



HAL
open science

Toward collaborative and contextualized online instrumentation in pervasive and ubiquitous environments

Benjamin Jailly

► **To cite this version:**

Benjamin Jailly. Toward collaborative and contextualized online instrumentation in pervasive and ubiquitous environments. Other [cs.OH]. Université Jean Monnet - Saint-Etienne, 2013. English. NNT : 2013STET4003 . tel-00994274

HAL Id: tel-00994274

<https://theses.hal.science/tel-00994274>

Submitted on 21 May 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.



THÈSE

TOWARD COLLABORATIVE AND CONTEXTUALIZED ONLINE INSTRUMENTATION IN PERVASIVE AND UBI- QUITOUS ENVIRONNEMENTS

Présentée devant :
Télécom Saint-Étienne

Pour obtenir :
Le grade de docteur

Spécialité :
Informatique

Formation doctorale :
Informatique

École doctorale :
Système Ingénierie et Santé

Par :
Benjamin JAILLY

SOUTENUE PUBLIQUEMENT LE 22 MARS 2013 DEVANT LE JURY COMPOSÉ DE :

Jacques FAYOLLE, Maître de Conférences HDR, Télécom Saint-Etienne Directeur de thèse
Christophe GRAVIER, Maître de Conférences, Télécom Saint-Etienne Co-directeur de thèse
Marius PREDÀ, Maître de Conférences, Télécom SudParis Co-directeur de thèse
Alain MILLE, Professeur des Universités, Université Claude Bernard Lyon 1 Examineur
Jean-Claude DUFOURD, Professeur des Universités, Télécom ParisTech Rapporteur
Serge GARLATTI, Professeur des Universités, Télécom Bretagne Rapporteur
Giovanni CORDARA, Docteur, Huawei Munich Examineur

LABORATOIRE TÉLÉCOM CLAUDE CHAPPE (LT2C), TÉLÉCOM SAINT-ÉTIENNE

THÈSE

TOWARD COLLABORATIVE AND
CONTEXTUALIZED ONLINE
INSTRUMENTATION IN PERVASIVE AND
UBIQUITOUS ENVIRONMENTS

Benjamin JAILLY

À la mémoire de Patrice Jailly

Foreword

Thank-you note

Ce travail a été réalisé au sein du Laboratoire Télécom Claude Chappe (LT2C) de l'université Jean MONNET de Saint-Étienne en partenariat avec le département Advanced Research and TEchniques for Multidimensional Imaging Systems (ARTEMIS) de l'Institut Télécom. Il a été financé grâce au projet "Futur&Rupture" de l'Institut Télécom.

Écrire les remerciements d'un travail long de 3 années est une tâche difficile. Il n'est jamais simple d'exprimer sa gratitude envers toutes les personnes avec qui j'ai pu interagir de près ou de loin pendant ces 3 années.

Je tiens tout d'abord à remercier Jacques Fayolle, directeur de thèse, Christophe Gravier et Marius Preda, tous deux co-encadrant, pour l'encadrement, la reconnaissance et la confiance qu'ils m'ont accordé pendant ces 3 années.

Je tiens à remercier Jean-Jacques Rousseau, directeur du laboratoire LT2C, pour m'avoir accueilli au sein du laboratoire, ainsi que Frédérique Laforest, responsable de l'axe Système Adaptatif pour les Télécoms et Interfaces Numériques (SATIN) du laboratoire, pour ses précieux conseils de rédactions.

Je ne peux oublier l'aide précieuse, les discussions constructives et les moments de détente partagés avec mes collègues du laboratoire, Yves-Gaël, Antoine, Julien, Jérémy, Pierre-Oliver, et les gérants de Delicias' App, Mehdi et Kevin. Merci à tous les membres du laboratoire que j'ai pu côtoyer à la machine à café ou dans les couloirs, Alain, Damien, Didier, Jean-Pierre, Bruno... et qui ont contribué au cadre agréable de travail au sein du laboratoire et de Télécom Saint-Étienne. Je tiens également à remercier les collègues du département ARTEMIS, Veronica, Nicolas, Margaret, Christina, Thomas, Ivica, Christian, Traian et bien sûr Évelyne.

Je n'oublie pas les aides reçues du personnel administratif de Télécom Saint-Étienne, Gaétane, Julie, Élodie, Viviane, Chantal, Sophie, Valérie, Marion, Gilbert et Christian, Frédérique, Florence, Claudine et Stéphanie.

Je tiens également à remercier mes amis, Philip, Mickaël, Benoit, Virginie, Nicolas, Julien, Paul, Juliette, Aurélie, Mélanie et Jérémy, ma mère et Alain, mes deux frères Sébastien et Michaël et leurs femmes Aurélie et Sophie ainsi que Stéphanie, pour leurs

soutiens tout au long de ces 3 années.

Résumé long

0.1 Introduction

TOUT au long des dernières décennies, les architectures logicielles ont évolué de façon significative. Avec l'explosion d'Internet, les systèmes informatiques sont capables de traiter des données de plus en plus complexes. Le développement des terminaux utilisateurs a également favorisé les transformations des architectures logicielles.

0.1.1 Il était une fois...

L'émergence des **Applications Internet Riches** (RIA, de l'anglais Rich Internet Application) est le résultat de l'évolution des architectures logicielles, du "mainframe" pendant les années 60 à 80, à l'architecture "deux-tiers" dans les années 80 puis à l'architecture "n-tiers" dans les années 90. Elles sont principalement exécutées dans un navigateur Web à l'aide de plug-in et de cadres externes. Les plus connus sont **Asynchronous Javascript and XML** (AJAX), **Adobe Flash** ou encore **Microsoft Silverlight**. Leurs principaux avantages sont de proposer :

- une ergonomie avancée,
- la possibilité de modifier l'interface utilisateur en fonction du business,
- une sécurité accrue grâce aux techniques de bac à sable (sandbox),
- la possibilité d'embarquer des animations et du contenu multimédia.

Les capacités de communication et l'évolution des architectures logicielles ont également rendu possible le contrôle distant d'appareils divers avec leurs représentation dans le monde applicatif. Les relations entre les humains et les machines ont évolué, que cela soit en situation de contrôle d'un appareil ou en situation de simulation. Ce contact n'est pas physique mais est réalisé à l'aide de l'informatique. On parle alors de "systèmes assistés par ordinateur" car l'ordinateur n'est pas seulement responsable de représenter l'Interface Homme-Machine (IHM) de l'instrument contrôlé ou simulé mais également d'assister l'opérateur humain dans sa tâche.

Si l'on met de côté l'aspect intrusif de l'informatique, il est important de remarquer que l'innovation n'est pas seulement technologique. L'informatique peut mettre en relation plusieurs personnes avec plusieurs machines, mettant en place une collaboration pour accomplir des tâches communes. Les télé-opérations, ou la capacité de contrôler un appareil à travers Internet, permettent de mettre en place des manipulations qui ne pourraient pas avoir lieu autrement.

0.1.2 L'ingénierie en ligne et les laboratoires distants

Cette étude est centrée sur l'ingénierie en ligne (de l'anglais online engineering) et les laboratoires distants.

Le facteur financier peut être déterminant dans le choix de mettre en place une solution de contrôle distant d'un appareil. Les coûts des appareils à la pointe de la technologie sont souvent trop élevés en regard de leur fréquence d'utilisation. La solution habituelle consiste à déplacer les opérateurs sur le site de l'instrument, induisant des contraintes administratives et des coûts indirects (logement, temps de déplacement, etc.).

Cependant, le coût **n'est pas le seul facteur** qui peut entraîner un contrôle distant d'appareils. D'autres facteurs comme la sécurité, l'accès à l'instrument, la disponibilité des personnes, l'observation en sont des exemples.

L'instrumentation en ligne est aujourd'hui principalement employé par trois type d'organisation :

- les instituts de formation pour l'éducation en ligne,
- l'industrie pour la maintenance à distance et l'entreprise étendue,
- les laboratoires de recherche pour le transfert de compétences.

Pour illustrer nos propos et nos travaux, nous proposons deux scénarios. Le premier concerne Steve, Heather et Jimmy. Steve est enseignant en Norvège et Heather et Jimmy sont deux de ses étudiants actuellement à l'étranger, le premier à Paris, la seconde à Tokyo. Pour pouvoir valider leurs diplômes, Heather et Jimmy doivent valider des cours dans leurs université de départ. Ce cours inclut des travaux pratiques sur un analyseur de spectre et sur un analyseur vectoriel de réseaux.

Le second scénario concerne Alice et Bob. Alice est technicienne dans une manufacture. Suite à une panne sur son appareil, elle contacte Bob qui est opérateur dans une entreprise de maintenance. Pour réduire le coût de la maintenance, Bob effectue tout d'abord une opération à distance sur l'appareil défectueux. Étant donné que Bob effectue des opérations de maintenance sur un large panel d'appareil, il ne connaît pas par coeur toutes les procédures à effectuer. Il utilise donc le système d'assistance de son entreprise qui inclut ces procédures.

Ces deux scénarios induisent différentes problématiques. En ce qui concerne l'éducation en ligne, ces problèmes sont les suivants :

- une grande fidélité de représentation d'un instrument dans son environnement.
- une interopérabilité permettant de contrôler l'instrument distant de façon pervasive¹ et depuis n'importe quel type de terminal (smartphone, tablette, PC, etc.),
- un contrôle de l'instrument collaboratif et mettant en place une sensation de groupe,
- une solution la plus réutilisable et simple possible.

1. n'importe où, n'importe quand

Le scénario sur la maintenance à distance permet de soulever les problématiques suivantes :

- le système doit proposer un formalisme pour exprimer les procédures,
- la logique métier qui permet de déterminer les actions suivantes doit être découplée le plus possible d'autres composants logiciels pour être réutilisable,
- le système doit proposer une méthode pour analyser les actions effectuées et suggérer des améliorations à l'utilisateur.

Nous les compilons dans la liste suivante et soulignons le fait qu'ils impacts de façon significative l'interface homme machine du système :

- un contrôle d'instruments distants qui présente une grande immersion et une grande compréhension de la scène appréhendée,
- un contrôle collaboratif de l'appareil distant,
- une adaptation de l'interface pour aider l'opérateur,
- un système scalable et capable de s'adapter à un large panel d'instruments.

0.1.3 Organisation du manuscrit

L'ingénierie en ligne permet de contrôler des instruments à distance. Elle est motivée par la volonté de réduire des coûts tout autant que par la volonté de répondre à des problématiques techniques telles que la dangerosité, l'observabilité, la disponibilité, etc. Principalement trois types d'organisations sont concernés par l'ingénierie en ligne : l'industrie pour la maintenance à distance et l'entreprise étendue, les instituts de formation pour l'éducation en ligne, et les laboratoires de recherche pour le partage et le transfert de compétences.

Nous avons présenté deux cas d'usage, l'un concernant l'éducation en ligne et l'autre la maintenance à distance. Ces scénarios nous ont permis de soulever des problématiques.

L'organisation de ce document est la suivante. Le chapitre 2 présente l'état de l'art dans les systèmes d'ingénierie en ligne. Le chapitre 3 présente le cœur de notre proposition et son architecture générale. Le chapitre 4 détaille certains points de l'implémentation de notre système. Le chapitre 5 conclut et propose des pistes d'ouvertures et de continuité de nos travaux.

0.2 État de l'Art

L'INGÉNIERIE en ligne et les laboratoires distants permettent de mettre en place des manipulations sur des instruments réels via Internet. Les *feedbacks*² de l'instrument distant doivent donc être représentés pour appréhender l'état de l'opération menée. Côté client, cela peut être permis grâce à la représentation des feedbacks de l'instrument en tant que composants graphiques dans l'IHM de pilotage ou alors en tant que composant vidéo de l'instrument filmé. Dans tous les cas, l'IHM présentée doit être fidèlement représentative de l'instrument piloté. La réalité mixte peut être un moyen d'améliorer l'immersion de l'utilisateur et sa compréhension de la scène distante.

À l'ère de l'informatique pervasive et ubiquitaire, les laboratoires distants doivent être accessibles n'importe quand, n'importe où et depuis n'importe quel type de terminal. Cela implique un fort degré d'interopérabilité dans les solutions à mettre en place. La réutilisabilité ainsi que le couplage lâche des composants logiciels sont des points à prendre en compte. La collaboration, i.e. la possibilité d'accomplir une tâche en groupe, ainsi que la conscience du groupe sont également des points importants à prendre en compte.

Dans ce chapitre, nous nous intéressons tout d'abord aux technologies et paradigmes qui permettent de répondre aux problématiques rencontrées au premier chapitre. Dans un second temps, nous présentons les projets de recherche principaux dans le domaine de l'ingénierie en ligne puis nous nous arrêterons sur une synthèse de ces projets.

0.2.1 Technologies

0.2.1.1 Réalité Mixte

La réalité augmentée est la combinaison d'objets virtuels dans un mode réel. La virtualité augmentée est la combinaison d'objet réel dans un monde virtuel. Puisque la frontière entre ces deux concepts n'est pas direct, nous préférons employer le terme de réalité mixte qui embrasse ces deux notions.

Pour réaliser un système de réalité mixte, trois règles fondamentales doivent être respectées :

- combiner les mondes réel et virtuel,
- prendre en compte de la cohérence entre ces deux mondes,
- prendre en compte l'aspect temps-réel et l'interactivité des systèmes de réalité mixte.

Pour pouvoir afficher cette combinaison d'objets réels et virtuels dans un monde commun, un environnement propre doit être choisi. Le plus souvent, c'est un environ-

2. réactions et retours

nement géométrique à trois dimensions. Dans cet environnement, les objets doivent être correctement alignés. Pour réaliser cette tâche, différentes méthodes de suivi d'objets sont mises en place. L'objet suivi peut être l'utilisateur lui-même ou des objets de l'environnement. La plupart des méthodes de suivi sont issues du suivi avec ou sans marqueur.

Le suivi d'objet avec marqueur consiste à placer un objet de référence facilement identifiable dans l'environnement. Ce marqueur peut être visible (comme celui proposé par la librairie ARToolkit), ou invisible (infra-rouge par exemple).

Les techniques de suivi d'objets sans marqueur peuvent prendre différentes formes. Ils peuvent se baser sur des méthodes de système de position (Global Positioning System et boussole), mécanique, ou encore sur des techniques de traitement d'image.

Le suivi basé sur le traitement des images se fonde sur la détection de points d'intérêt dans les images. On peut citer les méthodes FAST [Rosten05], le détecteur de coins de Harris[Harris88] ou encore la différence de gaussienne (DoG)[Lowe04]. D'autres techniques de traitement d'image ajoute à ces points d'intérêts de descripteurs qui permettent de caractériser ces points et faciliter leur identification. Tuytelaars et al. définissent les caractéristiques des descripteurs dans leurs travaux [Tuytelaars08]. Parmi les plus connus, on peut citer Scale-Invariant Feature Transform (SIFT)[Lowe04], Speeded-Up Robust Feature (SURF)[Tuytelaars08], BRIEF[Calonder11] et Oriented FAST and Rotated BRIEF (ORB)[Rublee11].

À partir des différentes méthodes de suivi, il est possible de créer un large choix d'applications dans des domaines divers tel que la culture[Marshall11], les jeux vidéo [Cheok04], l'armement[Julier00] ou la santé [Suzuki10].

0.2.1.2 Collaboration

Les travaux sont souvent naturellement collaboratifs. Cela signifie que plusieurs acteurs interagissent entre eux dans le but d'atteindre un but commun. En revanche, il est important de souligner que *collaboration* et *coopération* sont différents. La coopération implique une division des tâches à effectuer, contrairement à la collaboration où les tâches ne sont pas divisées.

La communication prend une part importante dans la collaboration. Différents outils peuvent être utilisés (e-mail, conférence audio-visuel, etc.), chacun apportant des fonctionnalités différentes (messages synchrones/asynchrones, etc.).

0.2.1.3 Adaptation au contexte et Connaissance Sémantique

En informatique, le contexte représente "*toute information qui peut être utilisée pour représenter une situation ou une entité. Une entité est une personne, un endroit, ou un objet*".

qui est considérée comme pertinente dans l'interaction entre un utilisateur et une application, incluant l'utilisateur et l'application eux-même." [Dey00].

Nous n'avons pas l'intention dans ces travaux de redéfinir les systèmes adaptatifs au contexte, mais plutôt de définir un élément clé du contexte de l'utilisateur : la procédure d'utilisation de l'IHM.

Dans le domaine de la maintenance assistée par ordinateur, l'ingénierie en ligne aide à effectuer des opérations en lignes. Les opérateurs de maintenance sont responsables d'un large panel d'instruments avec un large panel de procédures propres. Dans ce cadre, un système de contrôle distant d'instruments peut proposer un formalisme pour sérialiser ces procédures et adapter l'interface homme machine du système dans le but d'aider l'opérateur dans sa procédure de maintenance dans le but de limiter le nombre d'erreurs humaines.

Le Web Sémantique et ses technologies associées (RDF/RDFS) couplés avec un moteur de règle sémantique permettent le raisonnement nécessaire aux systèmes adaptatifs au contexte.

0.2.1.4 Description de Scène Multimédia

En informatique, on désigne par *média* un support d'information. Le terme *multimédia* désigne généralement des applications capables de générer ou contrôler plusieurs média tels que le son, la vidéo, les images, etc. Nous définissons un multimédia comme "*plusieurs média ayant une relation physique, logique, temporel ou sémantique présenté suivant les choix de l'auteur*". Pour pouvoir maîtriser de telles relations, il est possible d'utiliser les descripteurs de scène multimédia.

Ces descripteurs de scènes permettent de mettre en oeuvre les instructions de gestion des différents média. Les propriétés des descripteurs de scène sont les suivantes [Concolato07] :

- l'emploi de média audio-visuel (naturel ou synthétique),
- l'organisation spatiale des objets (relative ou absolue) dans un environnement à deux ou trois dimensions,
- l'organisation temporelle des objets,
- les capacités d'interactivités.

On représente le plus souvent une description de scène multimédia par un arbre de scène (en anglais *scene-graph*). Les plus connus sont Virtual Reality Modeling Language (VRML), Synchronized Multimedia Integration Language (SMIL), Scalable Vector Graphics (SVG), COLLABorative Design Activity (COLLADA), Adobe Flash, JavaFX, Microsoft Silverlight, HyperText Markup Language (HTML), MPEG-4 Binary Format for Scene (MPEG-4 BIFS et MPEG-4 Lightweight Application Scene Representation (MPEG-4 LAsER).

Les descripteurs de scène multimédia ne sont pas tous capables de présenter les

Descripteurs de scène	Composants Graphiques	Audio et Vidéo	Streaming	Compression	Interactivité
VRML	3D	oui	non	non	oui
SMIL	2D	oui	non	non	oui
SVG	2D	non	non	version SVGZ	oui
COLLADA	3D	non	non	non	non
Flash	2D/3D	oui	oui	oui	non
JavaFX	2D/3D	MP3 & VP6	non	compilation bytecode	oui
Silverlight	2D/3D	plusieurs codecs	non	compilation	oui
HTML	2D and 3D through WebGL	oui, mais partiellement standardisé	oui	non	oui
BIFS et XMT	2D/3D	oui	oui	oui	oui
LASeR	2D	oui	oui	oui	oui

TABLE 1 – Descripteurs de scènes multimédia

mêmes types de contenu. Le tableau 1 résume les capacités des différents descripteurs de scène. Seuls les descripteurs de scène proposées par le comité MPEG, i.e. MPEG-4 LASeR et MPEG-4 BIFS, sont capables de fournir des flux de la totalité de leurs contenu. LASeR, contrairement à BIFS, n'est pas capable de gérer les contenus en trois dimensions, ce qui fait de MPEG-4 BIFS le descripteur de scène le plus complet.

0.2.2 Ingénierie en ligne

Nous avons vu que la réalité mixte est fortement présente dans différents domaines. C'est aussi le cas dans l'industrie. Nous pouvons par exemple citer le projet européen **FP5 STARMATE**, le projet français **Télé Assistance Collaborative** ou encore le projet utilisé par l'US Navy **Augmented Reality for Maintenance and Repair (ARMAR)**.

Dans l'éducation en ligne, il existe de nombreux cadres pour contrôler à distance différents types d'appareils. Ces cadres apportent différentes briques. Par exemple, **iLab** propose une architecture basée SOAP pour gérer l'authentification, l'orchestration ou l'autorisation de l'accès à la ressource [Harward04]. **VISIR** quant à lui propose une interface basée sur Adobe Flash pour représenter l'IHM de l'instrument. Il propose éga-

lement son propre système de gestion d'éducation (Learning Management System en anglais).

0.2.3 Synthèse

Les tableaux 3 et 2 résument l'étude des projets existants. L'étude des projets existant pour l'industrie montre cependant que les projets de réalité mixte ne sont pas ou peu employées dans un usage distant des environnements contrôlés.

Bien que beaucoup de systèmes pour l'éducation en ligne permettent de contrôler des appareils via Internet, peu de ceux-ci utilisent la réalité mixte pour augmenter l'immersion de l'apprenant et encore moins de descripteurs de scènes sont proposés pour gagner en interopérabilité.

Peu de systèmes proposent un contrôle collaboratif de l'instrument distant. Le même constat est valable pour les systèmes capables d'adapter leurs interfaces en fonction d'éléments du contexte.

Au vu de l'analyse réalisée, il nous paraît important de pouvoir proposer une solution de contrôle d'instruments distants répondant au mieux aux questions suivantes :

- Comment proposer un système utilisable à la fois dans l'industrie et dans l'éducation ?
- Comment proposer un système de réalité mixte interopérable ?
- Comment proposer une RIA qui limite la bande passante ?
- Comment gérer la collaboration au sein d'une application de réalité mixte ?
- Comment proposer une solution la plus réutilisable possible pour diminuer le coût de développement des laboratoires distants ?
- Comment proposer un système qui prend en compte les procédures d'exécution des opérateurs et qui adapte le contenu de l'IHM en fonction ?

Projets	Local ou distant	Réalité Mixte	Description de scène multimédia	Collaboration	Adaptation au contexte
iLab	Distant	non	non	non	non
eLab	Distant	non (video feedback)	HTML, SVG, Flash	oui	non
VISIR	Distant	non	HTML, Flash	non	non
LabShare	non (video feedback)	non	non	non	non
Telelabs	Distant	oui	non (ActiveX)	non	non
PEARL	Distant	non (feedback vidéo)	non (Java Applets)	oui	non
WebLab	Distant	oui	Flash (environnent 3D)	oui	non
RExNet	Distant	non	VRML	travaux en cours	non
CORELA	Distant	oui	non (OpenSim)	oui (à travers Moodle)	non
ISILab	Distant	non	HTML	non	
Remote Lab at EPFL	Distant	non (video feedback)	travaux en cours pour HTML 5 et SVG	oui	oui
NetLab	Distant	non	non	oui	non
DIESEL	Distant	non (feedback vidéo)	Flash	oui	non
LiLa	Distant	non	non (Wonderland)	oui	non
[Andujar11]	Distant	oui	non (Java Applet)	non	non
[Odeh09]	Distant	oui	non	non	non

TABLE 2 – Analyse des solutions pour l'éducation en ligne

Projets	Local ou distant	Réalité Augmentée	Description de scène multimédia	Collaboration	Adaptation au contexte
FP5 STARMATE	Local	Appareil porté sur la tête	non	non	non
ARMAR	Local	Appareil porté sur la tête	non	oui	non
Chocolate Factory	Distant	oui	non (Unity)	oui	non
T.A.C	Local	Appareil porté sur la tête	non	oui	non
ARVIKA	Local	Appareil porté sur la tête	non	non	oui
ARITI	Distant	rendu 3D virtualisé	non (3DVIA & Java)	oui	non
AMRA	Local	oui	X3D/VRML	oui	oui
VRMedia	Local	oui	non	oui	non
Digital Ocean 1&2	Distant	oui	non	oui	oui
VNC	non	oui	non	non	non

TABLE 3 – Analyse des solutions industrielles

0.3 Proposition

LA SECTION précédente a soulevé des problématiques sur l'ingénierie en ligne et les laboratoires distants.

Nos travaux se compilent sous la forme d'un cadriciel pour construire des applications de contrôle d'instruments distants pour un usage dans l'éducation en ligne ainsi qu'un usage industriel. L'application cliente est construite à l'aide de la norme MPEG-4 BIFS, pour augmenter le degré d'interopérabilité et utilise la réalité mixte pour représenter au mieux l'environnement distant.

Ce cadriciel prend en compte le travail collaboratif, i.e la possibilité de contrôler l'instrument distant à plusieurs, tout en reproduisant une conscience de groupe. Pour faciliter cet aspect, certaines modifications ont été apportés à la norme MPEG-4 BIFS.

Nous proposons enfin de modéliser les procédures d'exécution à l'aide de technologies du Web sémantique. L'utilisation d'un moteur de règle permet de déduire la procédure actuelle et de proposer à l'utilisateur les interactions suivantes possibles.

0.3.1 Cadriciel pour l'ingénierie en ligne

0.3.1.1 Architecture

Nous proposons un cadriciel pour le développement d'applications d'expérimentations distantes. Ce cadriciel comprend deux modes. Le mode d'**exécution** est effectif lors d'un contrôle distant. Il délivre une vue **augmentée** et **collaborative** de l'instrument distant. Un mécanisme pour retranscrire la "conscience de groupe" est proposé. Le système permet aussi de proposer une **adaptation** de l'IHM en fonction de la procédure d'exécution.

Le second mode est un mode d'**initialisation**. Il propose différents outils auteurs pour permettre de **formaliser** la description de la scène distante ainsi que la description des pilotes de l'instrument.

0.3.1.2 Chaîne logicielle

Pour proposer un cadriciel complet de construction de solutions de pilotage d'instruments à distance, il est nécessaire d'étudier la chaîne logicielle complète entre les utilisateurs et l'instrument distant. Généralement, une unité de traitement est reliée à l'instrument distant pour améliorer les capacités de communication de l'instrument distant. Cette chaîne est principalement composée de trois liens :

- Le lien entre l'unité de traitement et l'instrument à piloter,
- Le lien entre l'unité de traitement et le terminal de l'utilisateur,

- Le lien entre le terminal de l'utilisateur et les utilisateurs.

Pour mettre en place le lien entre l'unité de traitement et l'instrument à piloter, il est nécessaire de maîtriser un grand nombre de standards d'accès aux instruments (GPIB, USB, PXI). Cependant, le standard **Interchangeable Virtual Instrument (IVI)** permet une grande interopérabilité pour des classes d'instruments définies (oscilloscope, analyseur de spectre, etc.).

Le lien entre le terminal utilisateur et l'unité de traitement peut être réalisé grâce à l'emploi d'un intergiciel (ou middleware en anglais). Ce type de logiciel permet l'échange de messages entre composants logiciels à travers des réseaux, notamment Internet. Il est important de garder en tête le côté collaboratif de l'architecture que nous souhaitons mettre en place. L'intergiciel à mettre en place doit donc répondre aux critères suivants :

- Asynchronisme : pour garder le contrôle de l'application même si le message n'est pas délivré,
- Robustesse et scalabilité : pour ne pas limiter le nombre de personnes connectées sur l'instrument distant,
- Garantie de livraison : l'intergiciel doit assurer que les messages soient livrés dans l'ordre d'émission, surtout pour assurer le bon comportement de l'instrument à piloter.

L'IHM est le dernier lien de la chaîne logicielle. Il nous paraît important que l'IHM propose les traits suivants :

- une certaine généricité et un couplage lâche avec l'intergiciel pour faciliter la réutilisation,
- une grande fidélité de représentation de l'environnement de l'instrument distant,
- une connexion à l'intergiciel pour communiquer avec l'instrument et avec les autres utilisateurs,
- un large choix de composants graphiques (images, son, vidéos, etc.).
- un haut niveau d'interopérabilité,
- une adaptabilité en fonction d'éléments de son contexte.

0.3.1.3 Conclusion

L'architecture complète du système est proposée dans la figure 3.2. La caméra IP filme l'instrument distant. Le serveur d'application construit un multimédia interactif basé sur MPEG-4 BIFS et le délivre aux utilisateurs. Lorsque qu'une commande est passée, le serveur d'application détermine les actions possibles suivantes et renvoie les résultats à l'utilisateur. La commande est ensuite passée à l'instrument. Le mode initialisation est illustrée par la figure 3.3. Cette division permet de supporter les quatre contributions majeurs que nous voulons adresser dans ces travaux. Le "Multimedia Builder" construit la descriptions de la scène **distante** à travers une interface de **réalité**

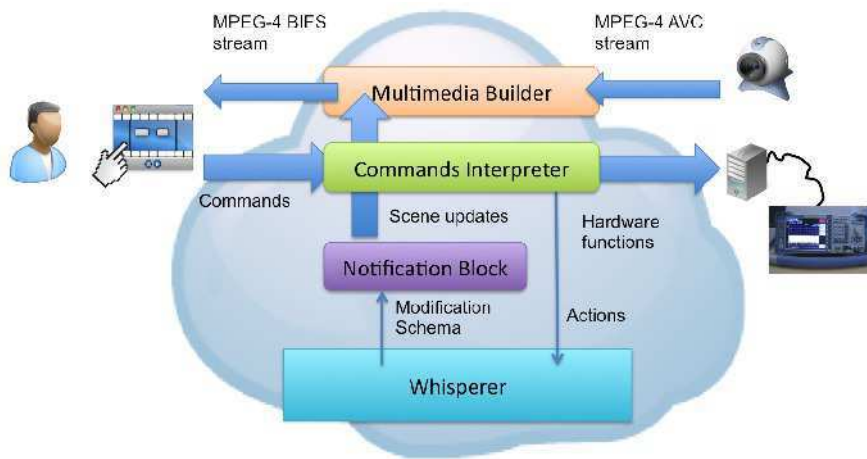


FIGURE 1 – Architecture générale en mode d'exécution.

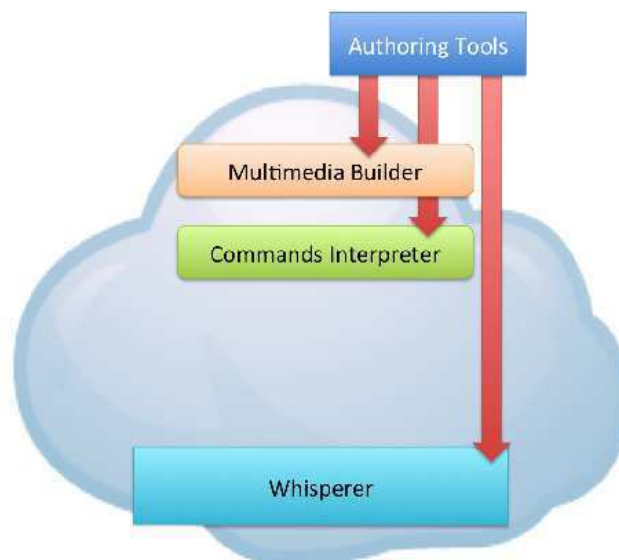


FIGURE 2 – Overall Architecture at set up.

augmentée. Les modules "Commands Interpreter" et "Notification Block" permettent le contrôle **collaboratif** de l'instrument. Le module "Whisper" propose l'**adaptation** du contenu en fonction de la procédure d'exécution. La combinaison des modules "Commands Interpreter" et "Multimedia Builder" permettent de garantir une **scalabilité** du nombre d'instruments supportés.

0.3.2 Réalité Augmentée

Pour construire l'IHM de l'instrument, il est essentiel d'analyser l'environnement de l'instrument à piloter. Il apparaît alors trois types de régions dans la scène :

- une région vidéo correspond au *feedback* de l'instrument,
- une région propose les composants graphiques de l'interface et l'interactivité nécessaire pour piloter l'instrument,
- un fond.

Ces différentes régions sont réalisées grâce aux caractéristiques de la norme MPEG-4 BIFS, à savoir une représentation des médias sous la forme d'un arbre hiérarchique, différents noeuds rendant possible l'interactivité et l'environnement de streaming pour la livraison du contenu multimédia.

Pour assurer la cohésion du monde réel et virtuel dans une application de réalité mixte, il est nécessaire de connaître à tout moment la position des composants de l'environnement entre eux. Nous choisissons donc d'effectuer un suivi de la façade de l'instrument à l'aide des descripteurs de points caractéristiques des images. Il s'agit de calculer ces descripteurs sur les images délivrées par la caméra IP et de les comparer avec ceux d'une image de référence enregistrée dans la base de données du système, ce qui permet d'identifier l'instrument présent devant la caméra.

Une fois l'instrument identifiée, les points caractéristiques sont mis en correspondance avec ceux de l'image de référence grâce à l'algorithme Fast Library for Approximate Nearest Neighbor [Muja09] (FLANN). Il est ensuite possible d'estimer la position de l'instrument dans le flux vidéo à l'aide d'homographies, calculées grâce au calibrage préalable de la caméra en suivant l'algorithme de Zhang [Zhang00]-[Zhang10]. L'homographie permet de calculer la transformation d'un solide entre un monde en trois dimensions (l'environnement en réalité mixte) et un monde en deux dimensions (l'IHM de pilotage délivrée à l'utilisateur) à l'aide d'un calcul matriciel.

0.3.3 Gestion de la Collaboration

La collaboration permet de mettre en relation des individus ou organisations dans le but d'accomplir une tâche ou de résoudre une problématique qui n'est pas facilement achevée seul.

Pour assurer la collaboration, un haut niveau de confiance entre les différentes parties doit être atteint, de même qu'une communication au sein du groupe. Ainsi, un

système collaboratif ne doit pas non seulement permettre le partage d'une ou plusieurs ressources au sein du groupe, mais également de retranscrire la conscience de groupe (groupe awareness en anglais). La conscience de groupe "est la compréhension de qui travaille avec vous, ce qu'ils font et comment vos actions influent sur les leurs" [Gutwin04]. La conscience de groupe doit donc être au centre du cadrage que l'on propose pour proposer un système collaboratif effectif.

0.3.3.1 Intergiciel

L'intergiciel doit permettre de gérer un très grand nombre de personnes connectées sur la plateforme au même moment et répondre aux critères que nous avons énoncés précédemment. Il paraît donc qu'un intergiciel orienté message (Message-Oriented Middleware ou MOM en anglais) soit la solution pour assurer ces critères.

L'intergiciel doit suivre le paradigme *publier/souscrire* expliqué par [Eugster03] qui est le suivant :

- les clients *souscrivent* à un canal de discussion,
- un protagoniste *publie* un message sur le canal,
- un *courtier* redistribue le message publiée sur le canal aux autres clients.

Pour pouvoir assurer un couplage lâche entre l'IHM de pilotage et l'intergiciel, nous avons créé deux types de canaux pour l'interaction. Le premier canal est le canal de commande. Celui-ci permet d'envoyer les commandes à l'instrument. Le second canal est le canal d'interface. Quand une action est réalisée dans l'IHM, les autres clients doivent être notifiés de cette action.

0.3.3.2 Interface Multimédia Interactive et Collaborative

Proposer un intergiciel qui supporte la collaboration est nécessaire mais n'est pas suffisant pour transcrire la conscience de groupe aux utilisateurs. Nous proposons ici les outils pour la transcrire.

Chaque utilisateur reçoit une couleur. Cette couleur est envoyée avec chaque action envoyée sur le canal d'interface. Le courtier renvoie donc la couleur associée à tous les clients connectés. Le composant graphique responsable de l'envoi de la commande est alors coloré avec celle de son activateur. Il est donc nécessaire de connaître les correspondances couleur/utilisateur, c'est pourquoi nous proposons un indicateur de télé-présence qui permet de nommer les personnes connectées et leurs couleurs associées.

Nous proposons également un système de messagerie instantanée pour que les utilisateurs puissent communiquer entre eux pendant la manipulation. Nous augmentons encore cette possibilité de communication à l'aide d'un outil de type *talkie-walkie*. Cet outil permet une communication audio sans faire l'acquisition continue de données au-

dio sur le terminal utilisateur, permettant ainsi de limiter la consommation CPU et la bande passante.

