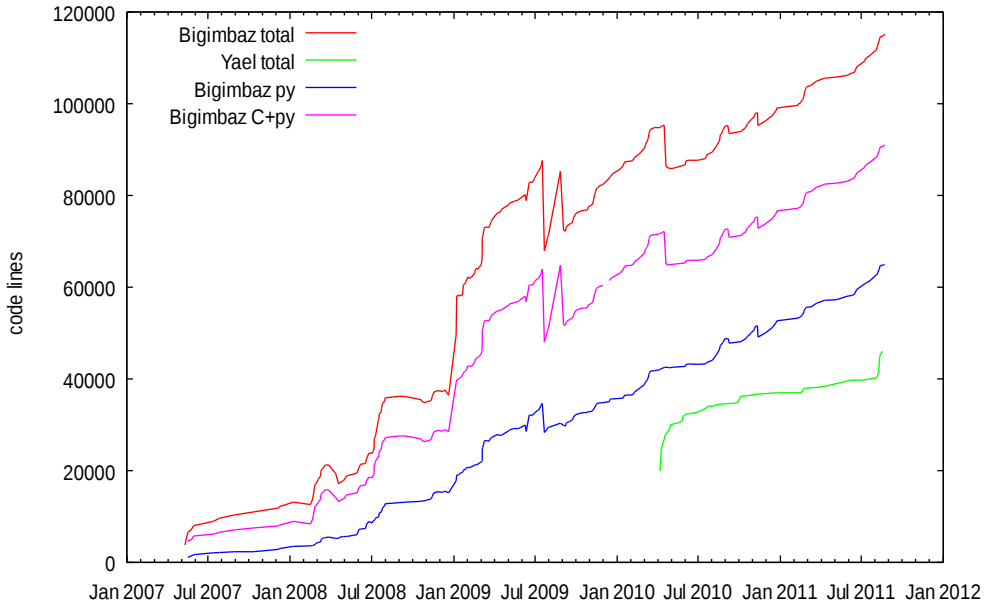
method	k	mAP	memory usage	memory scanned	query time
BOF	20k	0.227	7,322	860	22163
BOF	200k	0.315	8,885	148	2827
binary BOF	20k	0.307	5,858	688	14073
binary BOF	200k	0.381	7,108	117	2562
miniBOF, m=1	1k	0.066	20	0.19	71
miniBOF, m=8	1k	0.196	160	1.54	132
miniBOF, m=32	1k	0.244	640	6.14	352

Évolution


17 articles
1400 citations
Concours
Trecvid, ImageClef
testé sur
38 bases d'images
Extensions
Recherche de vidéos
Transfert Milpix
Indexation
descripteurs
globaux
Interfaces pour évaluation
Transferts gratuits
Bigimbaz
Recherche de
plus proches voisins
en grande dimension
Interfaces (démos + délivrables):
- démo simple
- bigimbaz.inrialpes.fr
- démo vidéo
- iss2 pour TNO
- démo sur portable
- serveur Exalead
- serveur AXES
...
backport Matlab...
Spin-off GPL
Yael
Classification d'images

Conclusion

- Le chercheur aime
  - le code simple
  - la traçabilité
  - les transferts qui coûtent 0 HM ingé
- Il n'aime pas
  - >1000 lignes de code
  - les nouvelles technologies
- Solution techno: Python + C via SWIG
- Retour sur expérience :
  - a supporté des dizaines d'extensions
  - est-ce une usine à gaz ?



C..... 26 k
Python..... 65 k
Yael..... 46 k

Total..... 115 k