0.3.3.3 Adaptation de l'interface homme machine

Nous choisissons d'exprimer les procédures d'utilisation à l'aide du Resource Description Framework (RDF) et du Resource Description Framework Schema (RDFS). Ces formats sont à la base du Web sémantique et permettent de représenter des ordres dans des ressources, grâce aux conteneurs RDFS. Les séquences sont alors des éléments des conteneurs.

0.3.3.4 Livraison du contenu multimédia

Le format de livraison du contenu multimédia est MPEG-4 BIFS, suite à l'analyse précédente, qui permet de décrire le contenu sous la forme d'un arbre hiérarchique. Il permet d'inclure les différents composants de l'interface, tel que le flux vidéo, les éléments interactifs et le fond.

Nous choisissons de tronquer le flux vidéo pour limiter la bande passante de l'application. Cette troncature correspond à la zone exclusive de *feedback* d'un instrument, par exemple son écran dans le cas d'un instrument de mesure de type analyseur de spectre.

Pour représenter le fond, l'image de référence enregistrée dans la base de données du système est utilisée. Les trois régions de la scène (vidéo, fond et graphiques) sont alors multiplexées dans un fichier MP4. Ce format est une autre partie de la norme MPEG-4. Le contenu multimédia complet est ensuite délivré à l'utilisateur à l'aide de technique de streaming *broadcast* via le protocole **Real-Time Streaming Protocol** (RTSP).

0.3.4 Conclusion

Cette section présente l'architecture générale du cadriciel que nous proposons pour construire des solutions de pilotage d'instruments distants. Le cadriciel permet un contrôle collaboratif de l'instrument et autorise la communication entre les utilisateurs de façon audio ou via un service de messagerie instantanée. Un système de colorisation des widgets actifs permet d'améliorer la conscience de groupe.

La partie cliente, réalisée à l'aide de la norme MPEG-4 BIFS, permet de contrôler la scène distante à l'aide de technique de réalité augmentée. Le contenu multimédia est délivré aux utilisateurs via le protocole RTSP.

Le contenu est adaptée en fonction de la procédure d'exécution, formalisée dans le cadriciel à l'aide de technologies issues du Web Sémantique.

Dans la section suivante, nous détaillerons les points essentiels de l'implémentation du cadriciel.

0.4 Réalisation et Résultats

LA SECTION précédente a présenté l'architecture principale de notre cadriciel avec ses principaux composants. Dans ce chapitre nous détaillons des points d'implémentation particuliers du cadriciel dans un premier temps. Dans un second temps, nous présenterons les résultats obtenus à travers de tests utilisateurs. Le cadriciel est également réalisé sous la forme d'un projet libre (open source en anglais).

0.4.1 Réalisation et Implémentation

Nous allons détailler certains points d'implémentation du cadriciel. Nous commencerons par l'implémentation de l'intergiciel. Nous verrons dans un second temps l'implémentation des pilotes logiciels de contrôle d'instruments. Dans un troisième temps, nous verrons l'implémentation de certains points de l'interface multimédia. Nous verrons enfin l'emploi du Web sémantique pour représenter les séquences de procédure.

0.4.1.1 Pilote d'instruments

Les pilotes de contrôle ou *drivers* permettent d'envoyer les commandes spécifiques à l'instrument. Les *drivers* suivant la spécification définie par la foundation IVI sont disponibles dans plusieurs langages, notamment C#, C++, Labview ou Matlab. Nous avons choisi d'utiliser le langage C# étant donnée sa facilité de prise en main. Un autre avantage d'utiliser les drivers IVI est la possibilité d'utiliser le driver dans un mode simulation si l'instrument n'est pas disponible.

Pour autoriser et favoriser la communication entre l'intergiciel et les drivers de l'instrument, une connexion TCP par socket a été mise en place. Elle permet d'envoyer à l'instrument les messages de commandes, qui sont préalablement interprétées et de renvoyer ses réponses.

Les classes qui permettent de représenter le comportement de l'instrument ont été établies de façon à reproduire les menus et sous menus de l'instrument. Le *proof of concept* de notre cadriciel a été réalisé sur un analyseur de spectre FSL 3 prêté par Rhode & Schwarz.

0.4.1.2 Implémentation Multimédia

L'application multimédia client est réalisée à l'aide de la norme MPEG-4 BIFS. Le contenu multimédia est interprété à l'aide du projet GPAC³ qui propose un cadriciel complet pour la gestion de contenu multimédia, notamment de contenu MPEG. Il pro-

3. <http://gpac.wp.mines-telecom.fr/>

pose en outre un lecteur multimédia capable d'afficher le rendu de contenu BIFS et un multiplexeur.

L'arbre de scène général du contenu multimédia est composé de 4 principaux modules. Ces modules sont les suivants :

- la fenêtre d'authentification, commune à toutes les instruments pilotés.
- le script général, permettant la communication entre la scène multimédia et l'intergiciel. Il permet également l'animation de certains composants de l'interface.
- les composants graphiques généraux (widgets) communs à tous les instruments pilotés (l'outil de communication instantanée, le bouton de fermeture de session utilisateur, etc.).
- l'interface propre à chaque instrument.

L'identification sur la plateforme est réalisée au niveau de l'intergiciel. Le client envoie une requête HTTP comportant les champs nécessaires à l'identification. Nous sommes conscient du problème de sécurité que cela entraîne. Une amélioration possible de ce problème serait de gérer l'identification en utilisant le protocole HTTPS par exemple.

Les différents *widgets* de l'interface (généraux ou spécifiques à l'instrument piloté) sont réalisés grâce aux différentes capacités de MPEG-4 BIFS. En ce qui concerne l'interactivité, elle est réalisée grâce aux différents noeuds **Sensor** de la norme qui permettent de reproduire le comportant de boutons, de potentiomètres, etc. Certains sont mêmes capables de "sentir" les touches entrées sur le clavier du terminal utilisateur, permettant la saisie de texte (pour l'identification et pour la communication instantanée). BIFS est également capable d'afficher le résultat de courbes de Bézier, permettant ainsi de dessiner des courbes à la place de la zone vidéo lorsque l'instrument est en mode simulation.

Pour ajouter des capacités de communication, nous avons développé un module pour GPAC permettant de prendre en compte l'API WebSockets [W3C12] dans les scripts ECMAScript BIFS. Le protocole WebSocket [IETF11] est en cours de standardisation par l'Internet Engineering Task Force (IETF) et permet de mettre en place une communication bi-directionnelle entre plusieurs clients et un serveur. Cela permet entre autre au server de "pousser" vers les clients les messages du canal d'interface. Ces messages sont ensuite interprétés par le script client qui met à jour l'interface en fonction.

Nous avons également ajouté une API au BIFS ECMAScript permettant de gérer l'outil de communication audio. Cette API permet le contrôle de la prise de son aussi bien que le contrôle de l'envoi des données vers le serveur qui est responsable de les renvoyer en retour aux clients connectés.

Pour faciliter la mise en place facilité de nouvelles interfaces de pilotage d'instruments distants, nous proposons un outil auteur qui facilite la création du contenu multimédia.

0.4.1.3 Module de chuchotement

Le "module de chuchotement" permet de raisonner sur les procédures d'action actuelles et qui permet d'afficher dans l'interface utilisateur la suite possible des actions à réaliser.

La trace d'utilisation de l'interface est également sauvegardée dans le système en utilisant les conteneurs RDFS. Cela nous permet de facilement extraire la trace d'un ou plusieurs utilisateurs et ainsi d'analyser les résultats. L'analyse porte par exemple sur l'équivalence de séquence. Si deux procédures sont équivalentes que la cardinalité de la première est supérieure à celle de la seconde et que l'opérateur a utilisé la première, le système est capable, toujours grâce au moteur de règles, de signaler que la seconde procédure est moins longue à effectuer que la première.

Un outil auteur est disponible avec le cadriciel et permet d'exprimer les procédures d'exécution. L'interface de ce module est réalisé grâce à MPEG-4 BIFS.

0.4.1.4 Conclusion

Nous avons dans cette section présenté des points d'implémentation du cadriciel que nous proposons. Les *drivers* pour piloter l'instrument distant sont réalisés à l'aide d'implémentation C# de la norme IVI et communiquent avec l'intergiciel grâce à une socket TCP.

L'interface multimédia est réalisée à l'aide des caractéristiques de base de la norme MPEG-4 BIFS. Ils permettent de reproduire le comportement logique d'éléments interactifs de type boutons, potentiomètre, etc.

Nous avons également enrichi la norme MPEG-4 BIFS avec l'API WebSocket dans l'ECMAScript BIFS. Ceci met en place une communication bi-directionnelle entre les clients et le serveur et permet au serveur de notifier les clients de l'utilisation d'un widget par exemple. Nous ajoutons de même une API permettant de contrôler la prise et l'envoi au serveur de données sonore du terminal client pour pouvoir réaliser l'outil de communication talkie-walkie.

Nous avons également présenté comment les technologies du Web sémantique et de moteur de règles permettent de représenter des séquences d'utilisation d'interface et aider l'utilisateur à améliorer sa productivité.

0.4.2 Résultats

Dans cette section, nous présentons les résultats obtenus suite à nos travaux sur le cadriciel présenté pour construire des applications de contrôle d'instruments distants. Nous présentons dans un premier temps les tests utilisateurs réalisés dans le cadre du projet *MIRA*. Nous introduisons ensuite le projet libre que nous avons initialisé.

0.4.2.1 Projet MIRA

Les projets MIRA⁴ ont été créés pour renforcer la coopération dans l'enseignement supérieur et la recherche internationale. Nous avons mis en place un laboratoire distant dans le cadre d'un projet MIRA avec l'université Libano-Française de Tripoli au Liban. Cette session a été effectuée en décembre 2011 dans le cadre d'un enseignement sur le traitement du signal.

Les objectifs pédagogiques de ce cours de traitement du signal sont de donner aux étudiants des compétences en acquisition, numérisation, et représentation spatiale et temporelle de signaux. Ce cours est divisé en cours magistraux et travaux pratiques. Ces travaux pratiques sont réalisés sur des simulations et sur des contrôles d'appareils réels. Les simulations sont réalisées sur Labview. Les étudiants affichent les représentations temporelles et fréquentielles de signaux en faisant varier différents paramètres tels que la bande passante, le span, etc. Les objectifs de la manipulation réelle étaient d'afficher ces mêmes courbes en faisant varier les mêmes paramètres mais avec un signal réel en entrée, ainsi que d'apprendre à manipuler un véritable appareil industriel. La manipulation a été mise en place sur un analyseur de spectre hébergé à Télécom Saint-Étienne (Saint-Étienne, France). Quinze étudiants ont suivi ce cours.

Les données récupérées des étudiants sont encourageantes. Les étudiants ont apprécié l'idée de manipuler un appareil réel de haute qualité qu'ils n'ont pas l'habitude de voir, et cela, en étant à 2000 km de distance. L'enseignant sur place a remarqué que l'IHM proposée était très proche de l'IHM réelle de l'instrument et avait "la sensation de manipuler l'instrument". Il a donné une note de 4 sur 5 sur la facilité d'utilisation de l'interface. Il a également remarqué que la configuration pour les étudiants étaient très simple, considérant le fait qu'il faille seulement installer un lecteur multimédia et se connecter à une adresse RTSP. La note donnée est de 4 sur 5 sur la simplicité d'installation pour les utilisateurs. L'enseignant a également souligné que le temps de transmission des commandes entre les clients et l'instrument était bon, considérant la distance entre les utilisateurs et l'instrument. Le temps de transmission des commandes est évalué à 600ms. La bande passante a été évaluée à 500 kbps. Les utilisateurs et l'enseignant n'ont pas utilisé les outils de communication, étant donné qu'ils étaient tous dans la même pièce pendant la manipulation, mais ont apprécié la notification d'actions par les autres. L'enseignant a souligné l'intérêt d'un tel cadre pour le partage d'instruments avec un coût d'achat élevé.

Toutefois, l'enseignant et les étudiants ont soulevé un inconvénient. La qualité perçue du *feedback* vidéo délivré par la caméra n'était pas aussi satisfaisant que souhaité. La qualité de la vidéo a été évaluée avec une note de 2 sur 5 sur une échelle MOS. Cela peut venir de la qualité de la caméra employée pour filmer l'instrument ainsi que de la bande passante disponible qui était faible. Une plus profonde analyse MOS et de qualité d'expérience (QoE) peut être envisagée pour améliorer ces problèmes. Toutefois,

4. Mobilité Internationale Rhones-Aples

étant donné que les étudiants n'ont jamais utilisé d'appareil de ce type au préalable, seule l'opinion sur la qualité visuelle des feedbacks peut être prise en compte. L'enseignant est un expert dans son domaine, il est le seul capable de comparer l'utilisation de tels cadrage.

0.4.2.2 Un projet Open Source

La désignation logiciel libre, ou logiciel *open source*, s'applique à des logiciels qui rencontrent les critères définies par l'**Open Source Initiative**, c'est à dire la possibilité de re-distribution gratuite des sources et des travaux dérivés.

Pour augmenter le développement de notre cadrage et pour faciliter son évaluation par la communauté scientifique, nous avons décidé de publier notre cadrage sous la licence Lesser GNU Public General (LGPL). Ce projet est intitulé Open Collaborative Environment for the Leverage of Online instrumentation (OCELOT).

0.5 Conclusion et Perspectives

Nous avons présenté nos travaux sur l'instrumentation en ligne et proposé nos résultats sous la forme de tests utilisateurs et d'un projet libre. Nous présentons dans ce chapitre une synthèse des travaux, suivie d'une discussion sur nos résultats. Nous finissons enfin par proposer des pistes de recherche.

0.5.1 Synthèse

Dans ces travaux, nous nous sommes attaché à proposer un cadriceil pour la construction de solutions de pilotage d'instruments distants, utilisant les techniques de réalité mixte pour améliorer la compréhension de la scène distante manipulée, utilisant les standard multimédia de la norme MPEG pour faciliter l'interopérabilité et proposant un contrôle collaboratif de la ressource. La procédure d'utilisation est prise en compte en tant qu'élément de contexte et permet d'adapter le contenu multimédia en fonction des actions suivant la séquence actuelle.

Le cadriceil proposé est divisé en plusieurs composants qui répondent aux problématiques posées au chapitre 2, à savoir :

- les drivers IVI, responsables du dialogue avec l'instrument distant,
- l'intergiciel, responsable de convoier les messages entre l'IHM et les drivers IVI. Il est basé sur un MOM,
- le centre de notification, responsable d'envoyer les messages d'interface aux clients de l'application. Il est basé sur un server WebSocket,
- le multimédia interactif, responsable d'afficher l'IHM de l'instrument aux utilisateurs. Il est réalisé à l'aide de la norme MPEG-4 BIFS, enrichie par l'utilisation des WebSockets,
- le module de chuchotement, capable de définir les actions suivant l'action courante,
- un outil auteur, facilitant la création d'IHM pour le pilotage d'instruments.

0.5.2 Discussion

Nous allons maintenant discuter les avantages et limites de notre approche suivant différents critères.

0.5.2.1 Cadriceils

Comme nous l'avons présenté dans l'état de l'art, la plupart des solutions de laboratoires distants utilisent des technologies de type Flash, Java, Labview, etc. Nous proposons d'utiliser MPEG-4 BIFS (avec des enrichissements) pour améliorer l'inter-

opérabilité entre les terminaux utilisateurs tout en mettant en place une solution de réalité mixte.

Ce cadriciel peut être également utilisé pour d'autres types d'applications telles que des services d'IPTV interactifs, vidéo conférence, etc.

0.5.2.2 Multimodalité

Les travaux sur la multimodalité des IHM soulignent le fait qu'elle peut améliorer les processus cognitifs et d'apprentissage et améliorer l'immersion de l'utilisateur. Notre approche se base uniquement sur le terminal utilisateur pour interagir avec l'instrument distant, i.e. clavier/souris pour un ordinateur classique et interface tactile pour un téléphone ou une tablette et pourrait en effet bénéficier d'interactions multimodales (commande vocale par exemple).

Mais cela pourrait aussi diminuer la compréhension de l'instrument manipulé en ajoutant des modalités qu'il ne supporte pas de base. Nous pensons fortement que pour apprendre à manipuler un appareil réel, les interactions naturelles avec l'instrument doivent être reproduites.

0.5.2.3 Collaboration

Tandis qu'iLab ne propose pas de contrôle collaboratif dans son cadriciel, la collaboration et la conscience de groupe sont présentes dans notre approche, même si les outils mis à disposition ne sont pas toujours exploités.

Les réseaux sociaux pourraient renforcer également les processus de collaboration. Notre système fournit des outils de communication mais ne fournit pas d'informations sur les utilisateurs à proprement parler (intérêts, localisation, etc.). Un scénario possible serait d'envisager l'intégration d'OCELOT avec un système de type LMS tel que Moodle par exemple. L'intégration des technologies du Web sémantique, notamment le projet FOAF (Friend Of A Friend) peut être une piste de recherche, puisqu'il décrit les utilisateurs et leurs relations respectives.

0.5.2.4 Limitations

Toutefois, notre système souffre de quelques limitations. Les drivers actuels ne prennent en compte que la norme IVI, ce qui implique un développement non négligeable à considérer lors du déploiement de nouveaux instruments. En considérant les retours des étudiants, la qualité perçue de la vidéo n'est pas suffisante. Cela pourrait être amélioré en employant une caméra de meilleure qualité ou en adaptant le contenu vidéo en fonction de la bande passante par exemple. L'actuel développement du cadriciel ne prend en compte que des instruments à façade (à deux dimensions). Étant donné

que BIFS gère également les contenus à trois dimensions, on peut également considérer un portage des applications vers des instruments à trois dimensions dans une future version.

0.5.3 Perspectives

Le cadriciel que nous avons présenté pourrait être étendu à d'autres types de cas d'usage, par exemple la chirurgie collaborative à distance, etc., ce qui doit impliquer des travaux sur l'intégration d'instruments à trois dimensions.

Étant donné que le cadriciel supporte la collaboration, des recherches sur les interactions sociales peuvent être menées.

Des recherches sur l'adaptation de l'application au contexte utilisateur peuvent également être menées pour améliorer la qualité d'expérience du service (bande passante, terminal disponible, taille de l'écran, localisation, etc.).

Des recherches actuelles dans la livraison de contenu multimédia tendent vers le streaming HTTP plutôt que le streaming plus classique RTSP. Le but du streaming HTTP est de faciliter la livraison du contenu en passant par les ports Internet ouverts par défaut, contrairement aux autres solutions de streaming.

Table of Contents

Foreword	i
Thank-you note	i
Résumé long	iii
0.1 Introduction	iii
0.2 État de l'Art	vi
0.3 Proposition	xiii
0.4 Réalisation et Résultats	xx
0.5 Conclusion et Perspectives	xxv■
Table of Contents	xxix■
1 Introduction	1
<i>Problems, objectives and structure of this thesis</i>	
1.1 Once upon a time...	3
1.2 Online Engineering and Remote Laboratories	6
1.3 Organization of the manuscript	13
2 State of the Art	15
<i>Related works : principles and formalization</i>	
2.1 Technologies	18
2.2 Remote Laboratories and Online Engineering	43
2.3 Synthesis	51
3 Proposal	57
<i>The heart of the proposal</i>	
3.1 Online Engineering and Computer Aided Maintenance Framework	59
3.2 Augmented Reality	65
3.3 Handling Collaboration	73
3.4 Adaptation of the Human Computer Interface	77
3.5 Multimedia Delivery	81
3.6 Conclusion	88

4	Realization and Results	89
	<i>What we have so far</i>	
4.1	Realization and Implementation	91
4.2	Results	116
5	Conclusion and Perspectives	121
	<i>Next step in Online Engineering</i>	
5.1	Synthesis	123
5.2	Discussions	125
5.3	Perspectives	127
6	Spectrum Analyzer implementation : FREQ mode	129
	<i>C# class for IVI driver</i>	
7	Features and descriptors graphs	133
	<i>Charts of features detection</i>	
8	BF and FLANN matching graphs for SURF features descriptors	145
	<i>Charts of descriptors matching</i>	
	Glossary	155
	Bibliography	169

List of Figures

1	Architecture générale en mode d'exécution.	xv
2	Overall Architecture at set up.	xv
1.1	Tryptic of online engineering.	4
1.2	Remote experimentations software chain.	7
2.1	Sensorama Machine by Morton Heiling (1957).	19
2.2	Mixed Reality Continuum [Milgram94].	20
2.3	A marker extracted from the ARToolKit library.	21
2.4	SIFT examples.	24
2.5	SURF examples.	25
2.6	BRIEF examples.	26
2.7	ORB examples.	26
2.8	X3D examples.	37
2.9	SVG examples.	38
2.10	WebGL examples.	39
2.11	BIFS examples.	40
2.12	VISIR client application.	44
3.1	Remote experimentation architecture.	60
3.2	Overall Architecture at runtime.	64
3.3	Overall Architecture at set-up.	65
3.4	Multimedia Chain Delivery.	66
3.5	A Spectrum Analyzer.	67
3.6	Display extraction.	68
3.7	Checkerboard with detected corners.	69
3.8	Homography principle.	71
3.9	Publish/Subscribe paradigm.	74
3.10	Collaboration and Messages delivery.	75
3.11	Whisper module.	78
3.12	MPEG-4 BIFS graph example.	81
3.13	Unicast streaming.	84
3.14	Broadcast streaming.	84
3.15	Multicast streaming.	85

3.16 Delivery.	86
4.1 Interactions modes.	94
4.2 BIFS Scene composition.	96
4.3 Authentication Screen.	97
4.4 Button pseudo-graph.	98
4.5 Button details pseudo-graph.	98
4.6 Drawing of Bezier curves using XCurve2D node.	100
4.7 BIFS Interface.	102
4.8 Chat tool and Tele-presence indicator.	105
4.9 ORB and BRIEF extraction.	106
4.10 SURF and SIFT extraction.	107
4.11 Brute Force and FLANN matching.	108
4.12 Whisper module implementation.	108
4.13 RDF Graph of an action.	109
4.14 RDF Graph of an action in the trace.	111
4.15 Sequence representation.	112
4.16 Authoring tool.	113
4.17 Sequences Editor	114
4.18 OCELOT running on an Android tablet and on a laptop.	118
4.19 Sequence diagram.	119
7.1 Agilent E4407B.	134
7.2 Agilent E4407B : BRIEF and ORB extraction.	134
7.3 Agilent E4407B : SIFT and SURF extraction.	134
7.4 Agilent 54622D.	135
7.5 Agilent 54622D : BRIEF and ORB extraction.	135
7.6 Agilent 54622D : SIFT and SURF extraction.	135
7.7 Anritsu 37397A.	136
7.8 Anritsu 37397A : BRIEF and ORB extraction.	136
7.9 Anritsu 37397A : SIFT and SURF extraction.	136
7.10 HP 8561A.	137
7.11 HP 8561A : BRIEF and ORB extraction.	137
7.12 HP 8561A : SIFT and SURF extraction.	137
7.13 HP 54510A.	138
7.14 HP 54510A : BRIEF and ORB extraction.	138
7.15 HP 54510A : SIFT and SURF extraction.	138
7.16 HP 54600B.	139
7.17 HP 54600B : BRIEF and ORB extraction.	139
7.18 HP 54600B : SIFT and SURF extraction.	139
7.19 HP E5060A.	140
7.20 HP E5060A : BRIEF and ORB extraction.	140
7.21 HP E5060A : SIFT and SURF extraction.	140

7.22	LeCroy WaveJet 324A.	141
7.23	LeCroy WaveJet 324A : BRIEF and ORB extraction.	141
7.24	LeCroy WaveJet 324A : SIFT and SURF extraction.	141
7.25	Hameg HM407-2.	142
7.26	Hameg HM407-2 : BRIEF and ORB extraction.	142
7.27	Hameg HM407-2 : SIFT and SURF extraction.	142
7.28	Tektronix TDS 210.	143
7.29	Tektronix TDS 210 : BRIEF and ORB extraction.	143
7.30	Tektronix TDS 210 : SIFT and SURF extraction.	143
8.1	Agilent E4407B.	146
8.2	Agilent E4407B : brute force and FLANN matching.	146
8.3	Agilent 54622D.	146
8.4	Agilent 54622D : brute force and FLANN matching.	147
8.5	Anritsu 37397A.	147
8.6	Anritsu 37397A : brute force and FLANN matching.	147
8.7	HP 8561A.	148
8.8	HP 8561A : brute force and FLANN matching.	148
8.9	HP 54510A.	148
8.10	HP 54510A : brute force and FLANN matching.	149
8.11	HP 54600B.	149
8.12	HP 54600B : brute force and FLANN matching.	149
8.13	HP E5060A.	150
8.14	HP E5060A : brute force and FLANN matching.	150
8.15	LeCroy WaveJet 324A.	150
8.16	LeCroy WaveJet 324A : brute force and FLANN matching.	151
8.17	Hameg HM407-2.	151
8.18	Hameg HM407-2 : brute force and FLANN matching.	151
8.19	Tektronix TDS 210.	152
8.20	Tektronix TDS 210 : brute force and FLANN matching.	152

List of Tables

1	Descripteurs de scènes multimédia	ix
2	Analyse des solutions pour l'éducation en ligne	xi
3	Analyse des solutions industrielles	xii
2.1	Scene Descriptors review	41
2.2	Research project in e-Learning	52
2.3	Research project in industrial maintenance	53
3.1	RDF containers	79
4.1	Features Descriptors : rotation and scale invariance	107
4.2	Open source realization	118

The value of a prototype is in the education it gives you, not in the code itself.

Alan Cooper

1

Introduction

▷ *This chapter is the introduction of this thesis. We discover issues of online engineering and remote laboratories. We also present the objectives, our motivations, the scientific perimeter of this study as well as the structure of this document.* ◁

Chapter Plan

1.1	Once upon a time...	3
1.2	Online Engineering and Remote Laboratories	6
1.2.1	Motivations	6
1.2.2	Software Chain	7
1.2.3	Online Engineering and its usages	8
1.2.4	Use Case scenario	10
1.2.5	Arising Problems	10
1.2.6	Conclusion	11
1.3	Organization of the manuscript	13

WE consider modern remote computing as the period after the expansion of the Internet. Over the last five decades, application architectures have evolved significantly. Applications used to provide limited services, such as data access. With the rapid growth of the Internet, systems have become more and more complex and provided more advanced features. Of course, the development of terminal capacities have also improved the application architectures, since terminals, even mobile terminals, are nowadays able to process complex data.

1.1 Once upon a time...

Between the 60s and the 80s, the mainframe architecture was the dominant. In this architecture, all the components are stored on one central computer. Users are connected to the same central unit and share the same data processing. Terminals are passive. Performances of mainframe architectures rely on the central unit capacities, and because of their importances, the teams in charge must be highly qualified. The proprietary evolution of the system implies that changes are complex and costly.

The evolution of application architecture between the 80s and the 90s brought the "client-server" architecture. The system is distributed between two parts : the server is providing services and the clients are consuming them. Parts of the overall data processing can be achieved on the client side. Communication between parts are reached through local network or Internet. In this architecture, client applications are said to be "heavy". This architecture is also known as *two-tiers architecture*. The major problem arising with this architecture is the evolution of the system. Its modification entails a very large amount of time for configuring all client applications. This leads to the appearance of Web architecture in the middle of the 90s.

Because of maintenance issues with the need to process more and more complex data and the need for better security, applications evolved to *three-tiers architectures*. The application server is separated into two parts : application and data. The client application requests the application server. The application server requests and obtains data from the data storage server, processes it and returns results to the client. Web applications are three-tiers application where the client applications are executed in a Web browser. Since all calculations are executed on the server-side, Web applications are referred to as "thin client". Deployment and evolution do not require the configuration time of two-tiers applications.

During the last decade, Web applications have emerged with the same look and behavior as desktop applications. They are usually referred to as **Rich Internet Applications** (RIA). They are usually delivered via specific browsers and plugins. Often, a framework for executing RIAs is installed in the client system in order to execute it. Some of the most popular framework are **Asynchronous Javascript and XML** (AJAX), **Adobe Flash**, **Microsoft Silverlight**. The major advantages of RIA are :

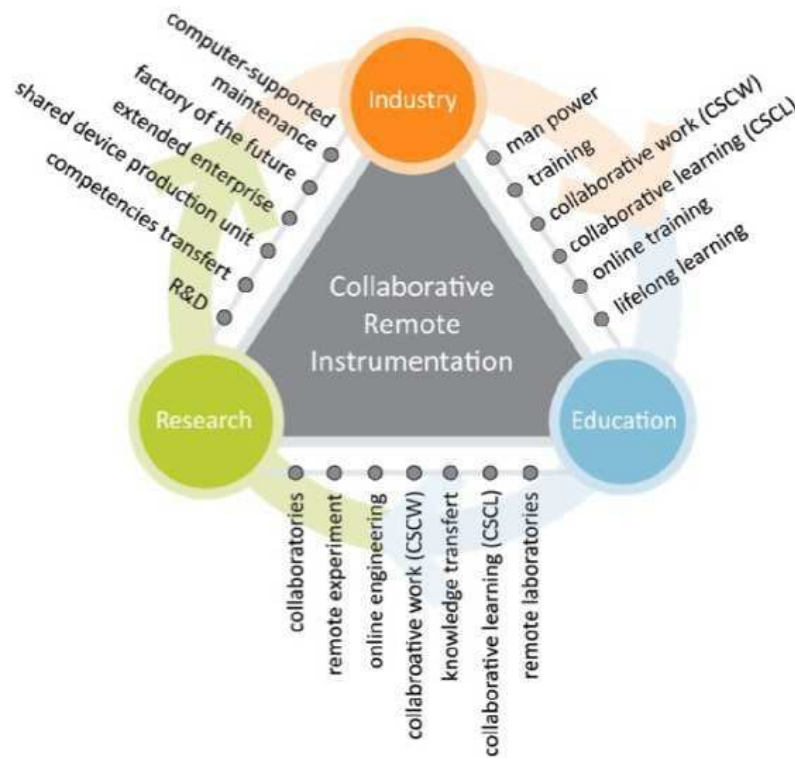


FIGURE 1.1 – Tryptic of online engineering.

- Enhanced ergonomics.
- Possibility to adapt the interface depending on one's business.
- Security may be improved using sand-boxed solutions that limit the access to the user's system storage.
- Possibility to embed animations and multimedia content.

The communication capacities of machines coupled with the development of the Internet made the remote control of devices and their representations in the applicative world possible.

Relationships between humans and machines evolved, whether in real controls of devices or simulations. This contact is not physical but happens through computer systems. We talk about computer-assisted systems, since computers are not only responsible for displaying the **Human Computer Interface** (HCI) that users interact with, but also can help users in their tasks in various ways.

Leaving aside the potentially intrusive aspects of computers, it is important to notice that innovation is not only technical. Information Technology, (IT), can effectively combine multiple people to multiple machines, as opposed to bijective previous relation-

ship. This is done to connect mixed processes but also collaborations between people, in order to achieve a shared objective. This scientific area is referred to as **Computer Supported Collaborative Work (CSCW)**, that "*focuses on the study of tools and techniques of groupware as well as their psychological, social, and organizational effects*"¹.

Tele-operations allow to conduct manipulations that could not occur otherwise. Actors of such systems come from different areas and have together different relationships, as illustrated in Figure 1.1. The remote control of devices is nowadays employed in different fields, but especially in :

- Learning institutes mainly for distance learning curriculum. In the field of e-learning (meaning online curriculum), online engineering is most of the time referred to as **remote laboratories**.
- Industries for maintenance operations, production controls, etc. In the e-maintenance area, such systems are usually referred to as **Computer-Assisted Maintenance (CAM)**.
- Research institutes to manage experimentations with researchers from various geographic locations (collaboratories).

The evolution of the Internet and the applications architecture allow new paradigms for online engineering. The next section presents the subject of this study.

1. sources : wikipedia : <http://en.wikipedia.org/wiki/CSCW>

1.2 Online Engineering and Remote Laboratories

The study of this thesis is focused on online engineering and remote laboratories. This section presents the motivations of using remote laboratories and illustrates these motivations through two use case scenario. In a second step, we present the problem arising with online instrumentation. Finally, the organization of this document is presented.

1.2.1 Motivations

1.2.1.1 Financial factor

The cost of state of the art devices is often very high considering their usage frequency. It also implies the presence of highly qualified operators *in-situ* in order to manage the machine park. The alternative, consisting in moving devices, entails technical as well as responsibility difficulties (insurance for instance). The usual solution consists in moving the operators to the site of the devices.

This solution entails other arising issues, such as administrative tasks, access demands for the site, accommodation, etc. Thus, remote control of devices can lead to saving direct as well as indirect money (time² for instance). Despite the possible outcomes, we must however consider the development time and cost for online engineering solutions.

In our work, concerned devices come mainly from research laboratories. Such devices have a high cost of purchase as well as of maintenance. They are bulky and present fragility of transport.

1.2.1.2 Overcome constraints

The financial cost **is not the only constraints** that could lead to an online engineering solution. Other factors, such as the moving of people or devices are fundamental. Laboratory sites are not always suitable for the hosting of such groups. Problems arising come from different criteria :

- Security : of people and goods, insurance contract respect, etc.
- Accessibility : limited number of people in the same room, disabled people, etc.
- Availability : of people.
- Observability : infinitesimal or infinitely large.

Online engineering solutions are not bounded to decrease the financial constraints of instrumentation.

2. "Time is money". Benjamin Franklin.

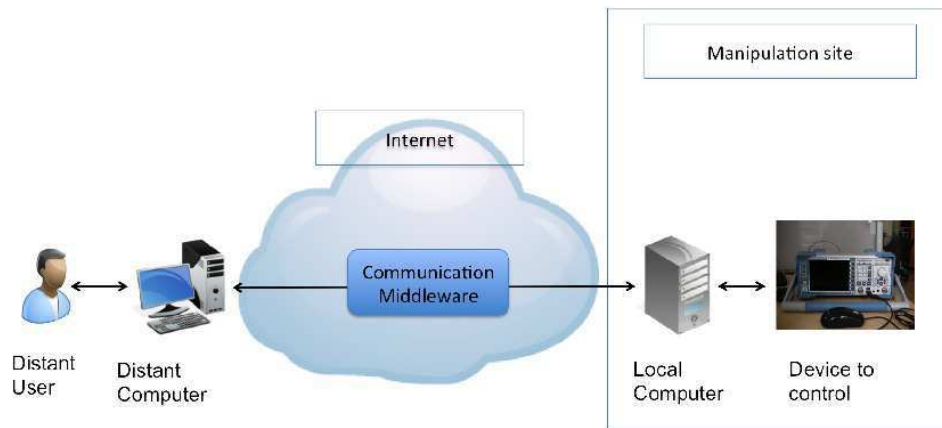


FIGURE 1.2 – Remote experimentations software chain.

1.2.2 Software Chain

In this subsection, we review the main software architecture of remote experimentations. Figure 1.2 shows the end to end software chain used nowadays. This chain, derived from n-tiers software architectures, is composed of three main links :

- The link between the local computer and the device to control.
- The link between the distant computer and the local computer. It is the middleware responsible for the application discussion between the remote computer and the local computer.
- The link between the distant computer and the distant user : the Human Computer Interface.

1.2.2.1 Local Computer to Device Link

The link between the local computer and the device entails mastery of standards and of access protocols. This is complex due to a) the heterogeneity and the number of technologies and b) the number of material interfaces allowing interfaces with devices : PXI, USB, GPIB, etc.

But since some devices from different constructors or from different series allow the same kind of functionality, some efforts were made in order to propose common drivers through standards. Those drivers define *classes* of devices, in order to ease interoperability between devices and to decrease the development time and cost.

Unfortunately, those standards suffer from limitations. [Landragin06] exposes some limitations (translated from French source) :

- "It is clear that results will not be identical between devices as measurement performances are very different, even if each seems to work perfectly in the environment".
- "It is obvious that devices that combine specific functionalities, unique in their category [...] do not ensure interchangeability".

1.2.2.2 Remote Computer to Local Computer Link

This link is referred to as *middleware*. The goal of the middleware is to convey, using the Internet, commands or requests from remote users to devices. It is also responsible for transmitting data from the device to users.

[Bernstein96] defines a middleware as a link between the application layer and its operating system. Its main goal is to convey messages between applications, being the most transparent both for the user and the developer. The middleware is then in charge of conveying the commands from the remote user to the device, and also of giving a behavior of the collaborative platform, i.e. conveying the messages from the devices to all the users.

1.2.2.3 Human Computer Interface

The Human Computer Interface is the software component that enables users to interact and communicate with the system. HCIs must present items to send commands to the remote device as well as items to receive feedbacks from the remote device.

1.2.3 Online Engineering and its usages

1.2.3.1 Collaboratories

Collaboratories can be defined as follows :

"[a] center without walls, in which the nation's researchers can perform their research without regard to geographical location, interacting with colleagues, accessing instrumentation, sharing data and computational resources, and accessing information in digital libraries" [Cerf93].

In order to spread results and experimentations, to exchange ideas or to ease communication between researchers, tools are needed. We can also notice that issues in the scientific world involve more and more interdisciplinary skills. A network of researchers can then occur in collaboratories. Online engineering and the remote control of devices can help and improve exchanges in the scientific world.

1.2.3.2 Remote Maintenance

Industrial processes, such as maintenance can also benefit from online engineering. Online engineering help industries to execute maintenance procedures on remote devices. Production and industrial maintenance sites rely on highly qualified technicians and staff. The latter are responsible for many assembly/disassembly operations, calibration, etc. These human operators are expected to understand and memorize a large panel of procedure steps, in a specific order. Moreover, moving to the production site presents several issues (cost, risk, etc.). As a consequence, a first maintenance operation is usually performed remotely.

Learning and being able to reproduce the full catalog of maintenance operations can be a difficult task. The complexity of this task is also rapidly exploding since the number of devices operated by a single human operator increases when these devices can be operated remotely.

1.2.3.3 e-learning

A lot of abroad exchanges happen during college studies, especially in the engineering field. In Europe, several exchange programs exist between universities. Among them, the Erasmus program³ is probably the most famous. This program was launched in 1987 and enables students to spend part of their studies in a partner university in Europe. More than 1.5 million students took benefit from this program. Students are asked to follow courses and pass their exams in the exchange university. Foreign courses must fit their home studies. But sometimes, it is necessary to validate home courses. This could happen if the host institution does not provide a particular lecture. Then students need to validate their home courses online.

In distance learning, lectures, exercises or projects are easily brought online. It is more difficult with practical laboratories. Meanwhile, they are a compulsory learning material in engineering studies, since they bring students practical skills and know-how. Bringing laboratories online aims at offering learners the possibility to manipulate devices from different locations. It allows reducing geographical laboratories constraints : "If you cannot come to the lab, the lab will come to you"[Harward04].

Industries and research laboratories can benefit from the possibility to manipulate online devices. They both could use research and training service delivery. Research as well as training service deliveries are punctual, and the financial cost of devices could be reduced to a minimum when manipulating devices online.

3. European Region Action Scheme for the Mobility of University Students - <http://www.europe-education-formation.fr/erasmus.php>

1.2.4 Use Case scenario

The previous subsection presented the ins and outs of tele-operations. Indeed, we saw that the remote control of devices is not only a financial issue, but brings a technological answer to manage otherwise intractable situations. Stakeholders make up a trio composed of laboratories, learning institutes and industries. In order to illustrate our works and proposals, we put forward two use case scenario in this subsection. They help illustrating needs and issues of the remote control of devices. The first use case scenario focuses on online engineering for e-learning while the second focuses on remote maintenance.

1.2.4.1 e-Learning

Steve is a teacher in Norway. Heather and Jimmy are both Steve's students. As part of their engineering curriculum, Heather and Jimmy are studying abroad for the current year. Heather is in Tokyo and Jimmy is in Paris. But in order to graduate, they still need to validate Steve's lecture, which includes practical sessions on a spectrum analyzer and on a vector network analyzer.

1.2.4.2 Industrial Maintenance

Alice is a technician working in a manufacture. Her device presents several malfunctions. She calls Bob who is an operator in a company performing maintenance services. In order to decrease the maintenance service cost, Bob first runs remote operations on the device to avoid in situ maintenance, if possible. But since Bob conducts maintenance on several devices for different companies, he does not know by heart all the procedures for all the machines. He uses the CAM system provided by his company to run the supervision that includes all the procedures for the many devices he is in charge of.

1.2.5 Arising Problems

We saw that online engineering and remote laboratories try to answer issues that are not easily solvable when controlling device locally. They also raise issues that need to be taken into account. Previous use cases raised some of these issues, which are presented in this subsection.

1.2.5.1 e-learning

The e-learning use case illustrates the need of online engineering. But it implies several issues as follows :

- *Human Computer Interface*. During the practical sessions, Heather and Jimmy are meant to learn how to use both analyzers in order to use their skills after graduation. Giving a high fidelity representation of the HCI of the analyzers improves Heather and Jimmy's understanding of the device.
- *Interoperability and Pervasive Computing*. With the expansion of different kinds of terminals and platforms (smartphones, tablets, computers, etc.), Steve wants to provide a system with the highest interoperability, so that he does not have to worry about Heather and Jimmy's terminal. Since distance entails latency (Heather, Steve and Jimmy are in different time zones), Steve also needs to provide a solution that limits the bandwidth and that is accessible anytime and anywhere.
- *Collaboration and group awareness*. *In-situ* practical sessions are often done by groups in order to achieve more complex tasks. Bob wants online practical sessions to work in a collaborative manner, so that Heather and Jimmy can achieve together the same goals as their *in-situ* comrades. Therefore, in order to reproduce the in situ collaboration, Steve's system needs to provide notifications of actions done by each users to give the group sensation. Notifications alone are not sufficient to reproduce a real collaboration, the system also needs to provide communication tools.
- *Reusability and simplicity*. For his practical sessions, Steve is using two different devices. Therefore, the system must be as much reusable as possible with the least efforts to build new online practical sessions. He also wants the shortest initialization.

1.2.5.2 Industrial Maintenance

In addition to the above-mentioned issues, the industrial maintenance use case raises others :

- *Procedure formalism*. The CAM system must provide a formalism to describe procedures.
- *Business Logic*. The business logic determines which procedure is being performed and what the next possible actions are. This logic must be loosely-coupled with any programming language in order to be fully reusable.
- *Analysis*. The CAM system should provide tools to analyze actions that have been made by the operators, in order to propose improvements.

1.2.6 Conclusion

Online engineering is motivated by the need to reduce financial costs as well as the need to answer technical issues such as visibility, dangerousness, availability, etc. It is mostly used by three kind of organizations : industries, learning institutes and laboratories, with their own needs and usages.

We presented two use case scenarios that illustrates online engineering situations. The first one is centered on e-learning. The second one focuses on industrial maintenance. Both of them raises issues.

From our point of view, these issues are of the utmost importance when designing online engineering. The work we conducted tries to answer them and both scenarios will be use in this document in order to illustrate our proposition. We can notice that these issues strongly impact the Human Computer Interface presented to the end user and are compiled as follows :

- a remote control of the device that presents a high level of immersion and understanding of the remote scene.
- a collaborative control of the remote device.
- an adaptation of the HCI in order to hint the operator.
- a scalable system able to control a wide range of different devices.

1.3 Organization of the manuscript

This chapter introduced the issues of online engineering. We have presented how the Internet and its mutations made it possible to bring devices online. We have presented the needs of bringing devices online and the actors of online engineering. In the second part of this chapter we have presented two use case scenarios that helped illustrating issues of remote control of devices, especially in the e-learning and the industrial maintenance area.

The need of high fidelity of the representation of the device has lead us to consider mixed reality techniques. Interoperability, meaning the possibility to manipulate the device from any kind of terminals has lead us to the usage of multimedia scene descriptions.

This document is organized as follows. Chapter 2 presents the state of the art of technologies and projects in the online engineering field. Chapter 3 presents the overall architecture of the system we propose in this work. Chapter 4 details implementation points of our solution and presents the results we obtained. Chapter 5 concludes the work and provides prospects.

*It is easier to move a problem around (for example, by moving the problem to a different part of the overall network architecture) than it is to solve it.
It is always possible to add another level of indirection.*

Ross Callon – RFC1925, “The Twelve Networking Truths”

2

State of the Art

▷ *While the previous chapter has raised the need for remote laboratories, we make in this section a review of online engineering and industrial maintenance systems.* ◁

Chapter Plan

2.1	Technologies	18
2.1.1	Augmented Reality	18
2.1.2	Collaboration	30
2.1.3	Context Awareness and Semantic Knowledge	31
2.1.4	Multimedia Scene Descriptions	35
2.1.5	Conclusion	41
2.2	Remote Laboratories and Online Engineering	43
2.2.1	e-Learning	43
2.2.2	Industrial Maintenance	47
2.3	Synthesis	51
2.3.1	Analysis	51
2.3.2	Towards Collaborative and Contextualized Online Instrumentation	54

REMOTE Laboratories and Online Engineering bring the possibility to perform manipulations on real devices over the Internet. With the rapid growth of the Web and its associated technologies, controlling remote devices has become to reality.

Instruments' feedbacks need to be represented to operators in order have an understanding of the operation state. This can be achieved on the client-side by representing feedbacks with graphical components in the Human Computer Interface or by displaying a video capture of the remote device. Anyhow, HCIs presented to operators require a high fidelity. Mixed reality could be a way to improve the user immersion and understanding of the apprehended remote devices.

In the pervasive computing area, remote laboratories are expected to be accessible anytime and anywhere. This implies a high level of interoperability in order to control devices from any kind of terminal, from personal computers to mobile devices able to display multimedia content. Reusability, i.e. the ease of deployment of new devices online, should also be considered when building Remote Laboratories frameworks. The difficulty of HCI construction is a story deadlock for remote laboratories development due to resources costs. Loose coupling between the instruments hardware commands and the HCI presented to operators is then necessary and client-side applications need to rely on standards to ensure interoperability, for both graphical and communication aspects.

Collaboration, i.e. the possibility to perform remote operations within a group, is a key component of online engineering. Sharing resources and interacting with them at in the same time is not enough, the "group awareness" needs to be taken into account.

In the previous chapter, we have raised four main issues of remote experimentations. The first part of this chapter presents notions and technologies in order to overcome these issues.

The second part of this chapter presents the state of the art research activities of online engineering and remote laboratories usage in industry and learning areas. We focus our analysis on the four criteria defined previously.

The third part of this chapter summarizes the analysis.

2.1 Technologies

In chapter 1, we have presented the four main issues we try to answer in this thesis :

1. remote control of the device that presents a high level of immersion and understanding of the remote scene.
2. a collaborative control of the remote device.
3. an adaptation of the HCI in order to hint the operator.
4. a scalable system able to control a wide range of different devices.

In this section, we review the technologies and notions that lead to in the resolution of such issues.

The first issue can be answered using augmented reality technologies. Therefore, section 3.2 presents the augmented reality paradigm and its associated technologies. Section 2.1.2 presents the issues that need to be taken into account when designing collaborative systems. Context-aware systems are systems able to react and adapt their behaviors according to context items. In section 2.1.3, we present such systems. Finally, in order to address the wider range of different devices, a description of the device environment has to be chosen. Therefore, a formalism to describe the device environment, such as multimedia scene description is needed. Section 2.1.4 presents multimedia scene descriptions.

2.1.1 Augmented Reality

Since the advent of Personal Computers, WIMP graphical interfaces (Windows, Icons, Menus and Pointing devices) have established themselves as the dominant interaction style. But WIMP interfaces are nowadays proving their limits in numerous situation (critical work, simulation, disability, etc.). Furthermore a virtual world is created, in complete opposition to the real world.

The real world is the world in which users evolve whereas the virtual world encompasses storage, communication and data processing capabilities. Since about twenty years ago, computer scientists have tried to bring together the real and the virtual worlds, leading us today to the mixed reality paradigm.

2.1.1.1 Mixed Reality Continuum

A virtual reality system is a computed model in which graphics are used to create a world that seems realistic. This modeling is not static since it answers users' orders (gesture, speech, external commands, etc.). The virtual reality notion exists since 1957 with the invention of the Sensorama Simulator by Morton Heilig. It was composed of a 3D video system, fans and a vibrating chair. Fans were located near the user's head



FIGURE 2.1 – Sensorama Machine by Morton Heiling (1957).

and provided wind and odors. The vibrating chair gives the sensation of movements. So it was possible to simulate motorcycle or car trips, as illustrated with figure 2.1.

Virtual reality is a simple way for humans to visualize, manipulate and interact with complex data with the help of a computer. Augmented reality regroups all the techniques allowing to associate a real world to a virtual world. It is a view of a real scene where computed information is added. Added Information can consist of texts, sounds, graphics, etc.

Augmented reality was born with Sutherland's works on virtual reality head mounted device in the middle 70's [Sutherland68]. The "See through System" displayed simple superimposed polygons on the view of the real world. The use of such a head mounted device displaying superimposed virtual objects on a real world view allows users to keep the contact with the real world while being in a virtual world.

Three fundamental rules must be respected in order to build an effective augmented reality application [Swan05] :

- Combine the real world and the virtual world.
- Take into account the consistency and the coherence between the real and the virtual worlds.
- Handle the constraints of real-time and interactivity.

This definition excludes simple 2D superpositions that do not respect the 3D coherence as well as post-production composition, which is not real-time. On the other hand, no constraint is set the photometric realism of compositions. For instance, virtual objects could be overlaid.

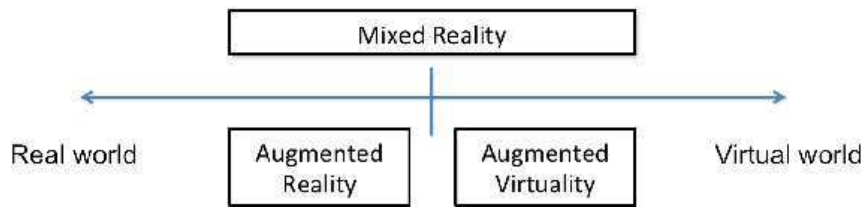


FIGURE 2.2 – Mixed Reality Continuum [Milgram94].

Augmented reality is a view of a real scene where computed information is added. By extension, augmented virtuality refers to the merging of real objects to a virtual scene. Augmented reality and virtuality can be expressed in a more global : term mixed reality. Mixed reality encompasses both augmented reality and virtuality [Milgram94], as shown in Figure 2.2.

Different kinds of devices and sensors are employed to realize augmented reality systems, and by extension, mixed reality systems. Of course, the display systems are mandatory in augmented reality applications. They enable visual rendering and the generation of virtual scenes. We can distinguish between different classes of such display systems :

- Head Mounted Devices : users hold a system on the head, such as helmets. They are divided in two sub-classes. The first ones are composed of an LCD screen coupled with a semi-transparent mirror and the mix between real and virtual objects is done by the eyes. The second sub-class are the video see-through head mounted devices. The mix is done with a combination of computed graphics and images produced by cameras.
- Classical screens : a camera delivers the real objects and a computer mixes real and virtual items.
- Projectors : the visualization is done on a surface in the real world. This surface can be a table, a wall, etc.

Before displaying the mix between virtual and real objects, a proper environment model has to be created. This model helps placing real and virtual objects in the scene. Most of the time, it is a 3D geometrical model. As well as proposing a proper system for visualizing data and a proper environment model, augmented reality systems must take into account the reality-virtuality alignment. Both real and virtual items must be correctly superimposed so that users feel that virtual and real worlds coexist.

2.1.1.2 Tracking Sensors

In order to realize such alignment, spatial references objects are needed in the scene. These objects are very important because they enable the computation of the spatial representation of the scene, needed at anytime. In the augmented reality fields, the

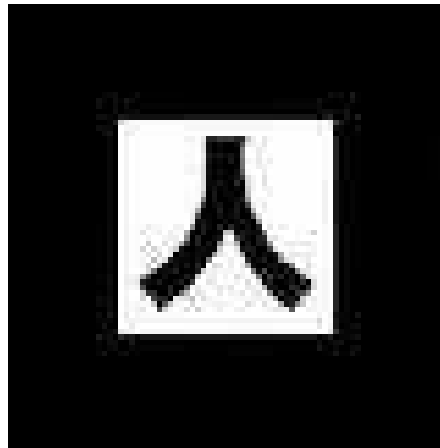


FIGURE 2.3 – A marker extracted from the ARToolkit library.

tracking of objects consists in following their orientation and position. Tracked objects can be the user himself or particular objects in the scene. Most techniques are derived from either marker-based tracking or marker-free tracking.

2.1.1.2.a Marker-based Tracking

Visual Makers

The tracking with markers consists in placing markers, i.e. a target objects, in a scene and detect their positions and orientations in every image. Markers must be easily identifiable, i.e. with minimum calculations, in order to increase the system efficiency. If the relative position between an object (real or virtual) and the marker is estimated, one can determine the absolute position of this object in the scene.

Figure 2.3 is an example of a marker extracted from the famous ARToolkit library¹. ARToolkit is a C library, which includes several models of two-dimensional fiducial markers. This library is based on the work of Kato & Billinghurst [Kato99].

Zhang proposes in [Zha02] a review of marker-based tracking. The HOM marker system is developed at Siemens AG for photogrammetry purposes. A 6 bits encoding side bar is added to the main square in order to improve the detection robustness. These HOM markers are used in "a Windows-based application for creating, updating and managing 3D computer models of factories from input digital images" [Navab99]. He also presented two other kind of markers developed with the ARVIKA² project, which aims to "research into and to realize Augmented-Reality-Technologies (AR) which will support development, production and servicing with relation to complex

1. <http://www.hitl.washington.edu/artoolkit/>

2. <http://www.arvika.de/www/e/home/home.htm>

technical products in a user-oriented and application driven manner". The first marker system is the IGD marker. An IGD marker "is a square divided into 6×6 square tiles of equal size. The inner 4×4 tiles are used to determine the orientation and the code of the marker". The second marker system is the SCR marker, also coded with inner matrix. They are designed as 4×4 grids with features points inside the boxes.

Markers may be in other forms than squares. Cho et al. presented in [Cho99] ring shaped fiducial markers. Naimark et al. [Naimark02] suggested a "2D barcode fiducial design that can generate thousands of different codes and can be used for wide area tracking".

But as mentioned by Zhou et al. in [Zhou08], after 2002, "there were no new general marker based systems presented". Their assertion is based on a analysis of the papers published in the International Symposium on Mixed and Augmented Reality. Their article has been published in 2008.

Invisible Makers

Another approach is to place invisible markers in the scene to analyze. In [Daniel11], Burnett et al. propose to using infra-red markers. Such markers are interesting because they are invisible for users but can be tracked easily with infra-red cameras and can be used in varied light conditions.

Active radio frequency identification (RFID) chips can be used for augmented reality systems[Zhang11]. They can be positioned inside structures to allow in situ positioning.

The arising technology Near Field Communication (NFC) is even used in order to create indoor navigation. NFC "is a bidirectional short range, wireless communication technology" between two devices, based on RFID [Ozdenizci11]. Nowadays, more and more mobile devices allow the NFC technology.

2.1.1.2.b Marker-free Tracking

Mechanical tracking

The first head mounted device designed by Sutherland [Sutherland68] used a mechanical tracking method. "Either of two head position sensors, one mechanical and the other ultrasonic, is used to measure the position of the user's head". The users movements were limited to a "working volume of head motion about six feet in diameter and three feet high", and "beyond these limits, head position cannot be measured by the sensor". Such method, with improvements, are still in use nowadays.

Positioning Systems

Some tracking methods use the Global Positioning Systems (GPS), gyroscopes and accelerometers. They have been widely used in mobile augmented reality applications

since the appearance of nowadays' smart phones and their capabilities. This method uses satellites signal in order to compute the geographical location of the user. Gyroscopes and accelerometers determine the vision line of the user [Erifiu11].

Interest Points Detection

The visual marker-free tracking methods are more contemporary than the marker-based ones. They are mostly based on image analysis and understanding and can be classified into two different classes.

The first class of detectors concerns the corner detectors. In 1988, Harris et al. presented the corner detector known as the "Harris Corner Detector" [Harris88]. The Harris Corner Detector is based on the Moravec [Moravec80] corner detector. The idea behind the Moravec Corner Detector is to analyze a pixel neighborhood (a window) and to determine the changes of intensity in this neighborhood when the window moves in different directions. The FAST corner detector by Rosten et al. [Rosten05] is an other famous corner detector. A point p is considered as a corner if there exists an arc of at least nine pixels that are all brighter or darker than p in a circle around p .

The second class of detectors uncludes the Blob detectors. They use local extrema of responses of certain filters as interest points. For instance, the Difference of Gaussian (DoG) detector uses differences of Gaussians as filter [Lowe04].

Feature Descriptors

In addition to interest points detection, some methods add descriptors. Such descriptors are descriptions of visual features of images such as color, shape, texture, motion, etc. Tuytelaars et al. [Tuytelaars08] describes the ideal local feature with the following properties :

- Repeatability : a high percentage of features must be matched between two images of the same scene taken from different viewing conditions (point of view, light, etc.). The features must be invariant to image deformations, noise, compression artefacts, etc.
- Distinctiveness : descriptors should distinguish features efficiently.
- Locality : the local property "reduce the probability of occlusion" and "allow simple model approximations of the geometric and photometric deformations between two images taken under different viewing conditions (e.g., based on a local planarity assumption)".
- Quantity : The number of detected features must be high enough to detect small objects and match them between images of different viewing conditions.
- Accuracy : "The detected features should be accurately localized, both in image location, as with respect to scale and possibly shape".
- Efficiency : The detection of features should take into account "time-critical applications".

Fayolle et al. have presented in [Fayolle00] a multi scale detection of feature points using wavelets. "Multiscale edges are defined using smoothed versions of the signal

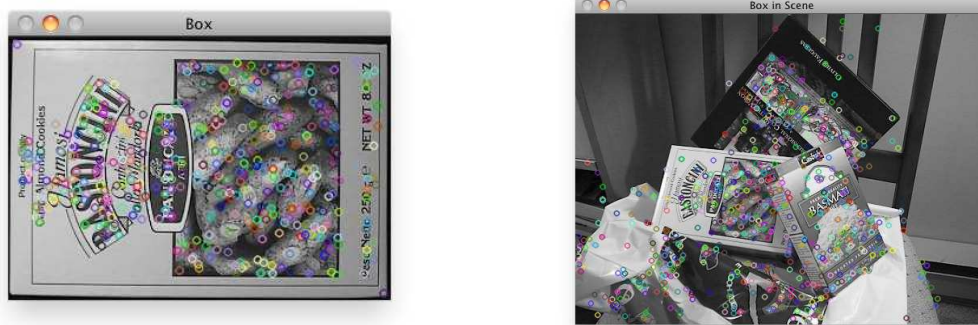


FIGURE 2.4 – SIFT examples.

and a first- or a second-order derivative". Feature points are detected studying "the behavior across scales of wavelet transform coefficients of the gradient phase signal". When detected, Fayolle et al. characterize feature points by the length and the amplitude of the phase transition of gradient phase signal. Edges routes are calculated using the Freeman coding [Fayolle98]. In order to determine the correspondence between two images, a similarity measure is computed. It is based on computing the distance between the gradient vectors of the feature points. "The distance used is a combination of the phase difference and the modulus difference of the vectors".

Among others, the Scale-Invariant Feature Transform (SIFT) by Lowe [Lowe04] is probably the most famous one. The first step of the algorithm is to detect circular areas around key points. In order to determine key points, the image is filtered at different scales with Gaussian filters. Key points are then local extrema of the DoG-blurred images. Descriptors are then computed by attributing key points local orientations determined from gradient orientation in the key point neighborhood. The SIFT method is used in a large variety of applications. In Figure 2.4, a box is represented alone and in a scene. The SIFT key points are extracted and displayed.

In 2008, Bay et al. [Bay08] presented the Speeded-Up Robust Features (SURF) descriptor. Tuytelaars et al. categorize this descriptor as efficient, because its implementation privileges the execution speed, SURF being 5 times faster than DoG [Tuytelaars08]. SURF is partially inspired by SIFT. In order to determine the interest points of an image, its integral is calculated. The interest points are determinants of the Hessian matrix applied to 2D Haar wavelets responses in scale space. The descriptions of each interest point is computed from the local distribution of the gradient orientation. The SURF descriptor is presented in the form of a 128-dimensional vector. Similar to figure 2.4, Figure 2.5 shows the extracted SURF key points.

Calonder et al. [Calonder10] proposed in 2010 a binary representation of descriptors. The goal is to decrease the size of descriptors. The original is firstly smoothed in



FIGURE 2.5 – SURF examples.

order to reduce noise with a 9×9 pixels box. The interest points are then selected using the test response τ defined as follow :

$$\tau(P; x, y) = \begin{cases} 1 & \text{if } I(P, x) < I(P, y) \\ 0 & \text{otherwise} \end{cases}$$

where $I(P, x)$ is the pixel intensity of the filtered patch P of size $S \times S$ at $x = (u, v)$. Then "choosing a set of n_d (x, y) -location pairs uniquely defines a set of binary tests". The BRIEF descriptor is the n_d -dimensional bit string that corresponds to the decimal counterpart of :

$$\sum_{i=1}^{n_d} 2^{i-1} \cdot \tau(P; x_i, y_i)$$

BRIEF has limited invariance to in-plane rotation. In order to overcome this limit, some derivation of BRIEF were achieved [Calonder11]. Figure 2.6 shows the extracted BRIEF key points.

In 2011, Rublee et al. presented an alternative to SIFT and SURF [Rublee11]. The key point detector is based on the FAST detector while the descriptor is based on BRIEF. Its name is ORB for Oriented FAST and Rotated BRIEF. Since FAST "does not produce a measure of cornerness", Rublee et al. employed "a Harris corner measure to order the FAST key points". They also produced FAST features on a pyramid of the original image, filtered by Harris, in order to have multi-scale features. In order to bring rotation invariance to BRIEF, Rublee et al. steered BRIEF according to the orientation of keypoints. However, ORB is not scale invariant. This means that the descriptor is efficient, but only on a defined and constant distance between the camera and the object. Figure 2.7 shows again the box, with its extracted ORB key points.

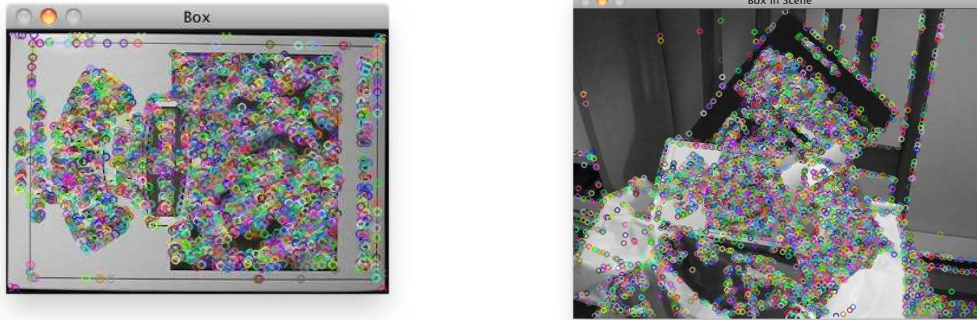


FIGURE 2.6 – BRIEF examples.



FIGURE 2.7 – ORB examples.

2.1.1.2.c Conclusion on Tracking Methods

In order to propose an efficient reality-virtuality alignment, spatial references objects in the scene are needed. Such objects can be the user itself, textures, rigid objects, etc. In order to determine the position of such objects in the scene, a augmented reality system should provide an efficient tracking method.

We have divided the tracking methods into two classes. The first class encompasses marker-based tracking, while the second one includes marker-free tracking.

Marker-based tracking can be realized using visual or invisible markers, such as infra-red markers. Visual markers are patterns easily detected using the minimum image processing. Tracking methods with markers are robust and efficient, but they suffer from an essential drawback : the marker itself. Adding markers in the scene could lead to a "pollution" of the scene. In some application environments it may not be possible to know measured locations of fiducial markers and so any approach that relies on them could not be used. Military battlefield applications or, generally, unfamiliar outdoor environments come to mind when considering this limitation.

Marker-free tracking methods is an alternative to marker-based tracking. Some methods uses mechanical or magnetic sensors, GPS and gyroscopes or also image processing. Image processing-based methods propose key points detection and sometimes key points descriptors. Descriptors permit to increase the matching of objects between two images of different viewing conditions but come at a time-consuming price.

When designing augmented reality applications, a trade-off between tracking and real-time has to be make made. Choosing the "good" tracking method depends on the application to develop.

2.1.1.3 Augmented Reality Applications

Augmented reality is a recent research area and is becoming more and more popular. Several applications are using augmented reality in different fields, from games to medicine. The overall objective of augmented reality is to improve the user's perceptions and interactions. We are now presenting augmented reality applications in different fields.

Entertainment

Entertainment and culture industries are more and more attracted by augmented reality applications.

Chien et al. in [Chien10] propose a colored-marker augmented reality system with 1-D barcode reader. The markers used in their application is a composition of a colored marker and the barcode of products. If the system detects the marker in the scene, a

3D graphics model is displayed in the scene, representing the product to analyze. This system is intended for advertisement and promotion.

In order to create a museum display platform, Chen et al. [Chen11a] propose the Augmented-Holograms Display Platform (AHDP). This system is a combination of optical holograms and tangible user interfaces. It makes use of a Web camera to track the user's head movement using computer vision in order to project augmented images on holograms according to the head's position.

Marshall proposes in his thesis report an Android application to navigate through Stockholm to discover Nobel's prize related places outside the walls of the museum [Marshall11]. The augmented reality tracking system is based on the positioning systems of mobile devices and uses their cameras and displays to add pieces of information in the scene.

Takacs presents a panoramic broadcasting for cultural presentation and entertainment services [Takacs11]. This project aims at bringing augmented streaming panoramic video onto mobile platforms. This project has been tested for various applications such as sport, bridge walkabout with added information³, etc. This application is based on Flash technologies.

Games

The games industry also perceives the benefits of augmented reality applications. They enable the users' better immersion.

Cheock et al. in [Cheok04] present Human Pacman, an augmented reality game based on the famous Pacman. Users evolve as teams, both as the characters of the Pacmen and Ghost. In this game, "players are provided with custom-built wearable computers" and head mounted devices to visualize the game world.

Not only mobile games are taking the benefits of augmented reality. Yamabe et al. proposed an augmented reality system for "traditional" games such as table games (poker, Go game, etc.) [Yamabe11]. The system is composed of a projector and a Web camera linked to a computer. The projector superimposes the feedbacks on the table (score, player turn, etc.) while the Web camera detects the position of game elements (cards, Go stones, etc.).

Serious gaming also comes with augmented reality benefits. The ARPuzzle is a "simple game that consists of six equal parts that represent parts of City University campus in London" [Liarokapis10]. The components of the City Campus are superimposed onto the table-top environment. Users receive information about the campus while trying to resolve it through text, audio or graphics feedbacks.

3. <http://www.panocast.com/>

Military

In the military area, simulation benefits through augmented reality have been proved. Gelenbe et al. in 2003 [Gelenbe03] presented a simulation environment for a military simulation area. A multi-agent system is coupled with the augmented reality application in order to provide intelligent behavior of injected artificial entities.

The Battlefield Augmented Reality System (BARS) presented in 2000 by the Naval Research Laboratory in Washington DC (USA) is a famous example of augmented reality application for military usage [Julier00]. BARS uses a head mounted device to display data, a wearable computer and a wireless network. In order "to prevent information overload", BARS proposes a mechanism to prioritize the information to add in the user's view depending on the location, the mission, etc.

Due to the confidential aspects of military systems, it is not easy to find a deeper analysis of existing solutions in the literature.

Health Care

Virtual surgery training implies a realistic simulation. Chen et al. in their works [Chen11b] studied in their works interactions involving vision and tactile sensations. They came up with force feedback modules for cutting functions. They modeled this behavior with collision detection algorithms. The 3D model implemented in this study is based on Computed Tomography images and was implemented for cranium surgery.

In endoscopic surgery, Suzuki et al. [Suzuki10] wanted to improve the "general function of the surgical robot system". The robot embeds a camera. The endoscopic images are displayed on a monitor and are augmented with additional information such as haptic sense information, patient information and navigation data.

In health care, augmented reality is used to "evaluate and appreciate the angio-architecture of the nidus with regards to position and spatial relationship of critical feeders and draining veins" [Ng09], or "assisted skull surgery with ultrasound tracking" [Alusi97] or even in order to "understand the blood vessel network of a patient" [Hutchison10].

2.1.1.4 Conclusion

Augmented reality is the combination of real and virtual objects in a scene. It can improve the understanding of the apprehended environment by adding extra or rendering invisible information to the user.

In order to build effective augmented reality applications, three fundamental constraints have to be taken into account : a) combine the real and the virtual world, b) consider the consistency and the coherence of the real and the virtual world and c) take into account the real-time and the interactivity of augmented reality applications.

In order to build an augmented reality application, a proper modeling environment has to be chosen as well as a tracking method to place objects inside this environment. Most of the time, a 3D environment is preferred. Marker-based tracking methods consist in placing targets that are easily detectable in the environment. Targets can be infra-red markers, RFID chips, fiducial markers, etc. Marker-free tracking methods represent an alternative to marker-based tracking. Some methods use mechanical or magnetic sensors, GPS and gyroscopes or also image processing.

Augmented reality can be applied in many areas. In the culture area, augmented reality applications enable users to have a better interaction, such as in museum applications. In the games area, augmented reality improves the immersion of players. In the military area it helps soldiers to have invisible data on the mission field while in health care it can give assistance to surgeons. The learning and maintenance areas are not represented in this short review on augmented reality applications, since they are the subject of a deeper analysis in following sections of this chapter.

2.1.2 Collaboration

Work is often naturally collective and collaborative. This means that it often makes many actors interact with each other in order to reach a common goal. It is however important not to mistake *collaborative work* and *cooperative work*. Cooperative work is a cooperation between actors in order to reach a common goal while sharing tasks. Collaborative work is done from the beginning to the end without tasks division. Its results depend on the motivation of people to collaborate, on the number of actors and on their skills. To ensure collaboration, a high level of trust has to be achieved and a frequent communication has to take place.

Of course, when applied to online engineering, collaboration also depends on the time and on the location of the actors. Collaboration can be synchronous (collaboration occurs at the same time for every actors) or asynchronous. It also can be centralized or distributed if people are located in the same place.

In learning theories, socio-constructivism [Jonassen91] is the theory that tries to make students learn from their own observations and from the interactions with others, both teachers and other students. In distant collaborative work, that entails that each actor is aware of the others. In online engineering that implies taking into account the "group awareness". As said in [Gutwin04], "Group awareness is the understanding of who is working with you, what they are doing, and how your own actions interact with theirs".

A forum can be seen as a distant asynchronous collaborative system. Users are able to communicate, and each actions is recorded and seen by other users in the form of a history.

The Google documents⁴ are examples of synchronous distant collaborative systems. In order to reproduce the group awareness, a tele-presence indicator is presented to users. A history of each actions can be seen in order to understand "who did what".

Communication takes a great part of collaborative systems. In order to achieve communication, different tools are used, such as Bulletin board, e-mail, chat, audio/video conferencing, etc. [Bafoutsou02]. Each communication tool brings its different functionalities and services. For example, e-mails allow an asynchronous possibility to communicate. Chats allow a synchronous way to communicate. E-mail notifications allow the possibility to notify an actor of changes in a project, without the possibility to answer. Whiteboards "allow two or more people to view and draw on a shared drawing surface. This may be used for discussing or describing objects, which are difficult to verbalize".

Bafoutsou et al. classify communication tools in four categories. The first one is textitgroup file. It involves working with files. Actors share a view of files with a possibility (or not) to edit, manage and store the shared documents in the system database. The second category is *computer conferencing* that provides tools to communicate in a synchronous or asynchronous manner. Whiteboards or forums are examples of those tools. *Electronic meeting systems* represent the third category. Most of the time, this category entails real-time communication such as audio/visual. The *Electronic workspace* is the last category. The basic idea underlying it is the possibility to "offer a common space to coordinate and organize" the group work. Anyway, some tools can be classified in different categories.

It is clear that collaborative work is based on information and resources sharing. Depending on the situation, different tools can be used in order to produce the group awareness. In the field of online engineering, a collaborative system should allow the possibility to control remote devices at the same time. It entails that real-time communication is required for this situation, but is not enough. Communication tools does not reproduce the group awareness, i.e. the knowledge of "who did what".

2.1.3 Context Awareness and Semantic Knowledge

In this subsection, we expand our vision on remote laboratories to remote laboratories able to adapt their HCI depending on their environments. Systems able to propose an adaptation depending on their context are referred to as *context-aware applications* or *context-aware systems*.

4. <http://docs.google.com>

2.1.3.1 Context and Context aware systems

A definition of context, extracted from a dictionary, is the following : "*the circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood and assessed*" [Press12].

This means that around a fact gravitates a group of elements that explicit it. The context is then the interpretation of a fact through a filter composed of elements. This means that each fact has its own context that allows to interpret it.

In the scientific literature, each work dealing with context starts with defining its own definition of the context. Definitions of context are relative to the goal of the application. This practice leads to a large diversity of definitions [Bazire05], all preserving a common pattern, that leads to the definition of the context by Dey [Dey00] :

"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves".

This definition is still the reference, even if it does not take into account the temporal aspect of context. Among these specifications of context, the notion of "context-aware computing" [Schilit94] divides the context of systems into three categories :

- Computing context (bandwidth, terminal resources, etc.).
- User context (profile, geo-location, etc.).
- Physical context (noise, lightning, etc.).

We do not intend to redefine context-aware systems or to provide a different approach to handle context-aware systems, but rather to define a key element of the context of remote laboratories we tried to answer in this thesis. This element can be classified within the *Computing context* : the procedure of control of the device. We think the orders of the actions taken by the operator, i.e. the user's activity on the HCI, is an element to take into account in the context of remote laboratories. In the following subsections, we will analyze our intention to propose HCI adaptation depending on the user activity.

2.1.3.2 Adaptation depending on the procedure

In the field of Computer-Aided Maintenance, online engineering help industries to execute maintenance procedures on remote devices. Production and industrial maintenance sites rely on highly qualified technicians and staff. The latter are responsible for many assembly/disassembly operations, calibration, etc. These human operators are expected to understand and memorize a large panel of procedure steps, in a specific order. Moreover, moving to the production site presents several issues (cost, risk, etc.). As a consequence, a first maintenance operation is usually performed remotely. Learning and being able to reproduce the full catalog of maintenance operations can be a difficult task. The complexity of the task is also rapidly exploding since the number of

devices operated by a single human operator increases. CAM systems must provide a formalism to express such procedures in order to ease operators actions and increase their productivity. As a consequence, on the Human Computer Interface side, the CAM systems have to provide indications on which action has been made and "whisper" the operator the next possible actions in the maintenance procedures. The aim is to limit the number of human errors, as well as to increase the efficiency (time spent, quality, etc.) of the maintenance process.

The scenario of remote maintenance with Alice and Bob showed the necessity of proposing a formalism to express sequences of actions. It also showed that the business logic determines which procedure is being performed and what the next possible actions are. This logic must be loosely-coupled with any programming language in order to be fully reusable. In order to achieve this loose-coupling rules engines system shall be used. After a review of related works, rules engines will be introduced. Following, we present Semantic Web technologies and reasoning capacities when coupled with rules engines. In the last paragraph, we present the necessity to develop a formalism to express maintenance procedures.

2.1.3.3 Related Works

Works were conducted in the literature in order to propose adaptation. In [Kock11], the authors propose a system to monitor and interpret sequential activities in e-Learning environments. Their system is based on pattern mining and data clustering. The goal is to offer a personalization of e-learning services based on the results of the analysis. Sequences are represented using Discrete Markov Models. This approach is interesting for e-learning processes since the system is able to determine if users have successfully completed the problem. However, this approach is not suited for remote maintenance since the sequences of actions are not pre-determined but computed afterwards.

The authors of [Schmidt08] propose an architecture for RIA user interface adaptation. The system uses ontologies to annotate the application, and to describe the user model. It also takes advantages of the Semantic Web for mining adaptation rules. The adaptation rules are located on the client side and are designed using OWL-DL and SWRL.

In [Lung09], the authors propose a complete conceptual framework for remote maintenance. This framework helps designing e-Maintenance applications using five abstraction levels such as strategic vision, business process, organization, service, data structure, and IT infrastructure. The sequences are designed as workflows. The framework does not describe the formalism of the sequences and the way to help operators in maintenance operations.

In [Lindstaedt09], the authors propose a user model based on ontologies. This model is used to characterize users in order to adapt Web application using rules. They

designed and implemented a rule-driven agent, which tries to capture interesting patterns of user-system interactions. This approach is user-centered but not on processes of pre-defined actions.

In [Vargas-Vera08], the authors provide a framework for a personalized e-learning system. The goal is to adapt the system depending on user context. This framework is using ontologies and rules.

2.1.3.4 Rules Engine

A business rules engine is a software component, the role of which is to execute pattern matching between facts and business rules. Business rules is "An atomic piece of re-usable business logic, specified declaratively" [Ross11]. This definition contains four important words :

- *Atomic* : a business rule cannot be divided under the penalty of loss of semantic value.
- *Re-usable* : a business rule is not specific to one application but can be shared by multiple applications.
- *Business* : rules are based on business, therefore they are intended to business experts. The latter has to manage the rules. A non-technical representation model is required.
- *Declaratively* : rules are presented in a declarative way.

An example of business rule can be the following :

```
A customer who orders more than 10 products during a
calendar year is a good customer.
```

A business rule is made of conditions, e.g. "A customer who orders more than 10 products during a calendar year", and decisions, e.g. "is a good customer". Its purpose is to automate a decision.

Rules engines enable a loose-coupling between business rules and programming components. They entail a high level of portability and reuse.

In the literature, many e-Maintenance systems are built on top of Semantic Web technologies [Muller08]. They provide a high level abstraction and description of resources. Coupled with rules engines, they offer reasoning capabilities. In the following paragraphs, we explain our motivations for the use of Semantic Web technologies.

2.1.3.5 RDFS/RDF and RDFS Reasoning.

The Resource Description Framework (RDF) is a graph-based model designed to describe in a formal way resources on the Web, along with their metadata to allow the

automatic treatment of such descriptions. RDF is a W3C specification [W3C04a] and is the skeleton of the Semantic Web. One of its serializations uses XML format.

RDF Schema (RDFS) is also a W3C specification [W3C04b]. It furnishes bases vocabulary to structure RDF resources.

We can find several RDFS reasoners. Jena is a popular Java framework for building Semantic Web applications. It proposes an API for reading, processing and writing RDF data in XML. It also includes a RDFS rule-based reasoner based on Datalog and RETE [Forgy82]. This rules engine allows to write rules and to obtain reasoning on RDF resources.

4rs [4rs06], 4store reasoning, integrates 4store's SPARQL and SPARQL/Update implementations with RDFS backward chained reasoning.

IRIS [IRIS06] is another RDFS reasoner also based on Datalog.

2.1.3.6 OWL and SWRL.

OWL stands for Ontology Web Languages. It is the most widely spread specification for ontologies and is divided into different levels of expressiveness OWL Lite, OWL-DL and OWL Full, ordered by increasing expressiveness. OWL-DL is the most widely used sub-language of OWL, because of its decidability. Expressing rules over ontologies axioms can be achieved using Semantic Web Rule Language.

SWRL, Semantic Web Rule Language, "is based on a combination of the OWL-DL and OWL Lite sublanguages of the OWL Web Ontology Language the Unary/Binary Datalog sublanguages of the Rule Markup Language. SWRL allows users to write Horn-like rules expressed in terms of OWL concepts to reason about OWL individuals" [O'Connor05]. It uses first order logic in order to perform reasoning.

Different frameworks are also available to deal with SWRL. Among others we have : Pellet [Pellet12] that is an open source Java OWL-DL reasoner that supports SWRL, Hoolet [Hoolet06] or SWRLTab that supports editing and executing SWRL rules in a Protege plugin-in.

Context-aware systems propose schema in order to achieve adaptation. Most of them rely on Semantic Web technologies. Semantic Web technologies allow describing resources and their links. Coupled with rules based engines, reasoning can be achieved.

2.1.4 Multimedia Scene Descriptions

In order to address a wide range of different devices, a formalism to describe the remote device environment has to be chosen.

In Latin language, *media* is the plural of *medium*. In computer science, a medium is any kind of support of information. The term *multimedia* grew at the end of the 80's with the appearance of Compact Disc - Read Only Memory (CD-ROM). It designated applications able to generate or control together different kind of media such as sound, video, images, etc. We prefer defining a multimedia as :

Definition (Multimedia). *Any kind of media related to each other with digital, logical, temporal or semantic relationship presented according to the author's choices.*

In order to carry on such relationships, it is necessary to master the media's presentation. It is achieved by incorporating media in multimedia scenes. Multimedia scene descriptions enable the instructions for staging media. Concolato, based on Scherp et al. works on scene description [Scherp05], defines the properties of scene descriptions [Concolato07] :

- the employment of audio-visual media or objects (natural or synthetic).
- the spatial organization of objects (relative and absolute) in 2D or 3D environments
- the temporal organization of objects (over time evolution)
- interactivity capabilities

A multimedia scene description is usually represented as a scene-graph. Different scene-graphs and scene description languages have been created. With the rapid growing Internet phenomenon, interoperability is needed in order to ease the broadcast of multimedia content.

2.1.4.1 Scene Description Formats

Multimedia scenes descriptions enable the instructions for staging media. In this section, we will present some of the most common description languages and formats.

VRML

The Virtual Reality Modeling Language (VRML) is one of the first description languages. Its second version is standardized within the standard organization ISO and is called VRML 2.0 or VRML97 [ISO/IEC-1477297].

This description language is oriented to the creation and visualization of 3D scenes. VRML proposes a hyperlink mechanism to create files and to refer to external ones. Interactivity is achieved using special nodes named *Sensors* and scripts supports in the *Script* node.

But VRML lacks the support for 2D graphics as well as compression. Streaming content is not allowed either. A third version of VRML has been published. This version enables to create VRML content in a XML manner and adds various improvements in 3D primitives. It is called Extensible 3D (X3D). Figure 2.8 shows two examples of X3D scenes.

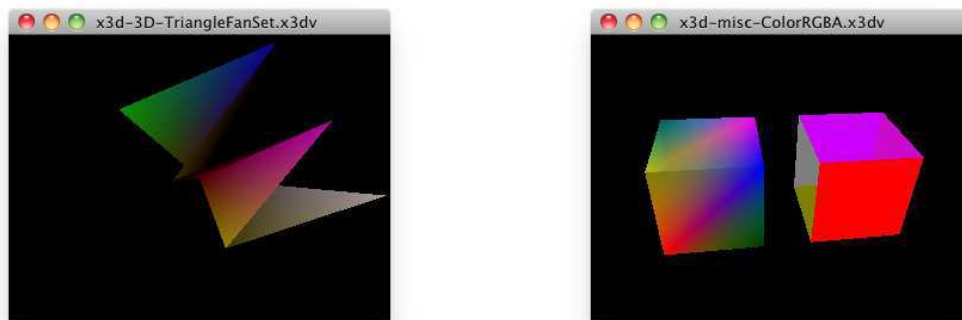


FIGURE 2.8 – X3D examples.

SMIL

The Synchronized Multimedia Integration Language (SMIL) is a W3C specification [W3C08]. Its first version was published on the 15th June of 1998 and its last version, SMIL 3.0, was published on the 1st of December 2008. This language defines how to present multimedia objects in a declarative and synchronized manner.

SMIL is based on the XML syntax allowing the presentation of different media objects such as images, video, audio and text. It provides support for 2D graphics but not for 3D graphics. Like VRML, SMIL lacks compression support.

SVG

The Scalable Vector Graphics (SVG) is also a W3C specification [W3C11b]. The second edition of its version 1.1 was published in August 2011. SVG is used to describe 2D scenes and graphical applications by using 2D primitives and vector graphics. It entails that objects can be viewed on any resolution with the best possible quality. Figure 2.9 shows two examples of SVG 2D graphics.

SVG supports ECMAScript. This characteristic enables to manipulate all objects of the scene and to perform animation on them.

As opposed to SMIL and VRML, SVG can be compressed using the gzip⁵ standard, producing the SVGZ format. No streaming support is provided.

COLLADA

Sony Computer Entertainment released in 2004 the initial version of COLLABorative Design Activity (COLLADA). Its current version, 1.5, is released by Khronos [Khronos08]. This description language is widely accepted, especially in the game industries.

5. <http://www.gzip.org/index-f.html>

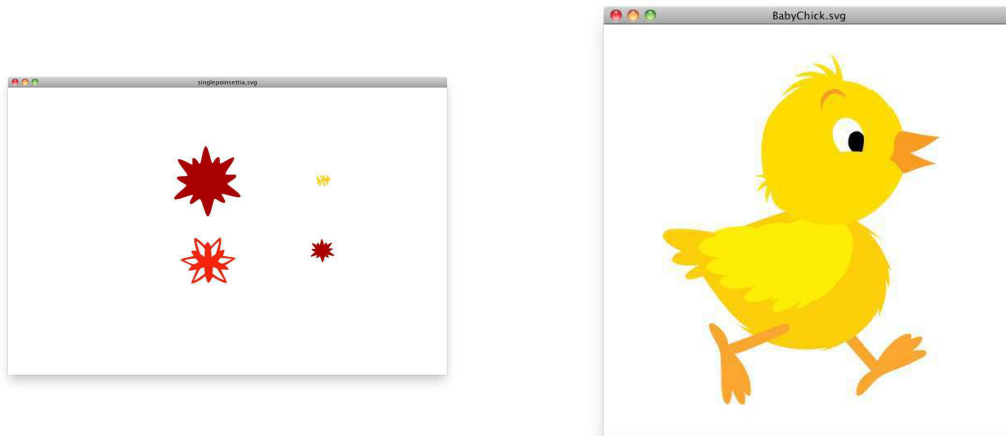


FIGURE 2.9 – SVG examples.

COLLADA also has an XML syntax. It supports both 2D and 3D primitives but does not support video or sound media. It does not support interactivity, nor streaming of data. Hence, COLLADA is mainly used for storing data.

Flash

Flash is a proprietary solution by the Adobe Systems company [Adobe12]. A lot of Web sites are using the Flash format in order to present elements in the page, such as advertisement or games. It enables to improve the interactivity, the dynamism and the animation of Web pages. That is why Flash is very popular.

Flash designates the authoring tool, the data formats to this tool as well as the publishing format. The Flash publishing format is a binary format.

Flash supports 3D scenes since its 11th version, published in October 2011, but this support is only restricted to personal computer for the time being. Flash supports scripting through ActionScript, a derivative of ECMAScript and proposes a compression of the scene.

Flash is probably still the most widespread format on the Web, even if HTML and its fifth version could possibly dethrone it, according to W3C and companies behind it.

JavaFX

JavaFX⁶ is a software platform developed by Oracle. The current release version is 2.0 and has been distributed since October 2011. Java FX is intended for Rich Internet Applications and the scene representation is also a scene-graph.

6. <http://javafx.com/>

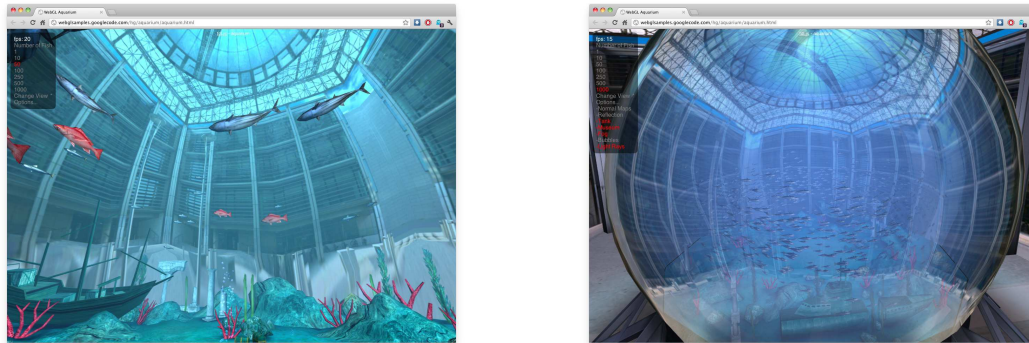


FIGURE 2.10 – WebGL examples.

Since its second version, JavaFX is implemented as a native Java library. Therefore, it allows developers to embed Java packages easily in JavaFX applications. JavaFX also supports FXML as an XML-based description language.

JavaFX provides supports for 2D and 3D graphics, interactivity through java code, a timeline. JavaFX scenes are compiled. JavaFX supports only one video codec, which is the VP6. Concerning audio support, only the MP3 codec is supported. There is no streaming support.

Silverlight

Silverlight is a product distributed by Windows⁷. Its current version is 5. Silverlight targets mobile applications as well as Web applications. It uses the XAML description.

In terms of video and audio, Silverlight supports several codecs such as MPEG-AVC, Theora QuickTime, MP3, etc. Interactivity is supported through .NET languages scripts. Silverlight proposes a support for 2D and 3D graphics. Like JavaFX, Silverlight is compiled.

HTML

The HyperText Markup Language (HTML)[W3C11a] is probably the most famous W3C specification. It is a markup language designed to describe Web pages. Its fifth version is currently under construction and brings a lot of new features to the language such as syntax alleviation, encoding, new nodes. An interesting node of the HTML 5 specification is the *canvas* node. This node enables the support of 2D graphics through a dedicated API. The WebGL specification [Khronos11] is an effort to bring 3D APIs to the HTML 5 *canvas* tag. This specification is not currently supported by all Web browsers. Figure 2.10 shows an animated WebGL scene in the Google Chrome browser.

7. <http://www.microsoft.com/silverlight/>

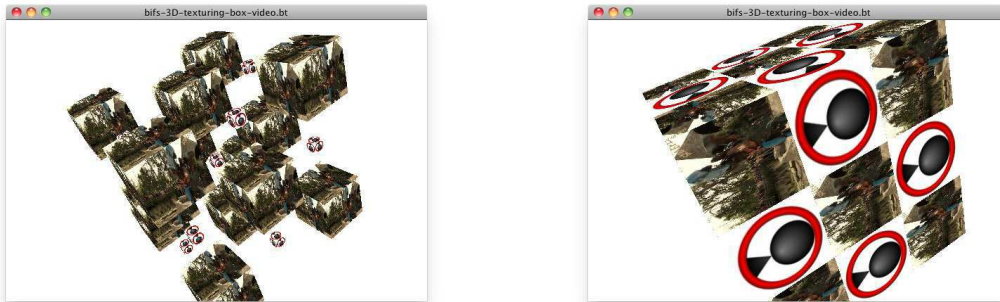


FIGURE 2.11 – BIFS examples.

The *video* tag adds video support to HTML. Currently, no video codec is associated with the *video* element, each Web browsers proposing its own implementation. Interactivity is achieved through JavaScript.

BIFS and XMT

MPEG-4 Binary Format for Scene (MPEG-4 BIFS) [ISO/IEC-1449608] is specified in chapter 8 of the standard ISO/IEC 14496-1 released in 1999 and then revisited as a Part of the standard ISO/IEC 14496-11 in 2005 and in 2008. BIFS is basically an extension of VRML and is a binary compressed format.

BIFS provides the support for 2D, 3D or mixed 2D/3D scenes. As opposed to VRML, BIFS provides tools to change scenes in a dynamic way by replacing, deleting, adding nodes. BIFS proposes an XML version of its language called Extended MPEG-4 Textual format (MPEG-4 XMT). Through the Object Description framework, BIFS is able to manage other kinds of media such as images, videos, etc. Figure 2.11 shows an animated 3D scene displaying a PNG image and an MP4 video in 3D boxes.

LASeR

Lightweight Application Scene Representation (MPEG-4 LASeR) is another scene description language standardized in MPEG-4 [Dufourd05]. It is based on SVG Tiny 1.2 for spatial representation and on SMIL for animations, temporal support and media management. SVG Tiny 1.2 is a sub specification of SVG designed for terminals with weak capacities, mainly with removing items to SVG such as 3D graphics, etc. LASeR also has streaming mechanisms allowing the compression of scenes.

2.1.4.2 Analysis

Scene descriptions represent methods for creating, presenting and sharing interactive and animated multimedia content. The previously presented scene descriptors are not suited for any kind of applications. They are not capable of managing the same

Scene Descriptor	Graphics	Audio and Video	Full Streaming	Compression	Interactivity
VRML	3D	yes	no	no	yes
SMIL	2D	yes	no	no	yes
SVG	2D	no	no	SVGZ version	yes
COLLADA	3D	no	no	no	no
Flash	2D/3D	yes	yes	yes	yes
JavaFX	2D/3D	several codecs	no	bytecode compilation	yes
Silverlight	2D/3D	several codecs	no	compilation	yes
HTML	2D and 3D through WebGL	yes, but not fully standardized	yes	no	yes
BIFS and XMT	2D/3D	yes	yes	yes	yes
LASeR	2D	yes	yes	yes	yes

TABLE 2.1 – Scene Descriptors review

kind of media content in the same way. Table 2.1 summarizes some of the properties of the presented scene descriptors. All the presented scene descriptors propose animation, though this feature does not appear in the table. By full streaming, we mean the capacity to stream all kind of media objects (2D or 3D graphics, audio/video objects, animation, etc.).

Scene descriptors can help developers design remote laboratories since they allow a spatial and temporal representation of scenes, as well as ensuring interoperability between terminals. The choice of a scene descriptor must be carefully analyzed in term of capabilities. As observed from table 2.1, only the MPEG scene descriptions are capable of :

- presenting all kinds of media objects (except for the lack of 3D support for LASeR).
- streaming all kinds of media objects (except for the lack of 3D support for LASeR).
- compressing the scene in order to reduce the size of transmitted data over networks.

2.1.5 Conclusion

Remote Laboratories and Online Engineering bring the possibility to perform manipulations on real devices over the Internet. We believe augmented reality technolo-

gies can help remote experimentation, providing a better understanding of the remote scene. Thus, the augmented reality paradigm and its associated uses have been presented. We have presented collaboration and group awareness and different associated tools in software systems. We have introduced Semantic Web technologies as a key component of Context-Aware System. Multimedia scene descriptors are a way in order to formalize the remote device environment. Thus, they also were introduced and discussed.

2.2 Remote Laboratories and Online Engineering

The previous section discussed technologies and paradigms that can answer the four main issues raised in Chapter 1. In this section, we review and analyze the use of the presented technologies and in Remote Laboratories and Online Engineering. We have noticed that in the scientific literature, augmented reality applications are widely used for industrial usages; online engineering is mostly used for educational purposes. That is why we present a review of scientific contributions divided into two parts. The first part of this analysis focuses on e-Learning. The second part of this analysis focuses on industrial usage and Computer-Assisted Maintenance.

2.2.1 e-Learning

In this subsection, we intend to present the major online engineering projects, especially remote laboratories frameworks. This study is not intended to be exhaustive, but rather presents an overall picture of the online engineering projects presented in the scientific literature.

The **iLab**⁸ project, developed by the Massachusetts Institute of Technology (Cambridge, MA, USA), is probably one of the best known online laboratory framework still in development. The overall objective of this framework is to propose a software architecture for the development of online courses. This architecture is based on the Service Oriented Architecture, SOA. Services are components for accessing online devices, such as authentication, scheduling, experiment storage or authorization. Any kind of client application can be used in order to control the device as the software developer follows the iLab Shared Architecture API⁹ based on SOAP Web Services¹⁰ [Harward04]. In 2012, the first iLab Android client application was presented [Deaky12]. iLab does not provide any tool for collaboration. All the feedbacks from the controlled online devices are designed as software components.

Inside the SOCRATES European programs, the "Minerva action aims to promote European collaboration in the fields of Open and Distance Learning (ODL) and Information and Communication technology (ICT)"¹¹ (2006-2008). This action leads into a noticeable project: **eMerge**. **eMerge** is an educational network structure whose purpose is to disseminate online laboratories experiments, especially in Europe. This project led to the creation of the remote laboratory platform **eLab** in Bordeaux (France) [Lang07]. The client application is a Web application based on the classical Linux Apache MySQL PHP technologies as well as ASP.NET, HTML and SVG for the client side application. A video feedback of the remote instrument can be inserted into the Web application

8. <http://ilab.mit.edu/ServiceBroker/>

9. Application Programming Interface

10. Simple Object Access Protocol - url: <http://www.w3.org/TR/soap/>

11. <http://www.socrates.um.edu.mt/minerva.html>

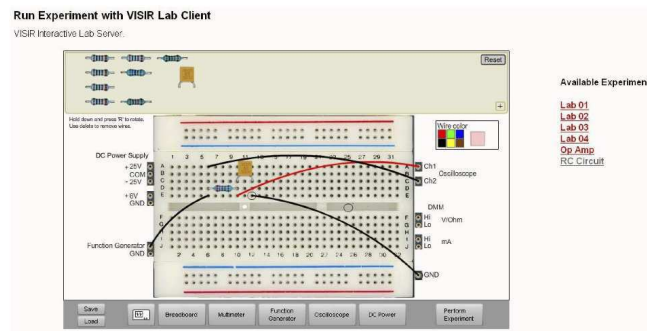


FIGURE 2.12 – VISIR client application.

using a Flash plugin in the browser. There is no mobile application mentioned for this platform.

The **Virtual Instrument Systems In Reality** platform (VISIR) is a platform for the remote control of electronic circuits, developed in Sweden at the Blekinge Institute of Technology [Tawfik12]. The Web interface of VISIR is developed using PHP programming language and Flash technologies. VISIR includes its own Learning Management System (LMS) inside the platform. Figure 2.12 illustrates the client application of the VISIR platform. No video capture of the remote scene is displayed in the client application. No mobile application has been mentioned so far. Some works were conducted in order to use the VISIR client application together with the iLab Shared Architecture [Zutin11]. The VISIR platform does not provide a collaborative control of devices.

The **LabShare** project is an initiative from Australia. The originality of this framework is the hybrid architecture they use in order to interact with the remote devices. A stand-alone application runs on its own virtualized server and the user accesses this application via a remote desktop client. The overall framework is constructed with ASP.NET technologies, PHP, JavaScript and Virtual Network Computing Tools for the client application access. This framework proposes video capture of the remote scene. No support for collaborative works is mentioned nor mobile application [Murray08].

Telelabs¹² is a project from the University of Western Australia. This remote laboratory is intended for students in the mechanical engineering curriculum. It allows students to remotely control different kinds of robots. This project is focused on the will to measure learners' performance and to optimize the organization between tutored time and non-tutored time of the learners while keeping an optimal learning. On the client side, a video capture of the remote robot is delivered to the learner. Some remote experimentations also use augmented reality techniques [Friz98]. The desktop application, called Labs-On-Line, is developed using ActiveX and LabView technologies. There is no mention about a mobile client application [Trevelyan04].

12. <http://telerobot.mech.uwa.edu.au>

The **PEARL** project, Practical Experimentation by Accessible Remote Learning, includes several European institutions from England, Scotland, Ireland, Greece, and Portugal [Cooper02]. This project started in 2000 and ended in 2003. Its first goal was to propose a unique system for accessing remote experiments but they finally divided their solution into two approaches. One is based on LabView, the other is based on CORBA. This project proposes communication tools between learners and also focuses on bringing laboratory accessibility to disabled learners. The client application is based on Java applets and proposes a video feedback of the devices.

WebLab¹³ is a project conducted in Spain [Zubía12]. The particularity of this project is that it offers different possibilities for the client application for accessing the remote laboratories such as virtual world or web applications. For mobile devices, Web applications are used. On the client side, WebLab provides libraries in three technologies : Java Applets, JavaScript and Flash [Orduna11]. Virtual world experiments can be achieved using the Second Life software development kit. In this Second Life world, a pseudo-collaboration is proposed using 3D chat rooms and 3D avatars in the virtual world. A learner can control a micro-robot and "other students can be in the same room in Second Life watching the same results without interacting with the experiment".

Mixed reality techniques for remote experiments can also be integrated in virtual learning environments (VLE). This enables to use both virtual instrumentation (i.e simulation) and real devices. The proposed system in [Pereira09] has been developed as part of the **RExNet** Consortium, financed by the Alfa II European project. A noticeable point of this project is the integration of the Moodle learning management system. The client application of this project has been developed using Java Applets technologies and has been improved with VRML support for 3D environment [Schaf09]. Institutions from Brazil and Germany are involved in the development of this remote experiments framework. A "complete computer-supported collaborative environment" is under development. There is no mention about a mobile application.

The objectives of the **CORELA** project (Collaborative Remote Laboratories), are to propose a "low cost learning environment for collaborative learning" [Muller09] and to spread "mixed-reality environments in which real devices interact with simulation models and 3D Virtual Worlds". This project is mainly intended for mechatronics remote laboratories. A video capture of the remote device is available in the client Web application, developed with PROCESSING¹⁴. PROCESSING is a programming language and a development environment based on Java Applets enabling. The 3D environment is based on OpenSim and integrates PROCESSING. This project uses Moodle in order to reproduce the collaboration and has been influenced by the **MARVEL** project.

The **MARVEL**¹⁵ project is a remote laboratory projects network focused on virtual laboratory in mechatronic. The funding for this project come from the European Com-

13. <https://www.weblab.deusto.es/web/>

14. <http://processing.org/about/>

15. <http://www.informatik.uni-bremen.de/mueller/en/projects/marvel.html>

mission, through the Leonardo Da Vinci program. This project includes several remote laboratories in Europe such as the **Remote Accessible Full-Scale Solar Plant** in Germany (Delmenhorst), the **Remote Accessible Pilot Solar Lab** in Cyprus (Nicosia), the **Robot Training Workshop for Remote Tutoring** in Greece (Athens), the **Remote Workshop for Robotics and Automation** in Scotland (Edinburgh), the **Mixed Reality Web Service for Mechatronics** in Germany (Bremen), the **Remote Workbench for Electronics** in Portugal (Porto) and the **Remote Mechatronics** in Switzerland (Sion) [Bruns05]. The Mixed Reality Web Service for Mechatronics (artecLab) is noticeable since it is the base of the client application of the **RExNet** project.

In Italy, the **Internet Shared Instrumentation Laboratory** (ISILab) proposes remote experiments in digital electronic circuits and digital signal processing board. This project started in 1998. In 2001, it started being used in engineering courses¹⁶. In practical sessions, a "startup source code is provided and students must complete, compile and test it online" [Buschiazzo10]. The feedbacks from the remote controlled devices are presented as software components in the user interface. The HCI is built on top of HTML and AJAX technologies and allows users communication through a video conferencing component. The ISILab is build using SOA and grid infrastructure of online devices [Buschiazzo09]. It provides a plugin for controlling *webcams*.

The **Remote Lab at EPFL** in Swiss (Lausanne) is still under active development and is part of different European projects such as **STELLAR**¹⁷ (2009) or **ROLE**¹⁸ (2009) [Gillet11]. The particularity of the proposed framework is to divide the HCI into reusable components in the form of widgets. Currently those widgets are being built using Java Applets and Flash and a migration to HTML 5 and SVG is being conducted. This would enable to easily build a mobile version of the Web application since mobile browsers are more and more compliant with this specification. Video streaming of the remote devices is achieved using MJPEG. A video conferencing tool is available for students collaborative communications as well as a Laboratory Journal that reports experimental data, graphs or parameters produced by the learners in the **eMersion** cockpit [Salzmann08]. The client application is adapted depending on the available bandwidth.

Another remote laboratory framework was proposed in Australia at the University of South Australia, the **NetLab** project. Like the **eMersion** project, **NetLab** proposes a "collaborative environment for effective interaction between users and the equipment, as well as between users themselves". The user-to-user communication is based on a chat room in the client application [Machotka10]. This application is developed with Java technologies and presents a video feedback of the remote devices. No mobile application is proposed for this framework.

The **DIESEL** project (Distance Internet-Based Embedded System Experimental Laboratory) is funded by the UK Engineering and Physical Sciences Research Council and

16. <http://isilab.dibe.unige.it/how-it-works.aspx>

17. <http://www.stellarnet.eu/>

18. <http://www.role-project.eu/>

is located in Derry (Ulster, Northern Ireland) [Callaghan07]. This project proposes video conferencing tools for collaboration as well as presentation/slide-show, a shared remote desktop, a shared whiteboard and a tele-presence indicator. A video feedback of the remote microprocessor is available in the desktop interface. This interface is developed using ActiveX and Flash technologies.

The **LiLa**¹⁹ project (Library of Labs) is funded by the *eContentplus* program of the European Community. It involves institutions from Germany, Switzerland, UK, Spain, Sweden, Greece and Netherlands. This project started in May 2009 and ended in May 2011. The goal of this project is to aggregate and disseminate remote experiments in Europe and to provide a common architecture for remote experimentations similarly to the **iLab** framework [Bohringer09], defining metadata on collaboratories. The virtual world Wonderland supports the LiLa architecture by providing a framework for collaboration. This virtual world enables constructing virtual "three-dimensional representations of real experiments that are close to look and feel".

Some remote laboratories focus more on augmented techniques for presenting remote laboratory interfaces. This is the case of the work conducted in [Andujar11]. The goal of this work is to control a remote robot. The client application is built using Java Applet. Odeh et al. present in [Odeh09] a remote laboratory with a distributed architecture applied to electrical courses. It is a Web based application which overlays real kits with virtual objects. This application is developed using C# and .NET technologies.

We presented in this section the major remote laboratories for e-Learning. One can notice that a lot of solutions are proposed in order to remotely control devices. Most of the time, the same kind of architecture is presented : a client application that can be a stand-alone application or a Web application, a middleware (in charge of relaying commands) and a device connected to a local computer (for hardware communications). We consider the Java Applets based solutions as stand-alone clients since they require the use of the Java Virtual Machine on the client side in order to run. We also can notice that only the Remote Lab at EPFL project presents context adaptation. The following section will summarize and analyze this literature review.

2.2.2 Industrial Maintenance

We list here the major projects identified in the scientific community for the industrial usage. Again, this list is not intended to be exhaustive but focuses on major projects.

The majority of projects in the area relate to the production and maintenance industry. Production sites and industrial maintenance personnel rely on highly skilled technicians. The latter are responsible for many assembly/disassembly operations, ca-

19. <http://www.lila-project.org/about/index.html>

libration, etc. These human operators are expected to understand and memorize a large panel of procedure steps, in a specific order. Moreover, moving to the production site presents several issues (cost, risk, etc.). As a consequence, a first maintenance operation is usually performed at distance. In order to address those issues (Remote Computer-Assisted Maintenance), several projects have emerged.

The **FP5 STARMATE** project is supported by the Thalès Optronique company. The project has raised the issue of assisted maintenance and training on site. This project allows an operator to bring with him the documentation related to maintenance equipments. It helps establishing training courses for *in-situ* operations of assembly and disassembly of equipment partners. The **STARMATE** project employs augmented reality technologies in order to access the documentation with voice commands and a cursor. The project ended in 2000.

The **ARMAR** (Augmented Reality for Maintenance and Repair) system is used by the US Navy. It uses augmented reality techniques so as to assist the human operator in the realization of a procedural task or repair [Henderson11]. This project requires to wear head-mounted devices and is based on movements detections to provide additional information to the human operator. Additional information can be presented as labels on components, real-time components diagnosis, etc. It allows deporting maintenance operation states in the virtual world by virtualizing the maintenance procedure so that remote operators can monitor and interact with the *in-situ* human operator using audio communication. Pictures of the system **ARMAR** situations are available on the project website²⁰.

The **Chocolate Factory** results from a collaboration within the Silicon Valley between TCHO and the FX Palo Alto Laboratory [Back10]. TCHO is a chocolate products manufacturer located in San Francisco, California (US). The system consists in a virtual world visualization on smartphones or personal computers of manufacturing processes used within the factory. Hundreds of sensors feed the chocolate factory engine 3D virtual world engine, that is based on the 3D graphics engine Unity²¹. The resulting virtual world is used for simulation, visualization and new staff training. It is mostly used in order to know the location of each item involved in the production chain. Although the system was centered on the monitoring and control of production elements, the authors I met in the International Conference on Multimedia & Expo 2011 in Barcelona confirmed that only monitoring is effective.

The **T.A.C** project (Télé Assistance Collaborative) is a French project intended to combine remote collaboration and industrial maintenance [Bottecchia08]. The *in-situ* operator wears head-mounted devices in order to capture the scene. A video stream is sent to a remote expert. The remote expert is responsible for augmenting the scene through a dedicated interface. The additional data are sent back to the *in-situ* opera-

20. <http://graphics.cs.columbia.edu/projects/armar/index.htm>

21. <http://unity3d.com/>

tor who visualizes them through his head-mounted device. Collaboration is achieved using audio-communication tools between the remote and the *in-situ* operator.

The **ARVIKA** project is a German project started in 1999 and ended in 2003. This project is especially intended for the automotive and aerospace industries. The operators also wear head-mounted devices in order to have the augmented view of the world, with contextualized data. Some tracking markers were developed during this project as explained in section 2.1.1.2.a. It seems like this project is able to deliver contextualized data to the operator, according to [Bottecchia08].

The **ARITI** project was initiated in 1996 [Boudoin08]. Early work led to the establishment of the first robot tele-operation system using augmented reality over the Web in France. The **ARITI** project aims at controlling and commanding a remote robot. Unlike the **T.A.C** and **ARMAR** projects, the Human Computer Interface uses a 3D virtualized renderer²². The human operator must make the cognitive effort of synchronizing this world with a video capture of the remote scene. Part of the results of this project constituted preliminary results for the **ANR RIAM Digital Ocean**. A particularity of this work is that it allows a collaborative remote operation on the robots using a multi-agent system.

The **ANR RNTL AMRA** (2001-2004) was intended to implement an augmented reality system running on mobile devices for industrial use, especially in the industrial maintenance field. More specifically, the project focused on the training phase of maintenance operations [Didier05]. The originality of this project is the study of contextual helps of inexperienced operators and the bringing-in of relevant information to maintenance staff on their workstation (maintenance documentations, operation processes, assembly video, etc.), the pillar of the project being augmented reality techniques in order to increase the availability of information on the site maintenance.

In Italy, **VRMedia**²³, a spin off from Scuola Superiore Sant'Anna de Pise, markets **REAL-AR**. It is an augmented reality application focusing on learning. The product provides real-time tutoring on devices maintenance. Tutors can send information to learners who receive them on a portable computer. Italians from Pisa are also involved in the **FP7 FLEXA** project (2008-2012). Even if the main objective of this project is to develop robotics for motor production in the aeronautic industry, they intend to use this product for *in-situ* maintenance at the University of Pisa. The remote dimension is not taken into account.

In the **ANR CONTINT** program, we can mention the **ANR RAXENV** project (2007-2010). The aim of this project is to demonstrate the feasibility of an outdoor augmented reality system in environment area, in terms of technology but also of adoption by end users. It focuses on reality enrichment with contextualized information using geographical sensors. The system provides no interaction since it was not the objective of the

22. <http://ariti.ibisc.univ-evry.fr/doku.php?id=manipulation>

23. http://www.mindthebridge.org/wp-content/themes/d_natives/VRmedia.html

project.

Other projects are ongoing concerning teleoperation on robots or devices using augmented reality techniques, especially the **FP7 Digital Ocean 2** (2011-2012). It is the follow-up of the **FP7 Digital Ocean** (2006-2009). It concerns the whole chain of digital content of the seabed from the creation, the production, the edition, the interfaces, serious gaming screenwriting to its distribution on dedicated terminals or mobile devices. It will allow to geo-localize and contextualize multimedia contents thanks to submarine robots.

Another solution to address remote maintenance operations consists in using remote desktop sharing tools such as **Virtual Network Computing** (VNC). This is often a last resort for basic maintenance. Using VNCs entails installing the entire device interface on a system able to receive such configuration. This kind of system does not have genuine collaborative working arrangements. A unique cursor also promotes the "scrollwars" [Lukosch01]. Any communication protocol is available and the interface cannot be contextualized.

This review of augmented reality systems used in the industry area shows that most of the proposed solutions are not conducted online. Few of them propose a collaborative control of the remote device and a lot of them require wearing devices such as head mounted devices. Also, few of them also propose contextualized data to end users.

2.3 Synthesis

2.3.1 Analysis

In this section we analyze the solutions presented in the scientific literature for controlling devices. The proposed analysis focuses on four main points : the distant control of devices, the augmented reality paradigm, multimedia scene descriptions and collaboration. A lot of frameworks for the remote control of devices are proposed in the literature. It is a difficult task to review all of them, knowing that a lot of them are reused in a variety of different projects. Table 2.3 and table 2.2 draw a synthetic view of our analysis.

In the e-Learning area, we can make the same kind of remarks. While some of remote laboratories propose augmented reality solutions, few of them propose a collaborative control of the remote device. Again, most of the time a video feedback is delivered to the remote learner. The video feedback allows the visualization of command results on the remote device, such as a new plot displayed by the device. But it does not allow to know which command was sent to achieve this. The consequence is the inability to transcribe the "group awareness", i.e. to make the user feel that he is in a group of people connected and to enable him to determine the author of a command, as well as to communicate with the people of his group.

In the industrial maintenance area, we can notice that very few of the analyzed solutions propose remote operations. The *in-situ* operator often wears a head-mounted device in order to have the "augmented information". The remote expert is either confined to an advisory role through an integrated audio system or interacts with the remote scene via a virtualized model of the remote scene, which is not the most immersive solution. At best, a captured video is provided and the remote expert, who is a stranger to the local reality, has to close the gap between the simulated model and the video feedback control. In this case, the *in-situ* operator perceives only the results of the actions made by the remote expert.

Very few of the presented works are using multimedia scene descriptions, especially those with augmented reality techniques. As presented in the previous sections, multimedia scene descriptions rely on standards. As a consequence, they ease interoperability between terminals. At the pervasive computing area, remote laboratories must be controlled by the highest number of terminals. From our point of view, interoperability is an utmost issue for remote laboratories. We can also notice that very few systems take into account some elements of the user or system context.

Projects	Local or Distant	Augmented Reality	Open Standard	Collaboration	Context Awareness
iLab	Distant	no	no	no	no
eLab	Distant	no (video feedback)	HTML, SVG, Flash	yes	no
VISIR	Distant	no	HTML, Flash	no	no
LabShare	no	no (video feedback)	no	no	no
Telelabs	Distant	yes	no (ActiveX)	no	no
PEARL	Distant	no (video feedback)	no (Java Applets)	yes	no
WebLab	Distant	yes	Flash (3D environment)	yes	no
RExNet	Distant	no	VRML	under development	no
CORELA	Distant	yes	no (OpenSim)	yes (through Moodle)	no
ISILab	Distant	no	HTML	no	no
Remote Lab at EPFL	Distant	no (video feedback)	ongoing works for HTML 5 and SVG	yes	yes
NetLab	Distant	no	no	yes	no
DIESEL	Distant	no (video feedback)	Flash	yes	no
LiLa	Distant	no	no (Wonderland)	yes	no
[Andujar11]	Distant	yes	no (Java Applet)	no	no
[Odeh09]	Distant	yes	no	no	no

TABLE 2.2 – Research project in e-Learning

Projects	Local or Distant	Augmented Reality	Open Standard	Collaboration	Context Awareness
FP5 STARMATE	Local	Head-Mounted Device	no	no	no
ARMAR	Local	Head-Mounted Device	no	yes	no
Chocolate Factory	Distant	yes	no (Unity)	yes	no
T.A.C	Local	Head-Mounted Device	no	yes	no
ARVIKA	Local	Head-Mounted Device	no	no	yes
ARITI	Distant	3D virtualized render	no (3DVIA & Java)	yes	no
AMRA	Local	yes	X3D/VRML	yes	yes
VRMedia	Local	yes	no	yes	no
Digital Ocean 1&2	Distant	yes	no	yes	yes
VNC	no	yes	no	no	no

TABLE 2.3 – Research project in industrial maintenance

2.3.2 Towards Collaborative and Contextualized Online Instrumentation

Our study on remote laboratories in e-Learning shows that the solutions presented do not try to reproduce enough the interactions between learner and machine. Some solutions are using video feedbacks in order to reproduce the reality of the remote scene. But from our point of view, it does not offer a high fidelity of the representation of the apprehended scene. Remote laboratories should go towards augmented reality in order to give a better representation of the remote scene. In the industrial maintenance area, augmented reality is widely used but very few solutions enable the remote control of devices. We can also notice that very few of the solutions proposed in e-Learning are used in industry, even if the need for remote control is the same.

Interoperability is an utmost issue in remote laboratories. When building online engineering solutions, we need to take into account the possibility to control remote devices with any kind of terminal. Multimedia scene descriptions contribute to interoperability by providing standards to rely on when designing multimedia applications. The study on multimedia scene descriptions shows that not all of them have the same capabilities. It shows that MPEG-4 BIFS is the only scene descriptor answering the problems of streaming, compression and media presentation.

Collaboration generally refers to individuals or organizations working together to address problems and bring results that are not easily achieved by working alone. Collaboration entails the sharing of the same resource at the same time. Collaboration is a key component in remote tele-operation, especially in the field of engineering learning. In learning theories, socio-constructivism [Jonassen91] is the theory that tries to make students learn from their own observations and from the interactions with others, both teachers and other students. So far, a remote laboratory solution should manage the "group awareness" [Gutwin04] mechanism, so that collaboration that occurs in traditional local sessions can be reproduced online. To ensure collaboration, a high level of trust has to be achieved and a frequent communication has to take place. Thus, collaborative systems should not only allow the sharing of a resource between the different protagonists, but it should also maintain group awareness. As stated in [Gutwin04], "Group awareness is the understanding of who is working with you, what they are doing, and how your own actions interact with theirs". Our study shows that few of the proposed systems enable a collaborative control of the remote device.

In this thesis, we try to answer the following questions :

- How to propose a remote control of devices that can be used in industry as well as in e-Learning ?
- How to propose a fully interoperable augmented reality solution ?
- How to propose a RIA that limits the bandwidth ?
- How to handle collaboration in augmented reality applications ?

- How to propose a solution as much reusable as possible in order to decrease the development time of remote laboratory construction ?
- How to propose a system that takes into account the user procedure, as an element of context ? How to adapt the user interface depending on this element of context ?

When in doubt, use brute force.

Butler Lampson

3

Proposal

▷ *Chapter 1 raises the four main issues of online engineering we try to answer in this thesis. These issues are the remote control of devices through augmented reality views, the collaborative control of devices, the adaptation of the interface depending on the context of use and the scalability in terms of different controlled devices. Chapter 2 presents the state of the art approaches that can overcome these issues. In this chapter, we present our contribution in the online engineering field, serialized in form of a framework.* ◁

Chapter Plan

3.1	Online Engineering and Computer Aided Maintenance Framework	59
3.1.1	Architecture	59
3.1.2	Software Chain	61
3.1.3	System at runtime	63
3.1.4	Conclusion	63
3.2	Augmented Reality	65
3.2.1	Scene Analysis	66
3.2.2	Device Identification and Pose Estimation	67
3.2.3	Conclusion	72
3.3	Handling Collaboration	73
3.3.1	Middleware	74
3.3.2	Collaborative Interactive Multimedia Interface	76
3.4	Adaptation of the Human Computer Interface	77
3.4.1	Whisper Module	77
3.4.2	Expressing Sequences.	78
3.4.3	Use Trace	79
3.5	Multimedia Delivery	81
3.5.1	Multimedia Composition using MPEG-4 BIFS	81
3.5.2	Multimedia Delivery	83
3.6	Conclusion	88

CHAPTER 1 raises the four main issues of online engineering we try to answer in this thesis. These issues are the remote control of devices through augmented reality views, the collaborative control of devices, the adaptation of the interface depending on the context of use and the scalability in terms of different controlled devices. Chapter 2 presents the state of the art approaches that can overcome these issues.

Augmented reality solutions provide a better understanding of apprehended scenes. We have observed that in the industrial maintenance area, few of the proposed systems enable a remote control of the device. In the e-Learning area, a lot of systems to control the remote devices are available, but few of them grant for an augmented view of the scene. In both fields, very few systems propose a collaborative control of devices, nor an adaptation of the interface depending on the context. Finally, most of the time, reusability is not often taken into account.

In this chapter, we present a framework for building online experimentations, both for industrial maintenance and e-Learning areas. This framework uses augmented reality to present the interface to the remote users. It enables a collaborative control of the devices and creates a group awareness in order to reproduce "the sense of being there". The framework also proposes an adaptation of the interface depending on the users' activity. Finally, it eases re-usability, providing the different tools to set up a wide range of devices.

Section 3.1 presents this framework and its main architecture. Section 3.2 presents how augmented reality views are created. Section 3.3 explains how collaboration is handled, both in the middleware and at the client side. Section 3.4 introduces the adaptation of the multimedia HCI, based on user's activity. Section 3.5 presents how the multimedia content is delivered to the end users.

3.1 Online Engineering and Computer Aided Maintenance Framework

3.1.1 Architecture

As previously explained, figure 1.2 presents the basic architecture of online experimentations. We can notice that the Human Computer Interface is closely bound to the device to control, since a specific interface is designed for a specific device. The same observation can be made for the driver implementation (the link between the local computer and the device to control). The architecture used nowadays implies that for a new device to be remotely controlled, a new instance of the schema has to be created. Moreover, the middleware in those situations is just a message conveyer between both computers.

In this thesis, we try to overcome those issues. Then, we propose the architecture illustrated in figure 3.1. Moving intelligence on the middleware side allows to add an

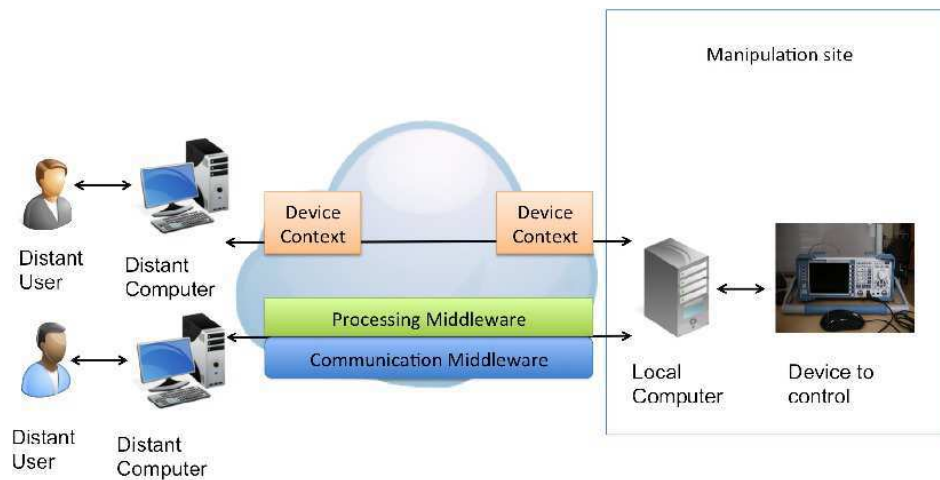


FIGURE 3.1 – Remote experimentation architecture.

abstraction layer to the drivers. The middleware is then not just a messages conveyer, it also allows to process commands. This feature is important for device scalability, i.e. the possibility to support a wide range of different instruments without creating a new instance of the system when running new experimentations. The communication middleware is designed in order to support collaboration. The **Device Context** component permits to formalize the description of the remote device. Firstly, it allows to design the augmented reality interface delivered to the end users. Secondly, it delivers the adaptation of the interface, depending on the context of use.

In this section, we present the main contributions of our work in the online engineering field. Those contributions are implanted in an overall framework for building and running online experimentations in the e-learning and in the remote maintenance areas.

The framework proposes two operating modes. The first mode is the **runtime** mode. This mode is effective when running online experimentations. It delivers an **augmented view** of the **remote** device. It allows a **collaborative control** of the remote scene. Users can interact in the same time with the remote scene. Since group awareness is at the center of collaboration, we propose a mechanism and tools to transcribe the "sense of being there" [Gravier11]. In order to achieve these goals, some improvements are proposed in the BIFS standard. The runtime mode also **adapts** HCIs of online engineering client applications based on the user's activity. When a widget of the interface is activated, the system proposes to the operator the next possible actions. The aim is to limit the number of human errors, as well as to increase the efficiency (time spent, quality, etc.) of the process.

The second mode concerns the **set-up** of remote experiments. It is basically composed of authoring tools that enables the creation and the deployment of new online

experimentations. It creates the **remote scene description**, based on multimedia scene descriptions and enables business rules experts to express sequences of actions in HCIs.

3.1.2 Software Chain

In order to propose a framework for building remote experiments using augmented reality techniques, we identify some factorable elements of the software chain, presented in section 1.2.2 and illustrated in figure 3.1. In this section, we study the possibility to factor the links of the software chain.

3.1.2.1 Devices drivers and Processing Middleware

The link between the local computer and the device entails the mastery of standards and of access protocols. This is complex due to a) the heterogeneity and the number of technologies and b) the number of material interfaces allowing interfaces with devices : PXI, USB, GPIB, etc.

As presented in section 1.2.2, some efforts were made in order to propose common drivers through standards. Those drivers define *classes* of devices, in order to ease interoperability between devices and to decrease the development time and cost. The **Virtual Instrument Software Architecture**¹, also known as **VISA**, is an Application Programming Interface (API) widely spread in the instrumentation area. This standard was developed following the **Standard Instrument Control Library** from Agilent Technologies. VISA was the starting point of the **IVI** standard.

The **Interchangeable Virtual Instrument** standards² (IVI), define instrument classes such as Oscilloscope, Spectrum analyzer, Switch, etc., in order to provide interchangeability between instruments of different constructors. The hardware functions, once designed for a class of instruments, are fully compatible with other instruments of the same class, assuming they are IVI compliant. Those drivers also provide simulation modes, enabling to simulate the device when it is not accessible. Unfortunately, those standards suffer from one limitation added to the previously announced ones [Landragin06] :

- "The additional software that entails the adoption of the IVI standard is likely to slow down the system speed".

We can add that a lot of online engineering solutions are from "home made" developments. We cannot guarantee the standards are respected. Yet, this represents a non-negligible number of remote laboratories in the reality.

A complete generic architecture for devices would be very difficult to achieve. Almost systematically, a dedicated software development is needed in order to perform

1. <http://www.ivifoundation.org/specifications/default.aspx>

2. <http://www.ivifoundation.org>

this link. The *proof of concept* developed in this thesis uses a spectrum analyzer FSL 3 from Rhode & Schwarz, which is fully IVI-compliant and was lent by the company in order to conduct our experiments. The IVI foundation proposes an implementation of the IVI drivers through Windows libraries. Although limited, the IVI standards seem the best compromise for the development of our *proof of concept*.

3.1.2.2 Message Middleware

We now focus on the middleware. The middleware is in charge of conveying the commands from the remote user to the device but also of giving a behavior of the collaborative platform, i.e. conveying the messages from the devices to all the users. Two different approaches exist in order to realize a middleware : *Object-Oriented Middleware* (OOM) and *Message-Oriented Middleware* (MOM). The difference between those approaches relies on the remote service access methods. OOMs instantiate a local representation of the remote object and use it as the remote object. On the contrary, MOMs do not directly expose access methods to remote objects but rely on a messaging mechanism in order to exchange messages containing operations to execute.

As Bernstein [Bernstein96] proposes, a middleware (OOM or MOM), can be perceived as a layer between the application layer and the operating systems, due to its conveying mission. The middleware is then an essential piece in all distributed applications. Our framework for remote operation is necessarily a distributed architecture, since remote users are not connected using the same terminal. We need tools and mechanisms to ensure the messages' integrity.

From our point of view, a middleware, as a basis of collaborative architectures, must have the following features :

- Asynchronism. A synchronous middleware is characterized by a blocking call for the message reception method. As no response is received yet, the program process is waiting. An asynchronous middleware allows to keep control on the application even if the message is not yet received.
- Robustness and scalability. The remote control of devices does not limit the number of users connected to the experiments.
- Scheduling warranty. The middleware needs to ensure that messages are received in the order they are emitted. This is a compulsory feature of online engineering middlewares, since the order of commands on the devices is crucial for the good behavior of experiments, even for the integrity of the system.

3.1.2.3 Human Computer Interface

We saw in the previous chapters that the Human Computer Interface used in online engineering and industrial maintenance are not enough interoperable and especially not for augmented reality applications.

In order to propose a generic HCI for online engineering we propose the following features :

- A high fidelity of the representation of the device environment.
- Interactivity as close as possible to the device interactivity.
- Connectivity to the middleware for sending and receiving commands/results as well as collaborative messages (users actions, text messages, audio, etc.)
- A rich graphics and widgets panel (possibility to embed 2D/3D graphics, images, animations, etc.).
- A high level of interoperability, enabling users to remotely control devices from any kind of terminals.
- A loose coupling between devices and the architecture, in order to be as reusable as possible (independence to device).

As presented in section 2.1.4.2, the MPEG-4 BIFS multimedia scene description is the only one capable of :

- presenting all kinds of media objects (2D, 3D, audio, video, etc.).
- streaming all kinds of media objects.
- compressing the scene in order to reduce the size of transmitted data over networks.

Therefore, the description of the remote scene delivered to the remote user in our framework is based on MPEG-4 BIFS.

3.1.3 System at runtime

The overall architecture of the system at runtime we propose is illustrated in figure 3.2. An IP camera is capturing the remote scene and sending an MPEG-4 AVC stream to the middleware, hosted on an application server. The middleware computes and delivers the interactive multimedia to the end user. Users send commands to the middleware through the multimedia HCI. Depending on the whisper, the stream is adapted and the scene is updated on the users' HCI. The interpreter is responsible for sending the corresponding hardware fonctions.

3.1.4 Conclusion

In this section, we have defined the factorable elements of the software chain. This chain can be divided into three main components : the driver instruments (the processing middleware), the message middleware and the HCI. Driver instruments are responsible for sending hardware messages to the device, corresponding to the messages sent from the HCI. The message middleware is responsible for conveying messages between users and the device and should allow a collaborative control of the device. The HCI is the link allowing remote users to interact with devices. The HCI should be accessible from any kind of terminal and should use mixed reality techniques in order to propose a better understanding of the remote apprehended scene.

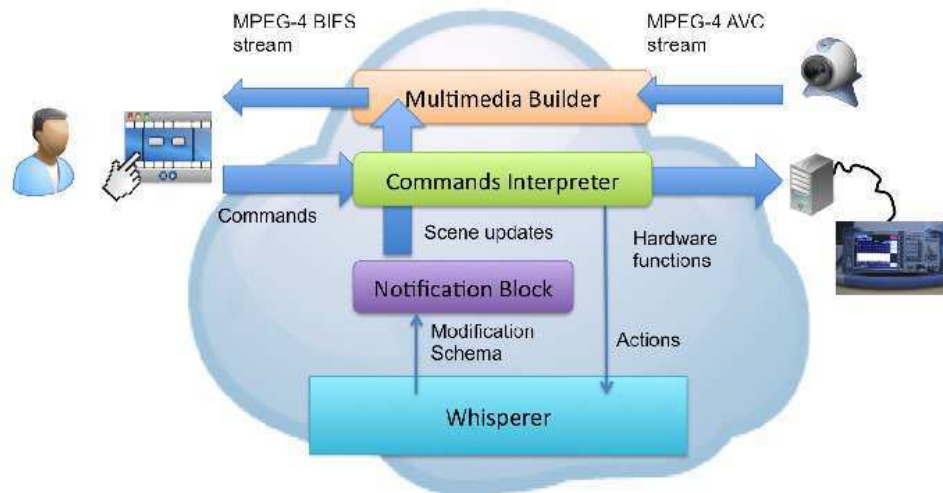


FIGURE 3.2 – Overall Architecture at runtime.

In the set up mode, the different authoring tools enable to create :

- the multimedia scene description of the remote scene. The Multimedia Builder uses this description in order to create the multimedia content delivered to the remote user.
- the drivers of the device. The Commands Interpreter is then capable of sending the hardware functions to the remote device.
- the description of the usage procedures. These descriptions enable the Whisperer to formalize the adaptation of the interface.

The set-up mode is illustrated by figure 3.3

We can notice that this division enables to support the four main contributions we try to answer in this thesis. The Multimedia Builder (thanks to the set-up of the system) constructs the multimedia scene descriptions of the **remote** scene using **augmented reality** techniques. The Commands Interpreter and the Notification Block enable the **collaborative control** of the device. The Whisperer module is the engine responsible for the **adaptation** of the interface. The combination of the Commands Interpreter and the Multimedia Builder enable to create a **scalable system able to control a wide range of devices**.

This section has introduced the overall architecture of the system proposed in this thesis work. The next section presents the multimedia scene reconstruction as well as the multimedia delivery chain to the end users.

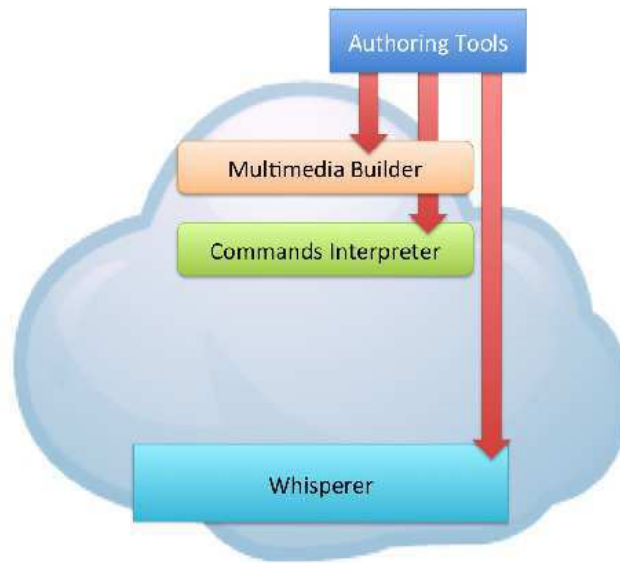


FIGURE 3.3 – Overall Architecture at set-up.

3.2 Augmented Reality

Augmented reality is the combination of real and virtual objects in a common environment. As presented in previous chapters, three fundamental rules must be respected in order to build an effective augmented reality application [Swan05] :

- Combine the real world and the virtual world.
- Take into account the consistency and the coherence between the real and the virtual worlds.
- Handle the constraints of real-time and interactivity.

This entails computing a proper environment (a 3D environment) as well as disposing reference objects in order to satisfy the first two constraints. Of course, when building remote laboratories, the device itself will be the reference object.

In this section, we present how we reconstruct the instrument feedback, seen as a natural video for the end-user and integrated in the multimedia interface. After analyzing the environment of the remote device, we present the algorithms employed to determine the device identification and its pose estimation in the scene.

The multimedia delivery chain is illustrated in figure 3.4 and serves as the reference in the following subsections.

The IP camera delivers encoded video. The processing block identifies the instrument present in the video stream by comparison with available devices in the database and computes the position of this instrument in the images. The instruments storage in the database is based on the instrument's image features. The processing block also

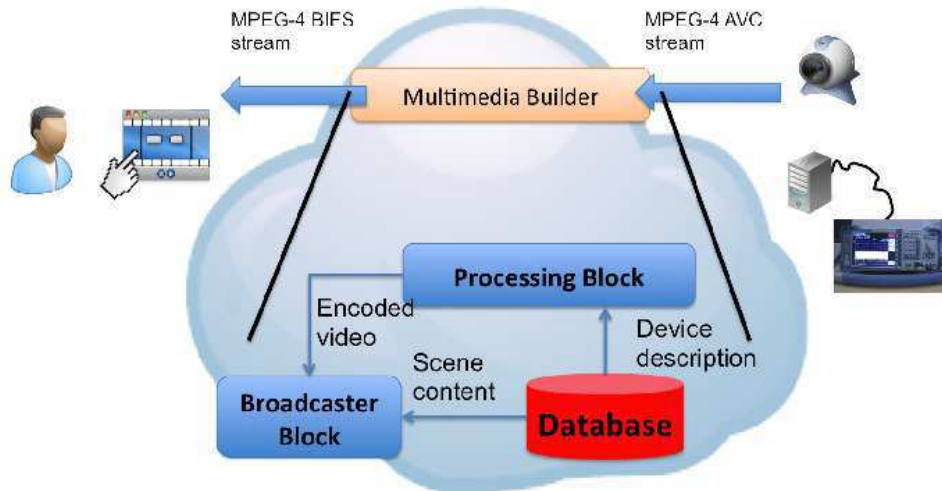


FIGURE 3.4 – Multimedia Chain Delivery.

adapts the video by cropping the region of interest, which in general corresponds to the device's screen. Let us note that the image processing operations in the processing block are effectuated in the non-compressed domain, therefore a video decoder and encoder are integrated. Once the instrument is recognized, the corresponding graphical interface is streamed using the broadcaster block.

3.2.1 Scene Analysis

Figure 3.5 represents the spectrum analyzer we have used in order to develop our *proof of concept*. In order to construct the multimedia HCI of the instrument, the different kinds of existing regions in the scene need to be identified. We encountered 3 types of regions for our use case :

- The video regions where objects are likely to change during time, corresponding to the feedback from the instrument (the display of the instrument in figure 3.5 for instance).
- The interactive regions enabling functionalities of the instrument, (the buttons in figure 3.5 for instance).
- The background (static unchanged pixels elements over the session).

In order to limit the bandwidth when transmitting the multimedia, only the video region is transmitted. For this kind of devices, only the display is a relevant video region. All the others regions are computed using graphics or static pixels. In order to improve the interoperability between terminals we choose multimedia scene descriptors to present the augmented scene. Based on our review on multimedia scene descriptions, MPEG-4 BIFS is the better compromise in order to deliver remote mixed reality

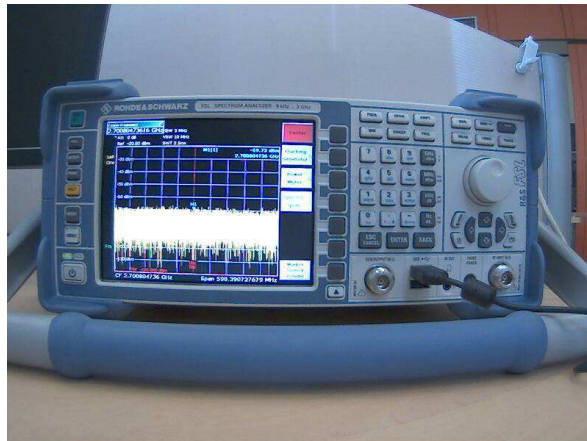


FIGURE 3.5 – A Spectrum Analyzer.

applications.

3.2.2 Device Identification and Pose Estimation

The processing block presented in figure 3.4 decodes the compressed video delivered by the IP camera. It then identifies the device in the video stream and computes its pose estimation. The video feedback is then extracted and encoded, broadcasted to the remote users. In this subsection, we present how we compute the identification and the pose estimation of the device in the video stream. The first section presents the general principle. The second paragraph presents the camera calibration, a must phase in every image processing phase. The third part explains how we determine homographies that help computing the 3D environment needed for mixed reality.

3.2.2.1 General Principle

To clarify why we need to recognize the instrument and to calculate its position in the video stream, let us consider the scenario of Steve, Heather and Jimmy, defined in section 1.2.4.

Steve streams the video of the instrument to our system with his smartphone. The system identifies the instrument and delivers the proper interface to Heather and Jimmy. Since Steve's hand is shaking a little, the video is not stable, so the system keeps calculating the instrument positions in order to crop the good region of the stream to broadcast it to the students. They can then realize the remote experimentation with a stable video feedback from the instrument by activating widgets in the multimedia interface.



FIGURE 3.6 – Display extraction.

While the marker-dependent methods offer important robustness and efficiency in computational terms as compared to the marker-free methods, the latter does not compel *a priori* intervention in the scene. Such markers would also entail a very high calibration phase because the system would have to know all the elements' relative positions and orientations in the scene from the marker.

All the devices from the database have associated descriptors. When the camera captures a device, the descriptors are extracted and compared with those in database. Due to the computation time and the real-time constraint, the descriptor is computed for one image out of two.

Once the device is identified, local features are used in order to calculate the pose estimation. The local feature points detected are matched with the local feature points of the reference image stored in the database using the Fast Library for Approximate Nearest Neighbor [Muja09]. It is then possible to estimate the position of the instrument in the video stream thanks to the Zhang algorithm [Zhang00],[Zhang10]. Knowing the position of the device in the video stream allows us to crop the feedbacks of the instrument, encode and stream them to the final user with the whole interactive multimedia content. In the spectrum analyzer scene, the feedback area is represented by the display of the device, as illustrated by figure 3.6. This allows limiting the bandwidth, since only the necessary area is a streamed video. The rest of the multimedia scene is construc-

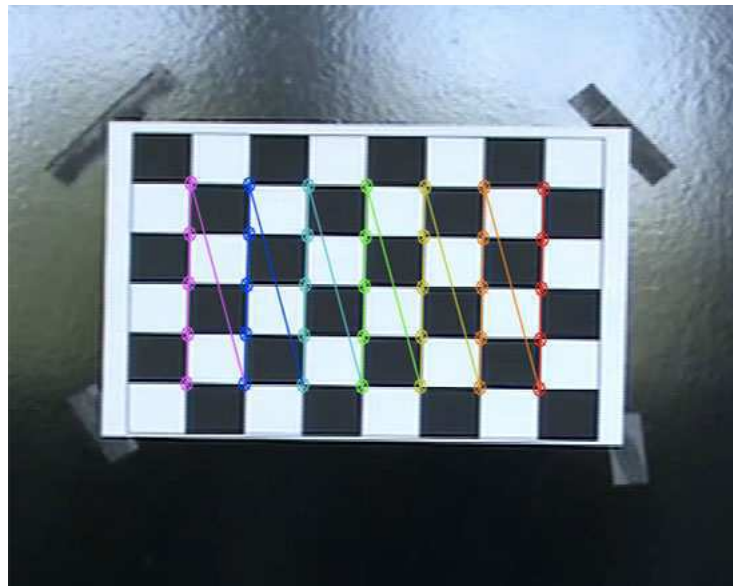


FIGURE 3.7 – Checkerboard with detected corners.

ted using graphics. We highlight the fact that the device is identified frequently. The importance of this feature is discussed in section 3.5.2.3.

In Chapter 5, a test benchmark of local features and their usage for remote laboratories are presented.

3.2.2.2 Camera Calibration

Before being able to solve the pose estimation of the device in the scene, we need to determine camera parameters. These parameters are presented as a matrix, commonly known as the **camera matrix**. It is a 3×4 matrix that describes the mapping of pinhole camera from 3D points in the real world to 2D points in a image.

In order to compute the camera matrix, the algorithm we describe here needs the coordinates of known points of a scene taken from different point of view. Zhang recommends using multiple images of a model whose characteristics we know. A checkerboard seems the best choice, since we know the size of squares.

In order to detect squares corners, we based our realization on the Vezhnevets algorithm [Vezhnevets06]. This algorithm first computes a binary version of the image using an adaptive threshold. Black objects squares are extracted from this binary image and their coordinates are extracted. Figure 3.7 is an image of the taken checkerboard and its detected squares. The Zhang algorithm allow computing the camera matrix using homographies and detected points. Let M_{int} be the camera matrix.

Let B and b matrixes such as :

$$B = M_{int}^{-t} \cdot M_{int}^{-1}$$

$$b = (B_{11} \ B_{12} \ B_{22} \ B_{13} \ B_{23} \ B_{33})^t$$

Let h_i be the matrix representing a homography. We have :

$$h_i = (h_{i1} \ h_{i2} \ h_{i3})^t$$

then :

$$h_i^t \cdot B \cdot h_j = v_{ij}^t \cdot b$$

with :

$$v_{ij} = (h_{i1}h_{j1} \ h_{i1}h_{j2} + h_{i2}h_{j1} \ h_{i2}h_{j2} \ h_{i3}h_{j1} + h_{i1}h_{j3} \ h_{i3}h_{j2} + h_{i2}h_{j3} \ h_{i3}h_{j3})^t$$

if n images are captured, then the last equation can be shorten as :

$$\mathbf{V} \cdot b = \mathbf{0}$$

\mathbf{V} is a $2n \times 6$ dimension matrix. The solution of this last equation is given by the eigen vector associated with the smallest eigen value of the $\mathbf{V}^t \cdot \mathbf{V}$ vector. When b is estimated, all the intern matrixes can be determined. We finally obtain the camera matrix estimation using a minimization of the previously determined matrixes.

3.2.2.3 Homography Between Planar Model and its Image

An **homography** is a transformation that allows the correspondence of points from a plan to another, as illustrated by figure 3.8. It is represented by a 3×3 matrix. It has 8 degrees of liberty (one component of the matrix is only an arbitrary scale factor). We then need four pairs of corresponding points to determine an homography.

The homography is determined using the **RANdom SAMple Consensus**, RANSAC [Fischler81]. RANSAC is an iterative methods that helps classifying items of a statistical population. We choose 4 pairs of points determined by the *FLANN*, Fast Approximate Nearest Neighbor, algorithm. FLANN determines corresponding points in high dimension vectors using a hierarchical tree. When the 4 points are determined, we compute the homography using them. We determine a distance d for each correspondence. This operation is repeated until obtaining the best homography.

The device pose estimation is an important step, since it allows to determine the position of the video feedback streamed to the remote user. Let $P_{3D,i,i=1..4}$ be points of

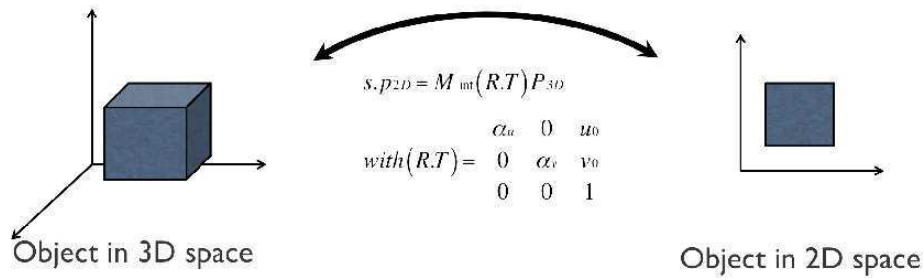


FIGURE 3.8 – Homography principle.

the device in a reference image in homogenous coordinates and $p_{2D,i,i=1..4}$ the corresponding points in images captured by the camera in homogenous coordinates. Thanks to the Zhang algorithm that is robust and simple to implement, for each $P_{3D,i,i=1..4}$ of the reference image, we can write :

$$s \cdot p_{2D,i,i=1..4} = M_{int} \cdot M_{ext} \cdot P_{3D,i,i=1..4}$$

with s being a scale factor.

M_{int} and M_{ext} represent respectively the camera matrix and the extrinsic matrix of the system. M_{ext} can be represented by : $M_{ext} = (R, T)$ with $R = (r_1 \ r_2 \ r_3)$

R and T are the rotation and translation matrixes mapping the object landmark (points coordinates in the reference image) to the camera landmark (points coordinates captured by the camera). M_{int} is defined as follows :

$$M_{int} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

$(u_0 \ v_0)$ being the coordinates of the principal point, i.e. the center of the image in the camera landmark. α_u and α_v are scale factors following the u and v axes of the image.

We have constructed our prototype using a spectrum analyzer. This device has a 2D font. When representing its coordinates in a mathematical model, we can easily write that their z -coordinates are null. Written in the homogenous coordinates, we can write :

$$\begin{pmatrix} s \cdot u \\ s \cdot v \\ s \end{pmatrix} = M_{int} \cdot (r_1 \ r_2 \ r_3 \ T) \cdot \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = M_{int} \cdot (r_1 \ r_2 \ T) \cdot \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

A point $P_{3D,i}$ is related to its image $p_{2D,i}$ by an homography H defined with a scale factor by :

$$s \cdot p_{2D,i} = H \cdot P_{3D,i}$$

with :

$$H = \lambda \cdot M_{int} \cdot (r_1 \ r_2 \ T)$$

The homography being a 3×3 dimension matrix, we can write that :

$$(h_1 \ h_2 \ h_3) = \lambda \cdot M_{int} (r_1 \ r_2 \ T)$$

This last equation can be solved with an analytical method, knowing all the parameters of the system.

$$\begin{aligned} r_1 &= \lambda \cdot M_{int}^{-1} \cdot h_1 \\ r_2 &= \lambda \cdot M_{int}^{-1} \cdot h_2 \\ r_3 &= r_1 \cdot r_2 \\ t &= \lambda \cdot M_{int}^{-1} \cdot h_3 \end{aligned}$$

with $\lambda = 1/N(M_{int}^{-1} \cdot h_1) = 1/N(M_{int}^{-1} \cdot h_2)$
where $N(X)$ is the norm of the X vector.

We can then determine points coordinates in the scene of pre-defined points in the image reference. For the spectrum analyzer, only the display is the feedback of the device. This method is used to determine the position of the display and crop it in the video stream. Only the video feedback as natural video is broadcasted to the remote user, as presented by figure 3.6.

3.2.3 Conclusion

In order to create an augmented view of the remote scene, we need to create a computed 3D environment of the scene. We identify 3 areas in the remote scene, a feedback area, a background and an interactive area (encompassing interactive elements).

The device is first identified in the video streamed delivered by the IP camera using image features and descriptors. Then, an homography between the 3D environment and a reference image is computed in order to determine the pose estimation of the device. Homographies are determined thanks to Zhang algorithms.

3.3 Handling Collaboration

Collaboration generally refers to individuals or organizations working together to address problems and bring results that are not easily achieved by working alone. Collaboration entails the sharing of the same resource at the same time.

To ensure collaboration, a high level of trust must be achieved and frequent communication has to take place. Thus, collaborative systems should not only allow the sharing of a resource between the different protagonists, but it should also maintain group awareness. As said in [Gutwin04], "Group awareness is the understanding of who is working with you, what they are doing, and how your own actions interact with theirs".

In remote control systems, teachers are concerned by group awareness since it eases the management of student's achievement. To manage collaboration and thus group awareness in remote laboratories, different approaches are used. In [Tuddenham10], Tuddenham et al. present a shared visual workspace for tabletop. Their system reproduces protagonists' arms when working on the remote tabletop and speakerphones are used to ease communication between the users.

Pereira et al. in [Pereira09] propose a "learning-material module". Depending on the logged user, different kind of operations are allowed. Different "learning modes" are suggested by the system, such as "active learning" or "team learning".

Communication tools, such as video, audio, chat and shared workspaces are not the only key issue when designing collaborative remote laboratories, especially in distance learning curriculum, as stated in [Lowe09] : to support supervision, they decided to display multiple devices at the same time, like "a tutor being able to walk around the laboratories and observe each student or group of students".

Previous works in the LT2C laboratory made by Gravier et al. [Gravier06] proposed to enforce the group awareness with a communication tool based on a chat, VoIP and a "group awareness enforcement service" which provides a tele-presence indicator and allocates a color to each user. When an interaction occurs, the widget's color of the front of the HCI of the remote laboratory changes to the color of the user responsible for this interaction. In addition to those tools for computing the group awareness, a system based on ontologies and semantic rules is applied to manage the access to the resources [Gravier11]. This system was also adapted to run in a virtual world, Wonderland [Fayolle10]. Virtual worlds can help increasing group awareness, since all users are virtually present in the virtual world, which already provides group awareness [21]. In addition, "the spatial layout of the 3D world coupled with the immersive audio provides strong cognitive cues that enhance collaboration" [Fayolle10].

Different approaches can be considered in order to achieve group awareness in collaborative applications. Communication tools are the mandatory components, but other tools can be added in order to enforce collaboration. Anyhow, in order to handle

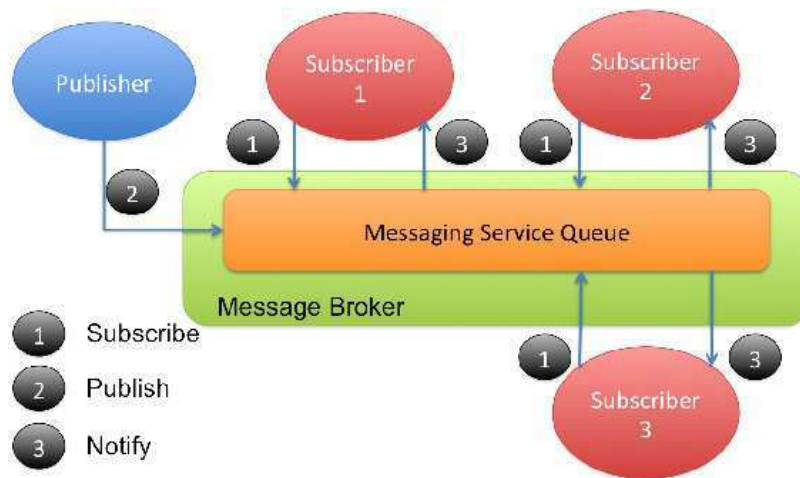


FIGURE 3.9 – Publish/Subscribe paradigm.

collaboration, the system, both on server and client side, should be able to send and receive messages from different users. The following subsections will present how we handled collaboration at the level of the middleware and in the multimedia HCI.

3.3.1 Middleware

A possible way of building middleware for remote laboratories is using RPC³. RPC enables the remote release of operations on an application server and to manipulate objects (in the Object Oriented Programming paradigm).

Because of its multi-platform capacities, RMI⁴ was studied. RMI adds object orientation to RPC. It is issued from the Java programming world. However, RMI is synchronous, thus until the client receives data, the execution of the program is blocked. Moreover, RPC was originally constructed for point-to-point communication, and then is not adapted for multi-user applications.

An alternative to RPC is the **Java Messaging Service (JMS)** standard. JMS is an asynchronous MOM, following the publish/subscribe paradigm explained by [Eugster03] and illustrated in figure 3.9. The procedure is the following :

1. Clients *subscribe* to a shared discussion channel.
2. A *publisher* emits a message on the channel.
3. The *broker* redistributes the published message on the channel to all users.

MOMs provide important features when designing collaborative remote laboratories :

3. Remote Procedure Call
4. Remote Method Invocation

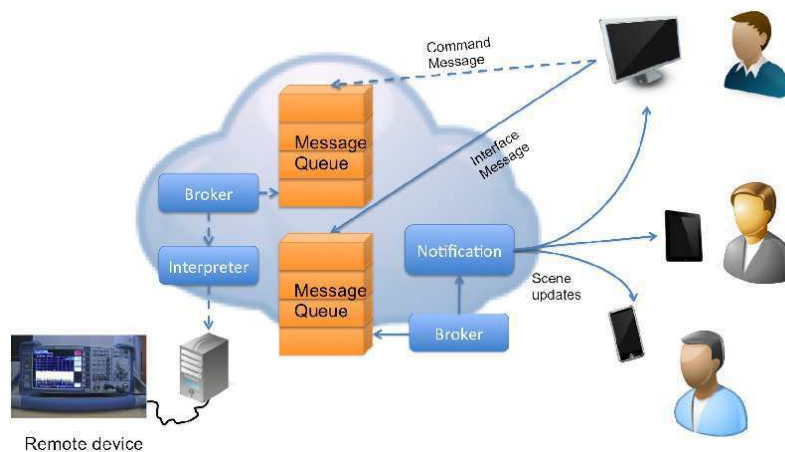


FIGURE 3.10 – Collaboration and Messages delivery.

- Asynchronism in the delivery of the response.
- Robustness and availability.
- Guarantees of scheduling that ensure that messages are received in the order they were emitted.

We propose to reproduce the publish/subscribe paradigm for multimedia services. The will to keep a loose coupling between the client and the server, pushed us to the creation of two kinds of channels for interaction. Both channels rely on the publish/subscribe paradigm. The first channel is the commands channel. Commands are sent to the middleware for processing. The second channel is the interface channel. This channel is responsible for updating part of the BIFS scene graph. It is one of the main components to present the group awareness in the scene. Figure 3.10 illustrates how collaboration is handled within our framework. All these components are part of the Command Interpreter presented in section 3.1.1.

3.3.1.1 The Command Channel

When a widget of the multimedia interface is activated, a message is sent to the middleware. This message contains information required for the middleware to convey the command, such as user's name, widget ID, the interaction's nature, etc. The message is formatted as JavaScript Object Notation (JSON). The middleware serializes it in the message queue. A message broker retrieves messages according to the scheduler and pushes them to the interpreter. The interpreter processes them in order to send to the device the corresponding hardware function.

3.3.1.2 The Interface Channel

The same kind of mechanism is employed for the Interface Channel. A JSON message containing the widget ID, user's name, etc. is published in a dedicated queue. The broker, according to the scheduler, pushes the message to the notification center. This module then pushes to users the scene updates.

The next subsection will present the solution we have adopted in order to realize the notification center and to create the group awareness on the client side.

3.3.2 Collaborative Interactive Multimedia Interface

Proposing a middleware that supports collaboration is necessary but not yet efficient for users to sense the group awareness. In this subsection we present how group awareness is supported in our system. This work is the follow-up of [Gravier08]; this previous work did not rely on multimedia but on a stand-alone Java application. Group awareness functionalities are a chat and VoIP tool for communication and changing colored based widgets for seeing "who is doing what".

For handling group awareness, each user receives a different color. The client sends widget information (user login, user color, widget ID) with the WebSockets protocol. The WebSocket server broadcasts this message to all the connected users, and the scene updates are handled in the ECMAScript code corresponding to the message received. The widget changes its color for every connected clients, depending on the user. This feature allows each user to know what action is made on the interface and who made it. A tele-presence indicator is then necessary in order to represent the association between a user and a color.

In order to improve group awareness, we ease the communication between users by using an audio communication tool. Receiving multimedia content (audio, video, BIFS) and streaming audio content to the server at the same time all over the session could require too much CPU capabilities, especially on mobile devices. In order to limit this, we have implemented a walkie-talkie tool, which offers audio communication without continuously acquiring audio data on the client side and streams them to the middleware. The user experience may not be optimal in this situation but the CPU consumption is limited.

Implementation details of the BIFS scene are presented in Chapter 4.

3.4 Adaptation of the Human Computer Interface

In the industrial maintenance area, learning and being able to reproduce the full catalog of maintenance operations can be a difficult task. The complexity of this task is also rapidly exploding since the number of devices operated by a single human operator increases when these devices can be operated remotely. Computer-Aided Maintenance (CAM) systems must provide a formalism to express such procedures in order to ease the operators' actions and increase their productivity. From our point of view, on the HCI side, the CAM systems must also provide indications on which action has been made and "whisper" to the operator the next possible actions in the maintenance procedures. The aim is to limit the number of human errors, as well as increase the efficiency (time spent, quality, etc.) of the maintenance process. We propose, as part of the presented framework, to express maintenance procedures based on Semantic Web technologies. A rule-based engine provides the context logic to indicate the operator which next possible actions can be made according to the current procedures. In this subsection, we present the architecture of what we call the "whisper module" and its uses.

3.4.1 Whisper Module

The whisper module is illustrated in figure 3.11. The procedure expert edits the sequences of actions that are serialized in the Knowledge Base (KB). The operator sends actions to the middleware through the HCI's widgets. The Processing Blocks (PB) interprets the actions. It requests the KB which the current possible sequences are, based on previous inferences. PB also places the new action as the current one in the KB. The Reasoner Bloc (RB) infers which the next possible interactions are, according to sequence rules and the Notification Block (NB) pushes the results to the HCI.

We take the following example : Let A_i be interactions made on a HCI through widgets. Let S_j be sequences of interactions pre-determined and serialized in the system. S_j is an ordered list of actions A_i . Let's assume that :

$$S_1 = \{A_1, A_2, A_3\} \quad (3.1)$$

and

$$S_2 = \{A_1, A_2, A_4\}. \quad (3.2)$$

The operator sends the action A_1 . The system whispers that the next possible action recorded is A_2 in sequences S_1 and S_2 . The widget corresponding to A_2 is highlighted in the HCI. The operator then sends A_2 . In S_1 and S_2 , the next actions are respectively A_3 and A_4 , and the corresponding widgets are highlighted in the HCI.

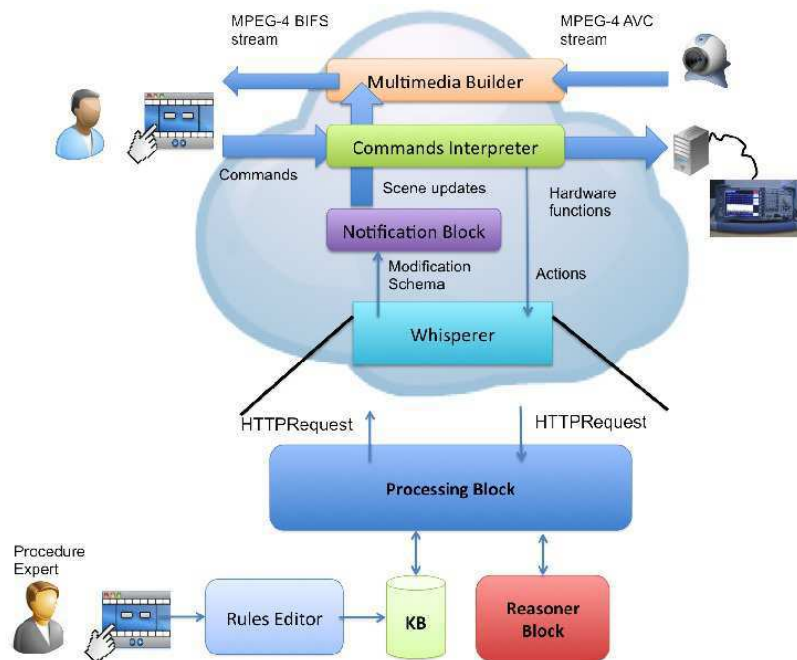


FIGURE 3.11 – Whisper module.

3.4.2 Expressing Sequences.

By sequences we imply an ordered list of actions on an HCI that compose a procedure. The usage of Semantic Web technologies to describe sequences of actions for e-Maintenance is motivated by the interoperability they provide, the detachment from specific programming language and mainly the reasoning they provide through rule languages.

RDFS and SWRL reasoners both follows the Open World Assumption (OWA), i.e. the assumption that the truth-value of a statement is independent of whether or not it is known by any single observer. Both rule engines propose a Human readable syntax for rule editing, which is in the following form :

antecedent => consequent

As a consequence, those criteria cannot be used to choose one solution over an other.

RDFS proposes in its core components `rdfs:Container` such as `rdf:Alt`, `rdf:Bag`, `rdf:Seq`. They are used to describe groups of things and are summarized in Table 3.1.

OWL has no support for ordering. `rdf:Seq` is not illegal but it has no logical semantics accessible to a DL classifier. Some propositions have been made in order to

Class	Container	Description
rdf:Bag	Bag container	indicates that the container is intended to be unordered
rdf:Seq	Sequence container	indicates that the numerical ordering of the container membership properties is intended to be significant
rdf:Alt	Alternative container	indicates that typical processing will be to select one of the members of the container. The first member of the container, i.e. the value of the rdf:_1 property, is the default choice

TABLE 3.1 – RDF containers

implement sequences such as [Drummond06]. But they are not a natural way to deal with sequences in OWL whereas rdfs:Container can express sequences with its core components.

Considering the better expressiveness of RDFS for this kind of problem, we made the choice of representing the sequences using RDFS/RDF.

3.4.3 Use Trace

Back to our use case, Bob has now finished to perform maintenance operations on Alice's device.

Let's T be the trace of Bob's actions.

An example of T could be :

$$T = \{A_3, S_2, S_5, A_3, A_3, S_1, \dots\} \quad (3.3)$$

where A_i is an action on the CAM system's HCI and S_j is a sequence of actions recorded in the system. Let's assume that in the Business rules of his company :

$$S_5 \equiv S_3 \quad (3.4)$$

and

$$Card(S_5) \gg Card(S_3) \quad (3.5)$$

The system should notify Bob that he performed one or several useless and/or suboptimal actions in the maintenance procedures. In order to realize it, the full use's trace of Bob has to be serialized in the system before being analyzed.

Since the system could be used by somebody else than Bob, the use's trace needs to contain the name of the operator. The system could also be collaborative, i.e. Bob could be using the system at the same time as James who has other competencies for

the task. The analysis of the trace can thus give details on the full remote operation but also details on a single operation.

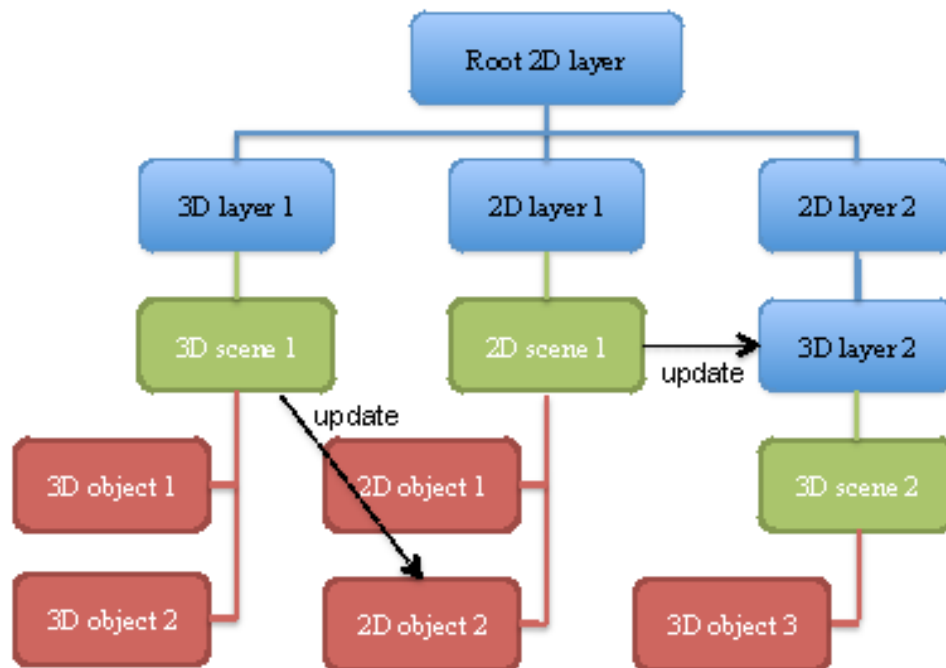


FIGURE 3.12 – MPEG-4 BIFS graph example.

3.5 Multimedia Delivery

3.5.1 Multimedia Composition using MPEG-4 BIFS

BIFS was introduced in section 2.1.4.1, therefore this section presents in more detail its features.

MPEG-4 BIFS defines two major components : visual and audio. The audio component can be a mix between various audio sources and the visual component can be 2D, 3D or a mix between 2D and 3D objects. Using MPEG-4 BIFS, one can represent a scene as a hierarchical dynamic tree, where leaves are audiovisual objects, as illustrated in figure 3.12. This property is derived from VRML. When designing complex behaviors, branches of this tree can be created, modified or deleted as demonstrated by Tran et al [Tran04]. Currently, more than 200 nodes are supported and implemented in the BIFS standards and every node can be assigned with a "name" in order to identify it. Each node can be characterized with fields that define the nodes, like attributes in XML tags.

In order to support both 2D and 3D scenes, BIFS defines the **Layer2D** and **Layer3D** nodes. This allows having a 2D scene rendered on top of 3D scene.

The **Shape** node is defined in BIFS in order to include a visible entity inside the

scene, using its main fields **appearance**, which specifies the visual appearance of the entity (texture, color, etc.), and **geometry**, which specifies the shapes of the entity (square, circle, complex shapes, etc.).

BIFS is also capable of referencing other scenes inside the tree, using the **Inline** nodes. It is then possible to include scenes in a BIFS scene inside another BIFS scene. This feature eases adding/deleting/changing content.

BIFS derives nodes from VRML in order to support interactivity. Interactivity is achieved through Sensor nodes such as *TouchSensor*, *TimeSensor* or *Proximity Sensor*. BIFS also embeds ECMAScript nodes. This feature brings the possibility to manipulate the media objects of the scene using JavaScript. It provides the ability to store state information in local variables, such as mouse position in the scene for instance.

In order to propagate events, BIFS uses the VRML concepts of **ROUTEs**. They are connections of values of one field to another. The syntax of a route is the following :

```
ROUTE <OutNodeID>.outFieldName TO <InNodeID>.inFieldName
```

meaning that values of the field "inFieldName" of the node "InNodeID" is assigned with the value of the field "outFieldName" of the node "OutNodeID".

MPEG-4 defines channels named **Elementary Streams** (ES). These channels contain a fully or partially encoded representation of a single audio-visual object, scene description, or control information. ES are distinguished using Object Descriptors (OD) stored in the ES. OD is a "structure similar to a Uniform Resource Locator (URL), containing pointers to ES"[Signes00].

The scene description and the stream description are considered as ESs and are fully separated. In order to point to object descriptors, the scene descriptions use OD identifier, the **OD_ID**. This allows to multiplex some other meta-information in the scene, such as images, video, etc.

The transport layer of MPEG-4 systems is "media unaware and delivery aware" [Signes00]. No specific deliver layer is defined by the standard, and MPEG-4 media can be transported on existing transport layers such as RTP, RTSP, MPEG-2 TS and more recently Hypertext Transfer Protocol (HTTP) streaming.

MPEG-4 systems and MPEG-4 BIFS scenes provide all the required tools in order to create multimedia remote laboratories HCI using mixed reality. The scene descriptions enable a high level of interoperability between terminals. Every kind of media object can be inserted in the dynamic scene and can be multiplexed inside a single format, the MP4 format, without having to deal with multiple formats. Moreover, MPEG-4 provides the ability to use classical transport protocol in order to deliver the multimedia content, like any kind of media. Other existing scene descriptors are designed in a way that the whole scene has to be downloaded before being rendered by the terminal.

3.5.2 Multimedia Delivery

In the previous subsections, the MPEG-4 BIFS standard was presented. This standard enables to create the multimedia content without having the hindrance of dealing with multiple formats. The multimedia HCI delivered to the remote user is built using MPEG-4 BIFS features. We have also presented how to estimate the pose of the device in the video stream delivered by the camera and how the video feedback of the remote device can be presented. In this subsection, we present the streaming technologies in order to deliver the whole multimedia content to end users.

The term **streaming** designates a principle for real-time (or slight delay) delivery of multimedia content. It is in opposition to the download delivering where the whole multimedia content has to be fully downloaded before being rendered. The streaming is still a download of multimedia content because of the data exchange between the client and the server. Data are downloaded continuously in RAM and are analyzed on the fly by the computer and quickly transferred to the media player for rendering.

3.5.2.1 Unicast, Broadcast and Multicast Streaming

Unicast is a point-to-point connection protocol between the client and the server. A connection is established for each and every client requesting the multimedia content, as illustrated by figure 3.13. In order to compute the bandwidth, all the bandwidths between all the clients and the server are added. Unicast is useful for on-demand media services. Each client is able to consume delayed media services.

The **Unicast Broadcast** is designed to consume real-time media services. Media content is transmitted on the media server, as illustrated by figure 3.14. The server sends packets that all terminals in the network receive. In order to compute the bandwidth, all the bandwidths between all the clients and the server are added.

The **Multicast** transmission is a routing process that consists in sending a packet to many clients simultaneously. It lightens the network's load. Packets are sent to a group without necessarily knowing the clients. Data is sent to all subscribers of the group. It implies the modification of the networks routers in order to access the multicast transmission.

The multicast streaming seems the best option in order to deliver multimedia content, especially for multiple users. However, multicast entails mastering all network elements in order to deliver media. Since it is impossible to control all the network items in our case (think of a student in a foreign country), we decided to use a broadcast solution in order to deliver the multimedia content to the end user.

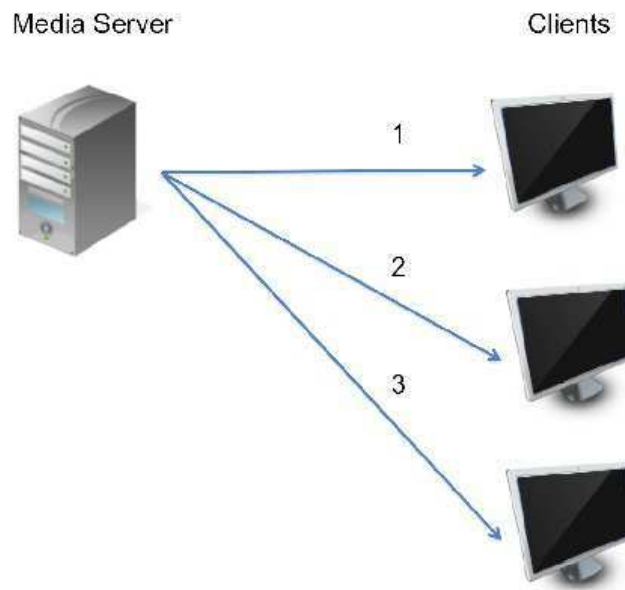


FIGURE 3.13 – Unicast streaming.

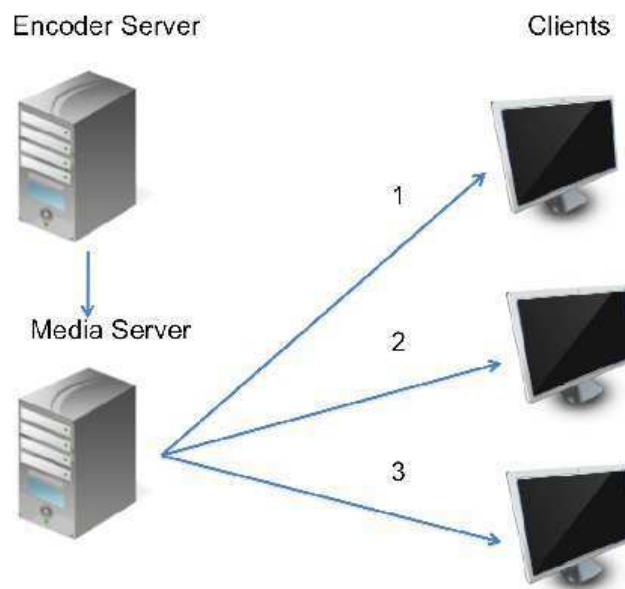


FIGURE 3.14 – Broadcast streaming.

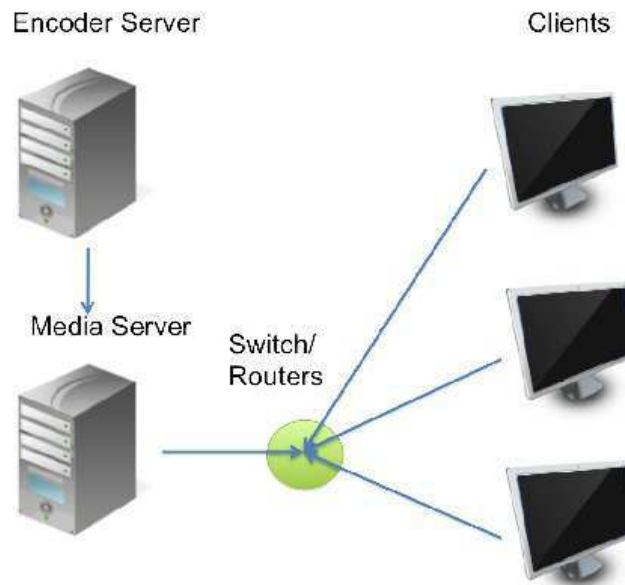


FIGURE 3.15 – Multicast streaming.

3.5.2.2 Streaming Protocols

The streaming methods mentioned above rely on streaming protocol to ease the transmission of data.

The **Transmission Control Protocol (TCP)** is a protocol for sending packets on a network. A dialog between the client and the server is established and copies of packets are kept on the server. The server waits for client agreement for sending data, thus if a packet is lost on the network, the server can send the copy to the client.

The **User Datagram Protocol (UDP)** protocol does not make a dialog between the sender and the receiver. If a packet is lost, the sender will not know it. The advantage of using UDP is its efficiency for real-time applications, if the loss of packets is acceptable.

The **Real-time Transport Protocol (RTP)** is a protocol for streaming real time data over the Internet. Usually, RTP packets are sent via UDP. RTP does not address resource reservation and does not guarantee Quality of Service for real-time services. For each multimedia session, a RTP Session is established, using the **Session Description Protocol (SDP)**. The SDP file provides information about the RTP packets, such as codec, timestamp, etc.

The **Real Time Streaming Protocol (RTSP)** is the central protocol for streaming applications. It is capable of adjusting the media bandwidth to the network congestion. It can also choose the best delivery channel such as UDP or TCP. It uses RTP as the

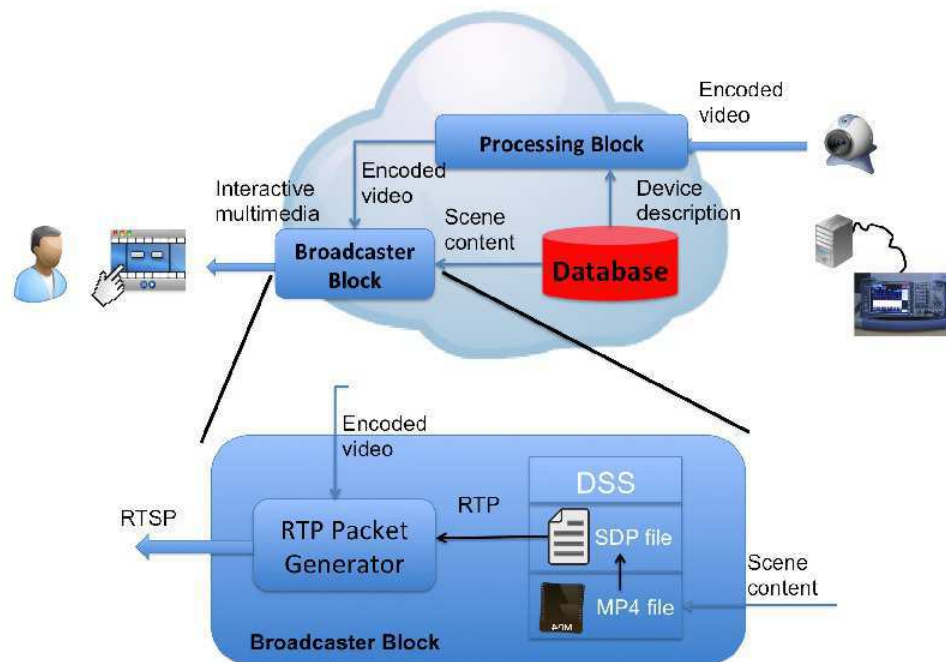


FIGURE 3.16 – Delivery.

underlying data delivery protocol since RTSP doesn't deliver the data itself. RTSP is a verbal protocol, it uses messages in order to control the media.

We presented here the streaming protocols. These protocols are included in the delivery of the multimedia content.

3.5.2.3 Delivery

The broadcaster block illustrated in figure 3.16 is realized using the Darwin Streaming Server, an open source server able to deliver multimedia content via RTSP and able to cast MP4 files. It is used as a relay of the RTP packets transmitted by the RTP packet generator (RPG). The BIFS scene contains a link to the SDP file describing the RTP packets delivered by RPG.

As highlighted in section 3.2.2.1, the device is identified frequently. This allow us to change the BIFS scene depending on the device captured by the camera. Changing the device interface is then a simple operation : changing the device in front of the camera. This features strengthens the reusability of the system.

On the client side, the multimedia player is responsible for decoding and displaying the RTP packets of the live video. The end user then just has to connect to the RSTP

address of the multimedia content to play and interact with it. RTSP clients are implemented in almost all media players.

3.6 Conclusion

This chapter presented our contributions to the online engineering field. These contributions are serialized in a framework for building and running online experimentation in the industrial maintenance and e-Learning areas.

We presented how augmented reality techniques are used to compute the 3D environment of the remote scene. We also presented how we extract the minimum video feedback from the stream (the display of the device in our example) in order to limit the bandwidth. We also presented how the interface is modified when a new device is placed in front of the camera.

In order to handle collaboration, we have presented how a middleware can be used on the server side. The will to keep a loose coupling between the client and the server, pushed us to the creation of two kinds of channels for interaction. The first channel is the commands channel. Commands are sent to the middleware for processing. The second channel is the interface channel. This channel is responsible for updating parts of the BIFS scene graph.

On the client side, group awareness is an utmost issue. In order to reproduce it, the multimedia interface needs to present the users' actions to each others. A tele-presence indicator is then necessary. Each user receives a different color and widgets are highlighted with one's color when activated. An audio communication tool is also proposed in order to create real communications between participants.

We have presented how the interface is adapted on the client side depending on the user's activity. We use Semantic Web technologies in order to represent sequences of actions.

Finally, we have presented how MPEG-4 BIFS is employed to present and create the multimedia content delivered to the end user. The broadcaster block is built on top of the RTSP protocol.

The next hapter of this work presents implementation details of the framework we propose.

When the only tool you have is a hammer, everything begins to look like a nail.

Anonymous

4

Realization and Results



While Chapter 2 presents issues of remote laboratories, Chapter 3 presents the overall architecture of the system resulting from our work. This Chapter details noticeable implementation points of the framework. We also introduce a first evaluation test and present the framework in the form of an open source project.



Chapter Plan

4.1	Realization and Implementation	91
4.1.1	Middleware	91
4.1.2	Instrument Driver	92
4.1.3	Multimedia Implementation	95
4.1.4	Feature descriptors Benchmark	105
4.1.5	Whisper Module	108
4.1.6	Authoring Tools	112
4.1.7	Conclusion	115
4.2	Results	116
4.2.1	MIRA Project	116
4.2.2	An Open Source Project	117

IN the previous chapter, we have presented the overall architecture of the online engineering framework. The framework can be divided into three main components : the driver is responsible for the communication with the device, the middleware conveys messages from clients to devices and the multimedia client is responsible for reproducing the device's HCI.

This chapter presents noticeable implementation points of the proposed framework and is organized into two major parts. The first one explicits instrument drivers construction and choices. MPEG-4 BIFS use for remote laboratories interface design are detailed as well as the ECMAScript API we proposed to enrich the BIFS capacities for remote laboratories. A local features and descriptors analysis is presented to determine the best compromise when tracking a device in the remote scene. We also present the user's interaction procedure modules implementation. The second part presents the overall realization of our system through a first evaluation test and an open source project realization.

4.1 Realization and Implementation

The realization of our *proof of concept* has been constructed using a spectrum analyzer. In this section, we detail implementation points of the proposed system. First, we present how the middleware is realized. Secondly, we present how instrument drivers can be implemented. We then explicit how MPEG-4 BIFS features can be employed to construct interactive interfaces for remote laboratories. Afterwards we present a benchmark of local features and descriptors. In the last part of this section, we present how semantic Web and RDF technologies can be used to model users' interaction procedures and how they can be analyzed consequently.

4.1.1 Middleware

Due to its multi-platform capacities and the previous expertise of the SATIN team in the LT2C laboratory, the Java programming language was chosen in order to implement the middleware.

An application server, based on the Java Enterprise Edition (JEE) is coupled with the MOM in order to provide mechanisms for :

- Users session.
- Network load.
- Concurrency.
- Persistency.

The JEE framework offers a large panel of application servers. A choice can be made in the main list composed of

- WebSphere from IBM¹.
- Glassfish from Oracle².
- JBoss from RedHat³.
- JOnAS from OW2⁴.
- Geronimo from the Apache Foundation⁵.

The choice was made for the Java Open Application Server (JOnAS). The first reason is the previous work conducted with it. The second reason of this choice is the implementation of the JMS APIs through the Java Open Reliable Asynchronous Messaging⁶ (JORAM). JORAM can be distributed as a standalone JMS server but it is also packaged inside the JOnAS distribution, ensuring a high level of compatibility.

4.1.2 Instrument Driver

4.1.2.1 Programming Languages

A complete generic architecture for devices would be very difficult to achieve. Almost systematically, a dedicated software development is needed in order to perform this link. The proof of concept developed during this thesis uses a spectrum analyzer FSL 3 from Rhode & Schwarz, which is fully IVI-compliant and was lent by the company in order to conduct our experiments. The IVI foundation proposes an implementation of the IVI drivers through Windows libraries.

Those libraries are proposed for the following languages : C++, C# and Visual Basic.NET, LabView, LabWindows/CVI, MATLAB, Measure Foundry, Visual Basic 6.0 and Agilent VEE Pro. Considering our programming language knowledge, the choice was made for C++ and C#.

The middleware we proposed in this thesis is implemented using the Java programming languages. In order to complete the link between the local computer and the local device, we firstly considered using C++ programs through **Java Native Interface**⁷ (JNI). JNI enables developers to access native code (through C++ functions for example) using Java functions. The steps in writing and running JNI programs are the following :

1. Create a Java class that declares the native method.
2. Compile the program (using *javac*).
3. Generate the C++ headers corresponding to the previous class (using *javah*).

1. <http://www-01.ibm.com/software/fr/websphere/>

2. <http://glassfish.java.net/fr/>

3. <http://www.jboss.org/>

4. <http://jonas.ow2.org>

5. <http://geronimo.apache.org/>

6. <http://joram.ow2.org/>

7. <http://docs.oracle.com/javase/6/docs/technotes/guides/jni/>

4. Write the C++ implementation of the native method.
5. Compile C++ code and generate native library.
6. Run the Java program using the Java interpreter (using *java*).

This procedure is long and difficult to debug for obvious reasons. In order to facilitate the development of the instrument driver, we decided to move to C# programming languages, even if a full driver could have been developed only using C++.

It is not easy to choose programming languages. While some claim that C++ is faster than C#, others claim that C# is a more user-friendly programming language. We found that the C# syntax is closer to the Java syntax and the IVI documentation is more suited for C# developers. Using the C# brought the problem of communication between the middleware and the instrument driver, given the different chosen languages. We think that in order to achieve a better loose coupling between the middleware and the device, both parts need to be separated and should exchange messages. Since messages from the client application are stored in a message queue and delivered according to scheduling, there is no need to propose such a component for the message delivery between the middleware and the driver. A simple TCP connection between the two entities exchanging the messages is enough in this case.

Since IVI libraries are only available for Windows platform, the C# programming language seems to be the easiest solution in order to write instruments drivers. In order to enable middleware and driver communication, a connection is required. We hosted our middleware in the same network as the device for our prototype. The access of the middleware is exposed to the Internet (using a public IP address) in order to use the platform, as opposed to the device. So, even if the security is not optimal, we constructed a TCP connection between the middleware and the device. In this connection, those two entities exchange messages. The following paragraph presents the messages content as well as driver implementation details.

4.1.2.2 Driver Classes

When designing the driver program, we choose to implement classes that match best the interactions with the spectrum analyzer. We have identified in the device's HCI modes, sub-modes and validation widgets as presented in figure 4.1. The modes of the spectrum analyzer correspond to the following functionalities of the device :

- **FREQ** : represents the frequencies acquisition mode. The sub-modes of FREQ are the frequencies parameters possible to set for the signal acquisition such as "Center Frequency", "Start Frequency", "Stop Frequency", "Frequency Offset".
- **SPAN** : represents the frequency span acquisition parameters. In this mode, it is possible to parameter the "Frequency Span", the "Sweep time", as well as start and stop frequency.
- **AMPT** : represents the amplitude parameters for the drawing of the acquisition curves. Sub-modes of AMPT enable the configuration of the amplitude.

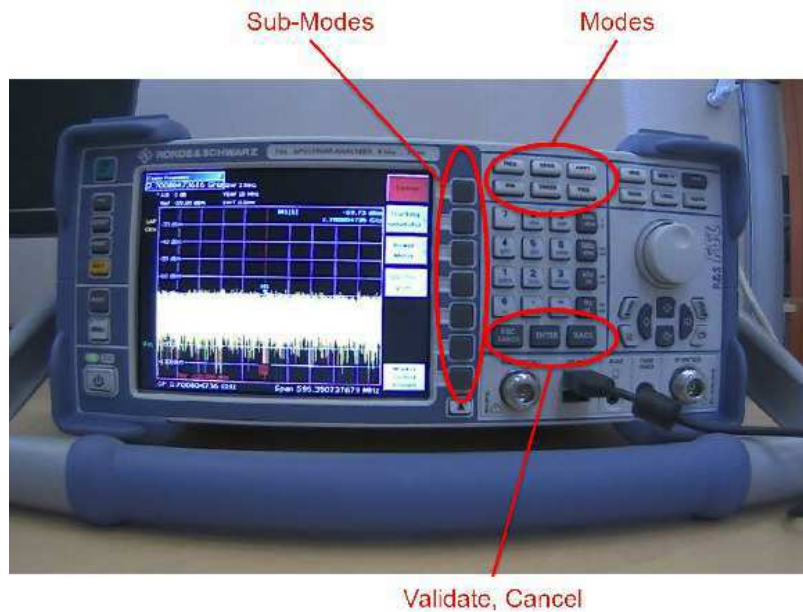


FIGURE 4.1 – Interactions modes.

- BW : represents the bandwidth parameters for the drawing of the acquisition curves. Sub-modes of BW enable the configuration of the bandwidth acquisition, such as Resolution, Sweep Time, etc.
- SWEEP : this mode enables the configuration of the sweep parameters for the acquisition of the signal.
- TRIG : this mode enables the configuration of the trig pulse parameters for the acquisition of the signal.

Thus, classes (in the Object Oriented Programming paradigm) represent the modes, while their attributes and methods represent the sub-modes and their access. As an example, the C# representation of the FREQ mode is available in Annex 6.

Messages between the middleware and the driver component are exchanged using a TCP connection. These messages must correspond to the driver implementation. They are composed of the mode, the sub-mode and the value set by the operator and are presented as a JSON object. The driver component parses messages, determines the mode and the sub-mode and then sends the corresponding hardware function to the device.

Another interesting feature of IVI drivers is their capacity to run in a simulation mode. This enables to run the driver component even if the instrument resource is not available.

4.1.2.3 Simulation

The IVI drivers offer the possibility to run the driver in a simulation mode. Simulation drivers are "virtual instruments" able to generate simulated data. The code developed can be reused without change when swapping to particular instruments.

This feature allows to use IVI drivers for remote as well as virtual laboratories. But this feature is not fully standardized. This entails some differences in the drivers implementations. For example, the driver for the FSL3 spectrum analyzer enable a running driver simulation but every signal delivered from the driver gives a zero vector. In order to conduct our tests, a very small simulation mode has been added to the middleware application. This mode is just developed in order to validate our approach.

4.1.3 Multimedia Implementation

In this subsection, we detail the implementation of the multimedia HCI using MPEG-4 BIFS. In Chapter 3, most of BIFS features for building remote laboratories clients are presented. The GPAC⁸ project [LeFeuvre11] is a multimedia framework that provides multimedia creation tools and a multimedia player capable of displaying BIFS scenes. The client application is realized using the GPAC framework. GPAC provides the **BIFS Text** (BT) format, that "is an exact textual representation of the MPEG-4 BIFS scene" in order to ease multimedia scene creation.

The general scene graph of the scene is illustrated in figure 4.2. The BIFS scene is composed of four main components. The **Authentication Screen** enables the user to log in the platform. This screen is composed of a background image, two text fields and a validation button.

The **General Script** component includes *Script* features used for the communication between the client and the server through ECMAScript functions. It contains global variables, mouse positioning, etc.

The **General Widgets** component includes widgets that are common to all remote laboratories HCI such as the "Presence indicator", the Chat tool, and the "Information Box". This box is necessary for displaying messages in the interface, such as network issues, etc.

The **Device's HCI** component is composed of all necessary widgets to present the device's HCI to the remote user. It includes background image(s), video area(s) and interactive widgets reproducing the interactions with the device.

In this subsection, we firstly presents how identification is achieved in the platform. The second paragraph illustrates how widgets are designed in the multimedia interface. The third paragraph illustrates the realization of an authoring tool to create remote laboratory HCIs.

8. <http://gpac.wp.mines-telecom.fr/>

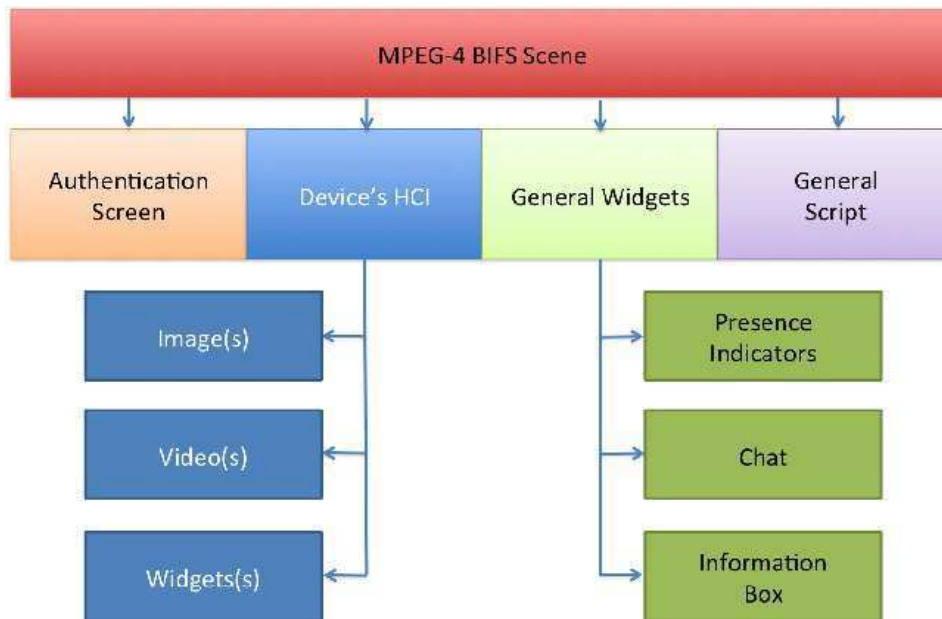


FIGURE 4.2 – BIFS Scene composition.

4.1.3.1 Authentication on the Platform

Authentication is needed on the platform to ensure that the user is allowed to interact with the remote device. The middleware and the JEE application server are responsible for hosting user information contained in databases and Lightweight Directory Access Protocol (LDAP) directories.

Authentication can be achieved using two mechanisms : when retrieving the multimedia content with the RTSP protocol and when using the multimedia content. Even if security is beyond this thesis' scope, we need to keep in mind that security is important.

The RTSP protocol provides tools to support authentication through "digest authentication" or "clear authentication". RTSP "digest" authentication [Franks97] comes from Web security mechanism. A hash function is applied on the password before sending it over the network. The "clear" authentication consists in retrieving the multimedia content through its URL and including the login and the password inside the URL. In this case, the password is sent in plain text over the network. While digest authentication provides a secured mechanism, it can be difficult to implement this protocol. Implementing digest authentication would entail a communication between the middleware and the RTSP server and a large amount of development on the Darwin Streaming Server to enable communication with the middleware has to be considered. We keep in mind this issue, but we decided not to implement digest authentication, but rather authentication through the multimedia interface. In our use case, somebody is

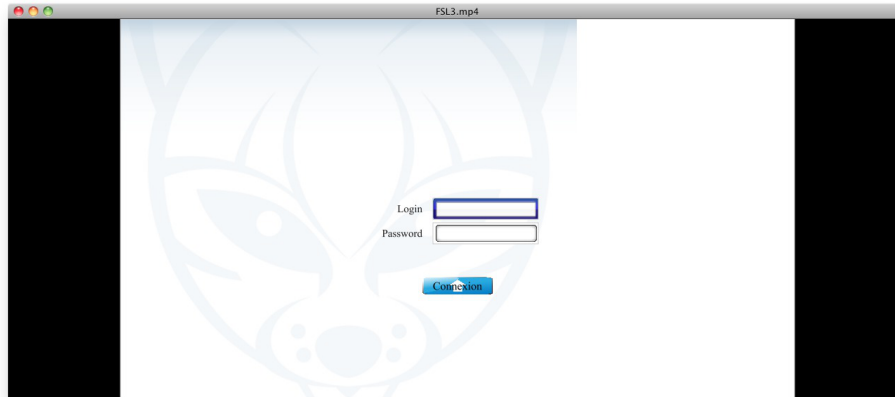


FIGURE 4.3 – Authentication Screen.

always near the real device and can verify the users identifications.

In order to achieve authentication on the platform, the multimedia client application needs to communicate with the middleware and send the user's login and password. The GPAC project implements a very interesting feature in order to ease client/server communication : **Asynchronous Javascript and XML**⁹ (AJAX). This feature enables to send HTTPRequests to the server. It is implemented as XMLHttpRequest in the BIFS ECMAScript. Using AJAX, we can send the entered user's login and password to the remote server for processing. Again, this entails sending user's login and password clear over the network. We could improve the AJAX implementation of GPAC to support the HyperText Transfer Protocol Secure (HTTPS), which is a composition of HTTP and Secure Sockets Layer (SSL), but it is beyond this thesis' scope.

On the client side, an authentication screen is proposed to the client. This screen is illustrated in figure 4.3. The middleware is responsible for getting the user's login and password and find the matching in the databases and LDAP directories. When the log in is successful, the device HCI is presented.

4.1.3.2 Widgets

A **Widget** is a contraction of the words **window** and **gadget**. It designates a graphical interface component such as button, list, scroller, etc. The BIFS graphics and interactive capacities enable to design and use widgets in remote laboratory HCI. In this paragraph, we detail the implementation of noticeable widgets. It is important to notify the reader that the plan's origin in the BIFS system is located in the middle of the

9. <http://www.w3.org/TR/XMLHttpRequest/>

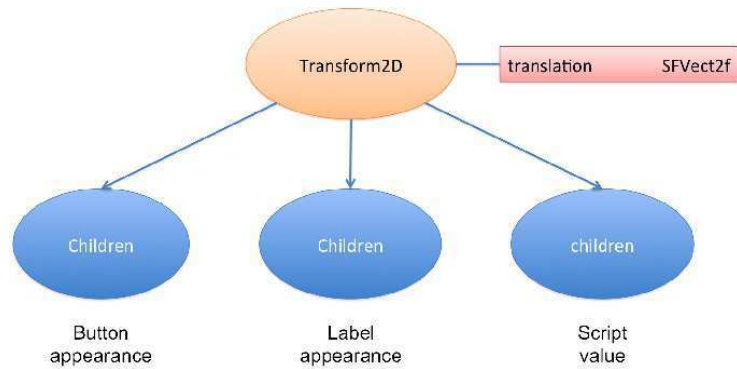


FIGURE 4.4 – Button pseudo-graph.

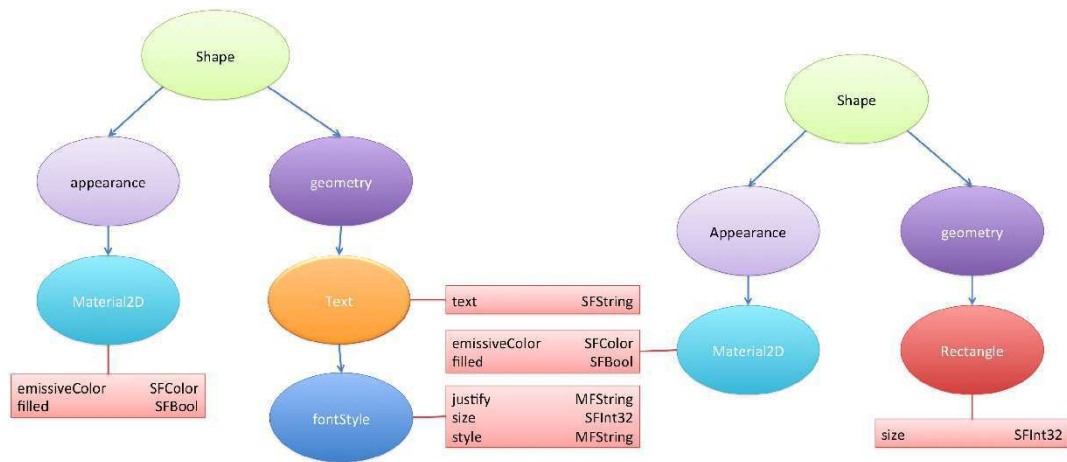


FIGURE 4.5 – Button details pseudo-graph.

rendered image.

A *button* widget is an interactive component that is characterized by its *touchable* feature, its position, its appearance, its color, its labels (and their position, color, font style, value, size). In order to design *buttons* in the BIFS interface, we propose the BIFS pseudo-graph represented by figure 4.4. Buttons are represented as *Transform2D* nodes having three children, one for handling the button appearance, one for handling the button's label appearance and the last one for handling Script values. The *Transform2D* node is characterized by its *translation* field that gives its coordinates in the scene. Both button and label appearance nodes are illustrated in figure 4.5. Buttons are coupled with BIFS *TouchSensor*, in order to reproduce the "touch" interaction of the device's real buttons. All buttons receive an identifier through their *Script* node. This identifier helps representing the interactions modes of the spectrum analyzer in the same way it does

for driver developments. In order to enable interactivity, we link those buttons using BIFS *ROUTES* with the *General Script* node in order to send their identifier to the middleware for processing.

A **knob** widget is a widget generally with a circular shape. Its goal is to reproduce real circular controllers such as potentiometers and is characterized by its rotation. This widget is constructed using the same schema, but using the *DiscSensor* node, that gives the possibility to access its rotation angle. The widget needs to rotate with user's interactions. This animation is realized using ECMAScript.

In order to let users enter text for login and password or chat messages, we also must reproduce **Text edit** widgets. Text edit widgets are realized with the **InputSensor** node, which "is used to add entry points for user inputs into a BIFS scene. It allows user events to trigger updates of the value of a field or the value of an element of a multiple field of an existing node" [ISO02]. Input devices are modeled as "devices that generate frame of user input". The frame consists in a list of values determined by its interface and their values are used to update the scene. The **StringSensor** device is the device we use to produce the Text edit widgets. Its interface is as follows :

```
StringSensorDataFrame [ [
    SFString enteredText
    SFString finalText
] ]
```

enteredText events produce characters when the keyboard is pressed. The values of this characters is a UTF-8 string. When a keystroke is pressed, the *enteredText* is moved to the *finalText* and is then set to the empty string. The **StringSensor** is available in the scene thanks to the OD framework :

```
ObjectDescriptor {
    objectDescriptorID 41
    esDescr [
        ES_Descriptor {
            ES_ID 8
            decConfigDescr DecoderConfigDescriptor {
                streamType 10
                decSpecificInfo UIConfig {
                    deviceName "StringSensor"
                }
            }
        }
    ]
}
```

In order to display the video streamed from the middleware, i.e. the video feedback of the remote device, the *VideoTexture* feature of BIFS contains the URL of the SDP files :

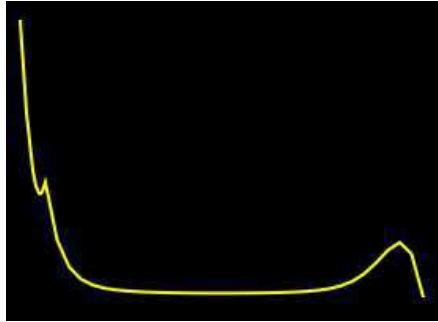


FIGURE 4.6 – Drawing of Bezier curves using XCurve2D node.

```

Transform2D {
  children [
    Shape {
      appearance Appearance {
        texture MovieTexture {
          url ["XXX.sdp"]
        }
      }
      geometry IndexedFaceSet2D {
        coordIndex [0 1 2 3]
        coord Coordinate2D {
          point [x1 y1 x2 y2
                x3 y3 x4 y4]
        }
      }
    }
  ]
}

```

Graphics capabilities of BIFS scene enable any kind of other widget families, such as *scrollpane*, *slider*, *label*, etc. using features such as *Layer2D*, *Layer3D*, *XCurve2D*, etc.

When the driver of the device is in simulation mode, the middleware is notified and forwards this notification to the client application. The feedback area of the multimedia HMI switches from the video mode to a graphics mode. The cloud points the driver delivers in simulation mode are then displayed in the multimedia, as shown in figure 4.6. This can be achieved thanks to the XCurve2D node of the MPEG-4 BIFS standard, which allows the drawing of Bézier curves.

Figure 4.7 shows an example of a device interface realized using BIFS technologies. Each button of the real device is reproduced using BIFS TouchSensors and have globally the following structure :

```

Transform2D {

```

```
scale 0.5 0.5
translation 163 119
children [
  Shape {
    appearance Appearance {
      material Material2D {
        emissiveColor 1 1 1
        filled true
      }
    }
    geometry Rectangle {
      size 70 30
    }
  }
  TouchSensor {}
]
children [
  Shape {
    appearance Appearance {
      material Material2D {
        emissiveColor 0 0 0
        filled true
      }
    }
    geometry Text {
      string ["SPAN"]
      fontStyle FontStyle {
        justify ["MIDDLE"]
        size 18
      }
    }
  }
]
children [
  Script {
    ...
  }
]
}
```

Buttons are composed of three children nodes. The first one represents the shape of the button. The second one represents the label of the button and the third one is javascript functions that enable communication with the middleware using HTTP requests.



FIGURE 4.7 – BIFS Interface.

4.1.3.3 MPEG-4 BIFS ECMAScript API and Group Awareness

In order to propose group awareness tools, the BIFS scene has to be updated for each and every user, every time a user has done a single action. The main issue here is to propose a mechanism for updating BIFS scenes using middleware messages.

Different approaches can be considered in order to realize it. The first one would be to implement a **BIFS-Commands** server. BIFS-commands can update any part of the scene using three basics commands : Insert, Delete, Replace.

The second one would be using the **MPEG-U** standard[Concolato09]. MPEG-U is a standard for rich-media user interface and has been built as an extension of the W3C Widget standard¹⁰ (WPC). Widgets defined in the MPEG-U standard are capable of communicating with each other through UPnP/DLNA services or local services of other widgets.

However, we strongly believe that one of the major feature of HTML celebrity is JavaScript due to its user friendly programming language and its strong API for inter-activity and data management. So, we decided that the push/pull functionalities of the notification center should be accessible from the BIFS ECMAScript. This would allow an easy way to develop and modify any BIFS scene. ECMAScript is integrated into the scene by wrapping the script into *Script* nodes. It allows to have enriched interaction in the scene. Scripts can be activated by different means, including timed events and user actions. Then we have chosen this to enrich the ECMAScript capabilities of BIFS.

4.1.3.3.1 WebSockets

The WebSockets protocol [IETF11] is currently under standardization by the Internet Engineering Task Force (IETF). This protocol permits to open a permanent bidi-

10. <http://www.w3.org/TR/widgets/>

rectional connection between the clients and the server. The goal of this protocol is to develop real-time multi-user applications.

After the *handshake* between a client and the server, i.e. if the server allows the connection with that client, the client and the server can freely send data to each other via this protocol. They are sharing a HTTP port with the available content, avoiding proxy or firewall problems. The WebSocket protocol provides real push/pull functions.

Since MPEG BIFS can embed ECMAScript, the usage of the WebSocket API [W3C12] could be added. The WebSockets API is currently under standardization by the World Wide Web Consortium (W3C).

In order to add push/pull functionalities to the framework, we have added the WebSocket API to the MPEG-4 BIFS ECMAScript API of the GPAC framework. GPAC is built on top of modules that help extending the player capacities. We then designed a GPAC module for interpreting the WebSocket API inside BIFS scripts. The current implementation in the GPAC module does not take into account the whole WebSocket interface¹¹ but is inspired from it and is as follows :

```
Class WebSocket {  
  
    WebSocket(SFString url, optional SFString protocols);  
  
    readonly attribute SFString url;  
  
    // ready state  
    const unsigned SFInt32 CONNECTING = 0;  
    const unsigned SFInt32 OPEN = 1;  
    const unsigned SFInt32 CLOSING = 2;  
    const unsigned SFInt32 CLOSED = 3;  
    readonly attribute unsigned SFInt32 readyState;  
    readonly attribute unsigned SFInt32 bufferedAmount;  
  
    // networking  
    attribute Function? onopen;  
    attribute Function? onerror;  
    attribute Function? onclose;  
    readonly attribute SFString protocol;  
    void close();  
  
    // messaging  
    attribute Function? onmessage;  
    void send(SFString data);  
};
```

11. <http://www.w3.org/TR/websockets/>

The constructor takes two arguments. The first one specifies the URL to connect with. The second one specifies a subprotocol that the server provides if any. The *url* attribute returns the URL specified in the constructor and the *protocol* attribute returns the subprotocol specified in the constructor. The *readyState* attribute specifies the current state of the WebSocket connection between the client and the server using the values :

- CONNECTING : the connection is not established. This is the default value.
- OPEN : the connection is established and communication is possible.
- CLOSING : the connection is closing.
- CLOSED : the connection has been closed or interrupted.

The *onopen* function is called after the handshake between the client and the server has been established. The *onerror* function is called when the connection is interrupted, following network issues for example, and the *onclose* function is called when the connection between the client and the server has to be closed. The *onmessage* function is called whenever a protagonist receives a message.

The *send* function sends the message to the WebSocket server. The message is the SFString argument *data*. The *bufferedAmount* attribute then returns the number of bytes of application data that have been queued using the *send* function. The *close* function calls the end of the connection.

This GPAC module has been realized using the libwebsockets¹² that is a C library available with the LGPL license.

4.1.3.3.2 Tele-Presence indicator

We have designed a chat tool inside the BIFS interface. Messages are printed in the dedicated window with the color corresponding to the writer of the message. Figure 4.8 shows the chat tool and a telepresence indicator designed in the multimedia interface using WebSockets. Each user's name is displayed with its given color. Below, the chat window shows the messages exchanged by users.

4.1.3.3.3 VoIP tool

For the VoIP tool, the **FMOD**¹³ libraries combined with the **FFMPEG**¹⁴ libraries are used to realize the acquisition of the voice of users at the client side, packetized and to stream it to the application server. The application server then multiplexes the digitalized voice within the BIFS scene, which is delivered to the end-user. The current available API is the following :

```
Class AudioTool {
    AudioTool(SFString url, SFInt32 port);
```

12. <http://git.warmcat.com/cgi-bin/cgit/libwebsockets/>

13. <http://www.fmod.org/>

14. <http://ffmpeg.org/>



FIGURE 4.8 – Chat tool and Tele-presence indicator.

```

readonly attribute SFString url;
readonly attribute SFString port;

// networking
attribute Function? onerror;
void acquire();
void close();
};

```

The constructor takes two arguments. The first one specifies the URL to connect with. The second argument is the server port to stream the audio content. The attributes *url* and *port* return respectively the URL and the port specified in the constructor. The *acquire* function is responsible for acquiring audio data from the terminal microphone and for streaming the audio content to the middleware. The *close* function stops the acquisition of audio data.

4.1.4 Feature descriptors Benchmark

In Chapter 2 we have presented some local features and descriptors algorithms along with their usage in mixed reality applications. In chapter 3, we have presented the general algorithm of our proposed solution in order to construct the 3D environment for building mixed reality applications. We showed that the solution uses local features and descriptors in order to compute the device identification and its pose estimation. In this subsection, we present a benchmark of local features and their descriptors applied

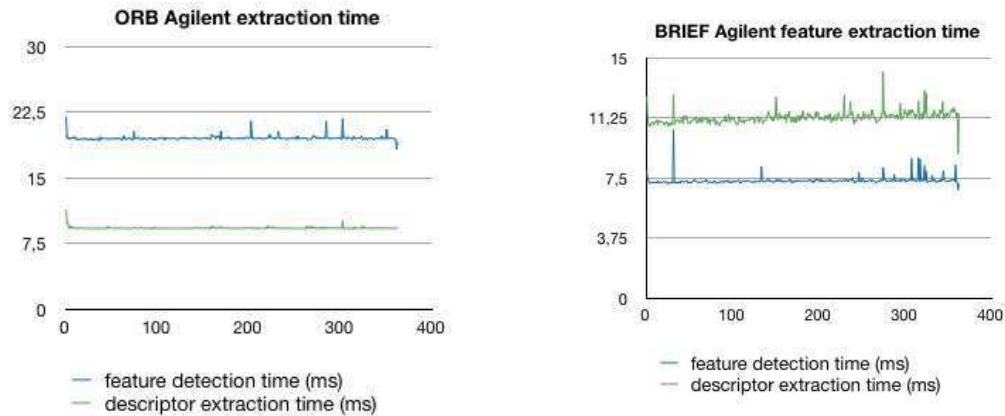


FIGURE 4.9 – ORB and BRIEF extraction.

to the potential devices we could control in engineering studies and available at the LT2C laboratory.

This section compares the SIFT, SURF, ORB and BRIEF features and descriptors through their capabilities against efficiency and invariance to rotation and scale. The implementation of those descriptors is provided with the **Open Source Computer Vision** library (OpenCV) [OpenCV12]. Its current version is 2.3.1.

This analysis shows the extraction time of local features, the extraction time of feature descriptors, the matching time between the reference image and the image and the sum of all those steps. This study is realized using ten devices, captured in the same conditions. The images' resolution is 640×340 . There are more than 360 images per devices. This benchmark is realized on a MacBook 2006 with a 2.13 GHz Intel Core 2 Duo processor and a 4 Go and 800 MHz DDR2 SDRAM memory.

Figure 4.9 and figure 4.10 show the results of feature extractions for the E4407B Agilent spectrum analyzer. BRIEF and ORB are faster in the extraction of both features and descriptors than SURF and SIFT.

This observation is made for all the devices we analyzed. There is a high difference of computation speed between descriptors. BRIEF is the fastest one in our benchmark, SIFT being the slowest. ORB is close to BRIEF, while SURF is a bit faster than SIFT. All the charts for the ten devices are presented in Annex 7.

Table 4.1 presents the scale and rotation invariance capacities of the analyzed descriptors. Only the SIFT and SURF descriptors are both scale and rotation invariant. Scale invariance ensures the recognition of objects in the scene when the distance between the camera and the object is changing over time. Rotation invariance ensures the recognition of objects when the angle between the object and the camera is changing. These are both important features to take into account when designing mixed reality

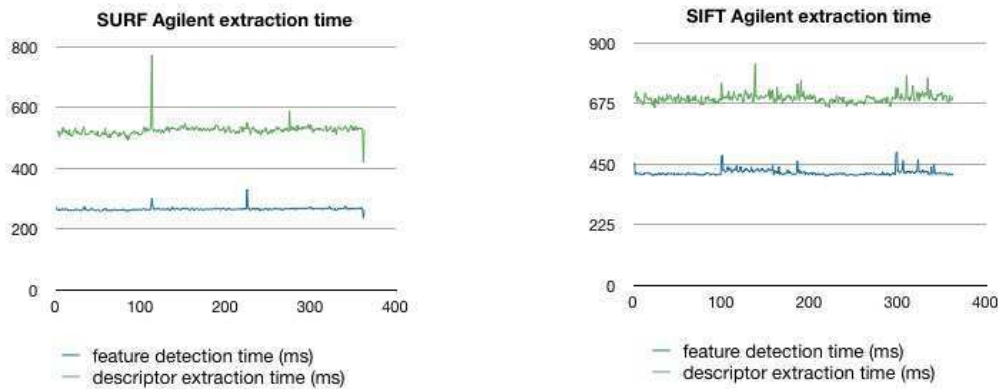


FIGURE 4.10 – SURF and SIFT extraction.

Features Descriptor	Rotation Invariance	Scale Invariance
SIFT	yes	yes
SURF	yes	yes
ORB	yes	no
BRIEF	limited	yes

TABLE 4.1 – Features Descriptors : rotation and scale invariance

applications. Since we want our system with the fewest calibration time, i.e. the angle and the distance between the camera and the object are not known and could change during the session, only the SURF and the SIFT descriptors can fit our requirements.

Extraction of features and descriptors is not the only time consuming operation to take into account. The matching between the reference image and the captured scene is also critical. In order to achieve matching between descriptors, we also used the interfaces provided in the OpenCV library. We compare here a brute force matching against the *FLANN* matching, after computing the SURF descriptors. The charts represented in figure 4.11 show the results of computing time for both matchings on the same device. The other graphs are presented in Annex 8.

Of course the extraction time of features and descriptors is strictly the same. The charts show that the *FLANN* matching decreases the computation time from 750 to 100 ms. The overall process takes about 800 ms.

These results show that computing SURF features and descriptors and matching in a real-time application using 640×340 image resolution is not possible on the tested computer. In order to solve this issue and to keep the real-time aspect of the application, the SURF extraction is not done for every images but for one out of 5. This implies that the relative position between the camera and the object does not move too fast. A more powerful computer could decrease this process.

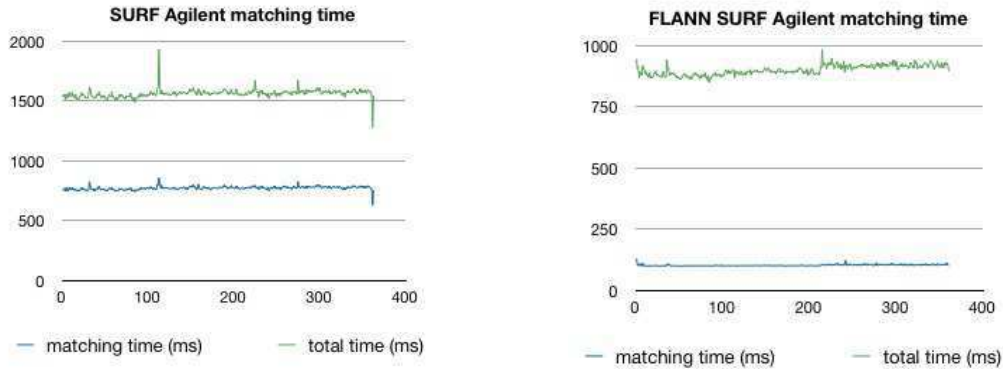


FIGURE 4.11 – Brute Force and FLANN matching.

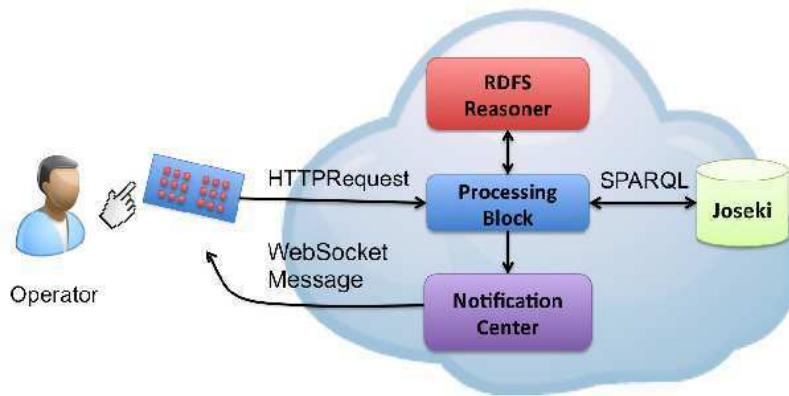


FIGURE 4.12 – Whisper module implementation.

4.1.5 Whisper Module

The whisper module hints users to next possible actions. It was presented in Chapter 3. The implementation of this module is a component of the middleware and is illustrated in figure ??.

In order to ease loose-coupling, the whisper module is accessible through a REST Web service architecture over HTTP requests.

The Processing Block (PB), that interprets HTTP requests, queries the KB through SPARQL Protocol and RDF Query Language (SPARQL) queries, which are the current possible sequences based on previous inferences. PB also updates the current system states (the current action) to the Knowledge Base (KB) through SPARQL/Updates (SPARUL) requests. The RDFS reasoner infers which the next possible interactions are

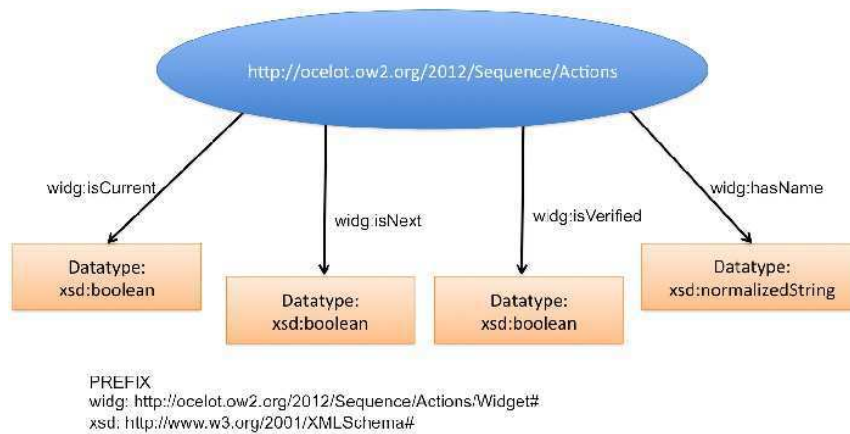


FIGURE 4.13 – RDF Graph of an action.

according to sequences rules and the Notification Center (NC) pushes the results to the HCI. The NC is the same as presented previously, since its goal is also to push HCI messages to remote users.

In order to achieve this module, a formalism to express sequences of interactions must be chosen.

4.1.5.1 Actions as RDF resources

Considering the better expressivity of RDFS for this kind of problem, we made the choice of representing the sequences using RDFS/RDF, as presented in section 3.4.2

Sequences are modeled using `rdfs:Container`. This allows the granularity of expressing compositions of ordered lists and choices of actions, with the power of Semantic Web tools. We define an action as an RDF resource, as presented in figure 4.13. The resource is identified with an Uniform Resource Identifier (URI) in the form of an Uniform Resource Locator (URL). This resource has 4 properties : `isCurrent`, `isNext`, `isVerified` and `hasName`. The three first ones are XML Schema boolean datatypes, whereas the fourth one is an XML Schema `normalizedString` datatype.

- `isCurrent` : defines if this action is the current one (i.e. if this action is the one being applied).
- `isNext` : defines if this action is following another. Its value is determined by the rules engine, as explained in following sections.
- `isVerified` : defines if an action has already been called in the sequence.
- `hasName` : defines the name of the action.

The full sequence is then a composition `rdf:Seq`, which represents the sequences of actions to perform and `rdf:Alt` which represents a choice between actions (e.g. A_3 or

A₄). A literal value is predicated to the `rdf:Seq` in order to identify the device.

As a framework, Apache Jena seems a good compromise with its RDF graph framework and its reasoner capacities. Jena is a Java framework that includes APIs to process RDF or OWL data. The RDF graphs are stored in Joseki [HP09], an HTTP engine that supports the SPARQL protocol and a protocol for SPARQL/Update.

4.1.5.2 Reasoning for Interactions Hints

The main goal of the rule engine is to determine the possible actions following the current one and depending on the previous ones. Rules can be described as the following pseudo-code inside a sequence :

```

PREFIX:
xsd: http://www.w3.org/2001/XMLSchema#
widg: http://ocelot.ow2.org/2012/Sequence/Actions/Widget#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

if {item_{m}, widg:isCurrent, 'true'^^xsd:boolean}
and if {item_{m-n}, widg:isVerified, 'true'^^xsd:boolean}
then
    {item_{m+1}, widg:isNext, 'true'^^xsd:boolean}

```

if the item (m+1) in the sequence has `rdf:Alt` type, we need to put all its elements at next also. In order to verify the triple described above, we implemented custom built-ins in the Jena rule-engine. This built-in is in the form `hasNext(?seq, ?x, ?y)`. It verifies that

```

{seq rdf:_(n) x} and
{seq rdf:_(n+1) y}

```

and that the previous actions in the sequences have been validated.

But the Open World Assumption is closely related to the monotonicity of first-order logic. It means that the reasoning system may accept new facts that might contradict previous ones. Then, the Jena Rule engine may deduce contradicting facts, such as :

```

{item_{m}, widg:isNext, 'true'^^xsd:boolean} and
{item_{m}, widg:isNext, 'false'^^xsd:boolean}

```

KB revision could be a solution in order to keep consistency. Some works have been conducted in this field such as in [Gros09] and [Halaschek-Wiener06]. In [Gros09], the authors introduce an annotation approach for prioritizing rules over the remaining

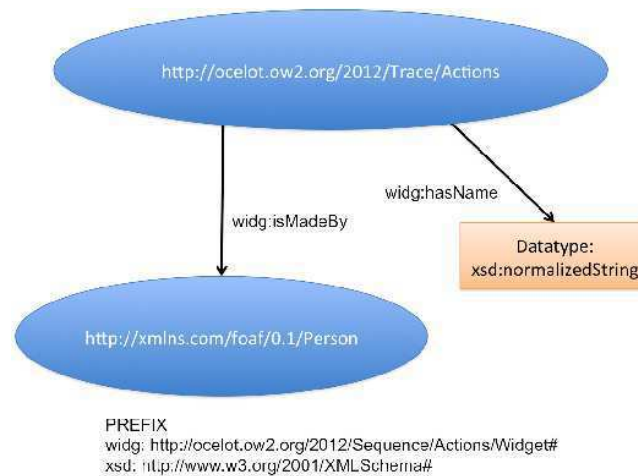


FIGURE 4.14 – RDF Graph of an action in the trace.

ones, in order to prevent contradiction clashes. The authors of [Halaschek-Wiener06] submit a system that maintains the internal state of DL reasoners. But this topic is beyond the scope of our work. In order to revise the KB and keep consistency, we therefore decided to remove old assertions and replace them with new deduced ones when a clash occurs.

4.1.5.3 Use Trace

Use trace analysis helps operators by proposing them improvements in their procedures. We also have decided to record traces using `rdf:Seq` for the same above-mentioned reasons. Each recorded action is identified by its name and by the person who made that action. The Friend of a Friend (FOAF) project¹⁵ aims at providing description of persons and links between them. The full trace is then a composition of RDF resources represented by Figure 4.14.

The use trace can thus be easily analyzed using pattern matching algorithm such as SuffixTree [Ukkonen92] in order to determine which actions and/or sequences have been made, and by whom.

But the RDF resources representing the sequences need a property to express equivalency in order to propose improvements to the operator. A disadvantage using RDF against OWL is that RDF does not provide support for symmetric properties. A symmetric property is a bidirectional relationship. OWL provides such a support with `owl:SymmetricProperty` that is a subclass of `owl:ObjectProperty`.

15. <http://www.foaf-project.org/>

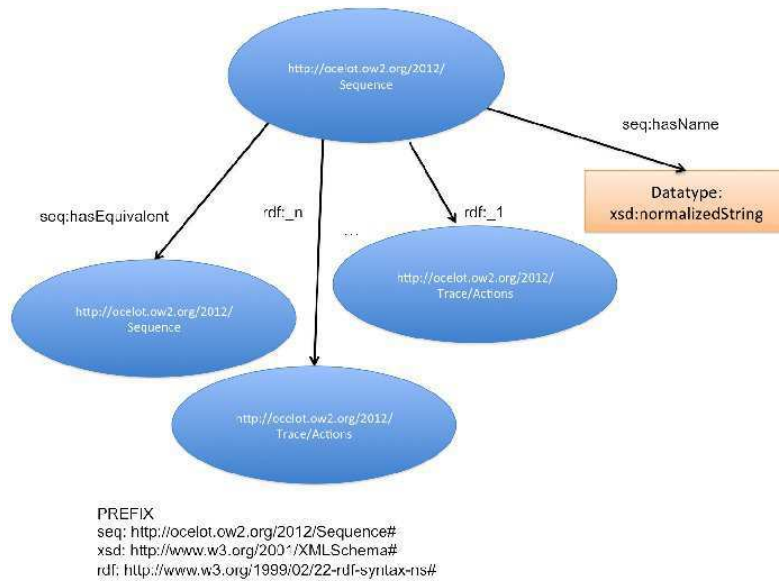


FIGURE 4.15 – Sequence representation.

Whence, when designing sequences, we need to explicitly express that :

$$S_i \Rightarrow S_j \quad (4.1)$$

and

$$S_j \Rightarrow S_i \quad (4.2)$$

The RDF Graph of the sequence predefined in the system can be summarized in figure 4.15. If the system determines that S_5 appears in the use trace of Bob, the system can notify Bob that he should be considering using S_3 instead. Again, the rules engine is responsible for doing those assumptions. The analysis of use traces is available in our system through a REST Web Service architecture. Once the rules engine infers new deductions and finds the next possible interactions, a notification has to be pushed to the client application.

Another option would have been to use the OWL sameAs built-in. However, this OWL built-in is designed to be used between individuals that capture the very same observable. This would have been an ontological mis-engineering issue, as the two sequences are *not* identical, although they are equivalent.

4.1.6 Authoring Tools

We have presented the overall multimedia scene graph of the interactive multimedia remote laboratory. We have also presented how we implement the whisper module.

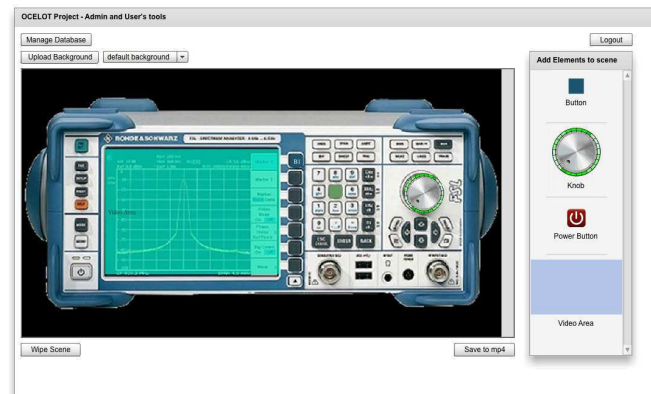


FIGURE 4.16 – Authoring tool.

In this subsection, we present the two authoring tools proposed in the framework. The first one creates the multimedia content for users. The second tool enables the creation of sequences of actions. It also enables to express the equivalence of sequences.

4.1.6.1 Multimedia Content Creation

The user needs to upload an image representing the background of the scene composed of static pixels unchanging with time over the session. We call this image the **reference image**. It is stored in the system database. The user can add widgets in the interface and chooses their color, label and size. Those widgets can be linked with already defined device functions stored in the database of the system. The user also defines the streaming area that is the area where the images captured from the camera will be displayed in the scene. This area is also stored in the database. Once the user finishes creating the interface, the BIFS scene is generated and multiplexed within an MP4 file as a media track. This is achieved thanks to the GPAC project. Figure 4.16 shows the authoring tool of the system, employed to design the multimedia content of the spectrum analyzer. In that case, only the display of the device needs to be a feedback zone.

This authoring tool has been developed by a engineer trainee, Léo Chew, in the LT2C laboratory that I supervised. It is a Flex application developed using the BlazeDS¹⁶, a "server-based Java remoting and web messaging technology" that eases the Web client application integration with the system's middleware. I supervised the trainee for a 6 month period.

16. <http://opensource.adobe.com/wiki/display/blazeds/BlazeDS>

FIGURE 4.17 – Sequences Editor

4.1.6.2 Sequences Creation

A sequence editor is provided in the framework. It enables users to create sequences and store them in the system's knowledge base, as well as to define which sequences are equivalent to others. The interface of the sequence editor is to be found in figure 4.17.

Sequences are displayed in the BIFS interface using the operators "||", "<", "(", ")" and the widgets labels. When stored in the KB, the "||" operator indicates `rdf:Alt` elements and the "<" indicates `rdf:Seq` elements. For example, the displayed sequence $FREQ < (kHz || MHz) < ENTER$ in the interface means that the first element of the sequence is *FREQ*, the second element is a choice between *kHz* and *MHz* (represented as a `rdf:Alt` element in the sequence) and the third element of the sequence is *ENTER*. When using the sequence editor, the operator directly uses the device interface. Activating a widget adds it to the edited sequence.

The second part of the tools proposes all the stored sequences. The user can then choose one of them and indicate equivalence between sequences. The sequences and their equivalence rules are stored in the Knowledge Base. This setup step is to be done only once for a given device.

4.1.7 Conclusion

In this section, we have presented implementation details of the system we propose for remote laboratories. On the driver side, we propose to use the IVI driver in order to ease interoperability, through their C# libraries. We have designed programming classes in order to represent best the interaction modes of the device. On the client side, we use MPEG-4 BIFS and have presented how widgets can be designed in order to recreate the "real" device interactions. The ECMAScript API addition which proposes the WebSocket API and the audio communication has been presented. A benchmark of local features and descriptors has been realized. This benchmark lead us to the choice of the SURF algorithm in order to track the device in the video stream delivered by the IP camera, even if its computation time is high. We presented how the semantic Web and RDF technologies can be used in order to model sequences of actions and how users' actions can be analyzed. Finally, two authoring tools are introduced. They ease the creation of the multimedia content for online engineering and the creation of sequences of actions in the processes.

4.2 Results

An online engineering framework has been introduced. This framework proposes to build collaborative remote laboratories using interactive multimedia based on MPEG-4 BIFS and mixed reality. We propose here to present the first evaluation tests conducted to validate our approach. This evaluation has been conducted as part of a *MIRA* project. We finally present the Open Source project resulting from our works.

4.2.1 MIRA Project

The MIRA projects¹⁷ aim at "strengthening cooperation in international higher education and research". As part of a MIRA project, we set up a remote laboratory session with the Université Libano-Française of Tripoli (ULF) in Lebanon. This session took place in early December 2011, as part of a course on signal processing.

The pedagogical objectives of the ULF's course on signal processing are to give students skills on signal acquisition, digitalization and temporal and frequency representation. The course was divided into lectures and practical sessions, which were divided into simulation exercises and real manipulations. The simulation exercises were realized with Labview. The students had to display temporal and frequency representations of a given entry signal by varying parameters such as bandwidth, span, etc. The objectives of the manipulations were also to display temporal and frequency representations by varying parameters but on a real entry signal and to learn how to manipulate a real industrial device. The manipulations were realized using a spectrum analyzer hosted in Télécom Saint-Etienne (Saint-Etienne, France). 15 students followed that lecture.

The feedbacks from the students and the teacher were encouraging. The students liked the idea of manipulating a high quality device they are not used to see and more than 2000 kilometers away. The teacher pointed to the fact that the HMI presented was "really close to the real one" and really had "the feeling of manipulating the device". He gave a score of 4 out of 5 on the usability of the interface. He stated the configuration on the user side was really simple, given the fact that it consists of installing a video player and connecting to a RTSP address with it. The score is also 4 out of 5. He also said the transmission rate of commands between the client and the instrument was good, given the distance between the students and the device. The transmission rate of commands was evaluated at an average of 600 ms. The bandwidth was evaluated at 500 kbps. The students and the teacher did not use the communication tools the framework provides since they all were in the same room when manipulating the device but appreciated the notifications of actions from others. The teacher emphasized the utility of such framework for the sharing of very expensive instruments.

17. Mobilité Internationale Rhône-Alpes

However, both teacher and students pointed to a drawback. The perceived quality of the video feedbacks from the IP camera was not as good as expected. The quality was not "good enough". The quality of the video feedback was evaluated at 2 on a MOS scale. That could come from both the IP camera quality and the available bandwidth in the ULF, which was not optimal. A deeper Mean Opinion Scale (MOS) and Quality of Experience (QoE) analysis would certainly be an interesting avenue of research. However, since students never used this kind of device before, only their opinion on the "visual quality" of feedbacks can be relevant for our analysis. The teacher is an expert in this domain, he is the only one able of comparing the usability of such devices.

4.2.2 An Open Source Project

The **Open source** designation applies to softwares whose license specifically meets the criteria established by the **Open Source Initiative**¹⁸, that is to say the possibility of free redistribution of the source code and derivative works.

In order to ease the software development of our system and to ease its evaluation by peers, we decided to open the source code of our system. This project is the **Open Collaborative Environment for the Leverage of Online insTRumentation**¹⁹ (OCELOT). OCELOT is a complete solution framework for projects of remote collaborative interactions. Using OCELOT, you can bring remote instrumentation solutions to your end-users faster. The possibilities range from online laboratories, remote industrial maintenance, cross-reality, etc.

OCELOT is distributed under the GNU Lesser Public License version 3²⁰, LGPL v3. This license, contrary to the GNU General Public License, limits the copyleft, i.e. the heritage of the license when deriving works. It is then possible to derivate the work and use other licenses.

This project is divided into four parts. The first one is the multimedia part, that is the BT file used to create the multimedia content. The second part is the player part. It includes the different plugins and modules for the GPAC player. It is mostly written using the C programming language. The third part is the image processing part. This part is responsible for decoding images, device pose estimation and RTP streaming. The programming language used for this part is C++. The fourth part is the instrument driver, written with C# languages. The last part is the middleware, mostly written in Java. Table 4.2 shows the different developed parts with their programming languages and some noticeable external libraries.

Figure 4.18 presents the final interface realized using the OCELOT framework on the spectrum analyzer, running on an Android tablet and on a laptop, with its different interface components. Figure 4.19 summarizes the functioning of the application.

18. <http://www.opensource.org/>

19. <http://ocelot.telecom-st-etienne.fr/>

20. <http://www.gnu.org/licenses/lgpl.html>

Tool	Language	Number of files	Number of code lines	External Libraries
Multimedia content	BT	1	5869	none
Player plugins	C	4	1524	FMOD, lib-websocket, FFMPEG
Image processing	C++	35	6742	OpenCV, FFMPEG, Xerces
Instrument driver	C#	5	661	IVI libraries
Middleware	Java	104	16348	Apache Jena, Commons Exec, BlazeDS

TABLE 4.2 – Open source realization

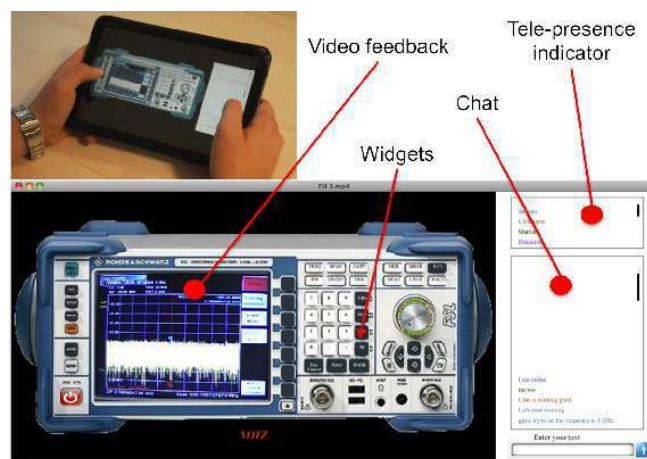


FIGURE 4.18 – OCELOT running on an Android tablet and on a laptop.

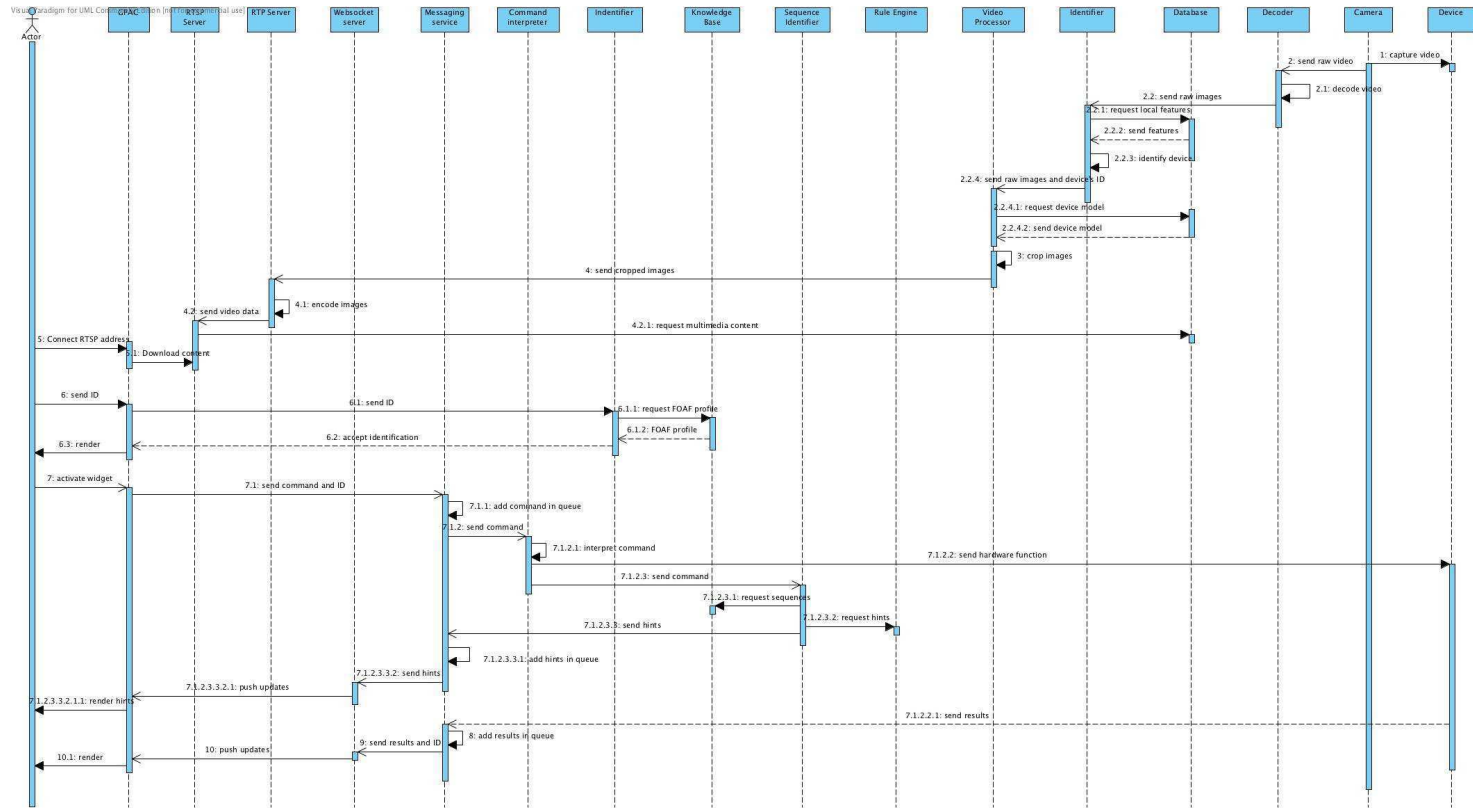


FIGURE 4.19 – Sequence diagram.

The sooner you start to code, the longer the program will take

Roy Carlson

5

Conclusion and Perspectives

▷ *Conclusion, discussions and perspectives* ◁

Chapter Plan

5.1	Synthesis	123
5.2	Discussions	125
5.2.1	Remote Laboratories Frameworks	125
5.2.2	Multimodality.	125
5.2.3	Collaboration.	125
5.2.4	Limitations.	126
5.3	Perspectives	127

ONLINE instrumentation and the remote control of devices must answer new challenges : **collaborative and computer-assisted work** as well as **usage context** in the HCI of remote devices. Nowadays, access modalities are mainly applicative, giving a better a video feedback of the remote apprehended scene. The objectives of this thesis are to break those limited approaches by proposing an interactive and collaborative multimedia interface of remote laboratories, using a contextualized multimedia stream and considering ubiquitous and pervasive computing.

In this Chapter, we first make a synthesis of the proposition of this thesis, resuming the main ideas of our approaches. The second section tries to discuss the advantages and limitations of our solution, through different criteria. The third section introduces the prospects following conducted works.

5.1 Synthesis

Online manipulation of devices is nowadays used in three different areas :

- Learning institutes mainly for distance learning curriculum.
- Industries for maintenance operations, production controls, etc.
- Research institutes to manage experimentations with researchers from various geographic locations (collaboratories).

Bringing devices online in the applicative world allows introducing new paradigms that were not possible on physical devices. The two main aspects in this domain are the **collaborative computer-assisted work** and the **contextualization**, i.e. personalization and screenwriting, of manipulation sessions of remote devices.

The **collaborative computer-assisted work** tries to allow operators to manipulate the same device while in different geo-locations to exchange competencies and knowledge in a common and shared task. In the industrial area, we can mention the Computer-Assisted Maintenance. When a maintenance operator reaches his competencies threshold, he calls a remote expert (phone, mail, etc.). This allows not only to realize the maintenance but also to bring the operator new competencies or knowledge. Collaborative online instrumentation permits to solve issues with a consolidation of competencies and performances by an enforcement of the dialogue between the operator and the remote expert, visualizing in real-time the apprehended scene. In the e-learning area, it amounts to reproducing online practical exercises that are often done by groups in local manipulation.

The personalization of instrumentation tries to adapt applications following the capacities/habilitations of users or tasks to realize.

The state of the art analysis conducted in Chapter 2 showed that mixed reality applications can improve the representation's fidelity of online manipulation. Seeing the real scene and interacting with the remote device leads to having a better understanding of the manipulated device and improves the "sense of being there", which is an utmost

issue in distance learning. Mixed reality systems demonstrate their performance and benefits in various fields and applications, such as remote maintenance, telemedicine, education, serious gaming, or culture. Most of the time, additional information is overlaid on the viewed scene, although the subject of study is in the same place as the user, the activity is limited to observation, and the application is single user. Online engineering need to go beyond : the subject of study is distant, interaction with it is compulsory, and collaboration with peers leads to a better metaphor of laboratories on the Web.

In this work, we answer the questions raised in Chapter 2 with a framework for online experimentations that :

- enables the remote control of devices.
- handles collaboration and group awareness.
- adapts its HCI depending on the usage procedure, thanks to Semantic Web technologies.
- improves the comprehension of manipulated remote scenes, thanks to mixed reality techniques.
- is interoperable, thanks to multimedia scene descriptors.
- is reusable.

The framework is composed of different components :

- The drivers, responsible for the dialogue with devices. This part is realized using IVI standards.
- The middleware, responsible for conveying messages from user side to the remote device. It is designed on top of MOM features.
- The notification center, responsible for pulling back interface messages from users to others. This part is realized using WebSockets technologies.
- The interactive multimedia, presenting the interface to the end users. This component is built with MPEG-4 BIFS features and mixed reality techniques.
- The whisper module, enabling the awareness of usage procedures and the adaptation of the multimedia interface.
- An authoring tool, enabling the creation of the multimedia content for new online devices.

This framework is serialized in the form of an open source project, called OCELOT. This project is hosted by the OW2 forge and is distributed in the LGPL license. Its evaluation has been introduced with remote practical exercises conducted in the UFL in Tripoli, Lebanon. Fifteen students manipulated the device hosted in Saint-Étienne during the session.

The next section discusses the advantages and limitations of our proposition.

5.2 Discussions

We presented how a complete solution for collaborative remote laboratories could be achieved using mixed reality techniques and interactive multimedia based on MPEG-4 BIFS. We also discuss the limitations and advantages of such an approach.

5.2.1 Remote Laboratories Frameworks

As presented in the state of the art, most of Remote Laboratories frameworks and HCs are built on top of technologies such as Java, Flash, LabView, etc. In our work, we propose to build interfaces using MPEG-4 BIFS, enabling to easily bring mixed reality and ensuring interoperability. Given the feedbacks from the first tests conducted, we can say that our approach gives a real sensation of manipulating the remote device. The mixed reality paradigm allows a high fidelity of the representation of the device to manipulate. Our approach gives a high level of interoperability since every terminal capable of displaying multimedia content can be used in order to control the remote device. The MPEG-4 BIFS standard is not bounded to any specific platform or operating system and proposes an efficient compression schema of the scene, up to 3 times according to our tests on different devices.

The middleware and the multimedia client allow to use the framework not only with remote laboratories but with any kind of collaborative multimedia applications, such as IPTV services, video conferencing, etc.

5.2.2 Multimodality.

Related works on multimodality highlight the fact that it can enhance cognitive and learning processes and increase learner immersion. Our approach relies on the learner's terminal to interact with the remote device, i.e. keyboard and mouse for a classical computer, a touchscreen for smartphones or tablets, etc. and could benefit from multimodal interactions. But it could also decrease the understanding of the manipulated device by adding extra modality, such as voice control on a spectrum analyzer. When it comes to remote laboratories, we strongly believe that the interactions on the remote device should be as close as possible to the ones on the *in-situ* device.

5.2.3 Collaboration.

While iLab does not provide collaboration within its framework, collaboration and group awareness can be achieved using the framework we propose, even if the communication tools are not used when all the users are located in the same room.

Social networks could enforce the collaboration processes. Our system provides communication tools for people connected and interacting with the same device but does not provide information on users such as location, interests, etc. One possible scenario for improving this point could be the integration in a Learning Management System such as Moodle, which provides community tools for learning. The integration of Semantic Web technologies, especially the Friend of a Friend (FOAF)¹ project could also be a track search since its goal is to describe "people, the links between them and the things they create and do".

5.2.4 Limitations.

However, our approach has some limitations. On the instrument side, if one wants to create a remote laboratory with a not IVI-compliant device, there is a non-negligible development time to consider. This problem could be addressed with driver introspection for example. Considering students feedbacks, the perceived video quality from the IP camera is not sufficient. This problem could be achieved using an IP camera with better quality and also by adapting the image resolution depending on the available bandwidth. This work will be a part of our future research on the system. The current developments on the framework allow only to build remote laboratories for devices with a 2D front, such as measurement instruments (oscilloscope, spectrum or vector network analyzer, etc.). Since MPEG-4 BIFS scenes can embed 3D scenes and graphics, we plan on supporting 3D instruments in a future version of the framework.

1. <http://www.foaf-project.org/>

5.3 Perspectives

Perspectives for the work conducted in this thesis can be sought from the broader viewpoint of online mixed reality. We presented here an architecture for online engineering using mixed reality techniques and interactive multimedia. The framework resulting from our research could be extended to different use cases, such as collaborative IPTV services, collaborative remote surgery, etc. This would entail deeper works on 3D scenes on the client side.

Since the framework supports collaboration, deeper research on social interactions could be led. In particular, the FOAF project for managing users and operators could also be a track search since its goal is to describe "people, the links between them and the things they create and do". Learning Management System (LMS), such as Moodle for instance, are systems built in order to assist anyone involved in a learning process in its management of learning paths. In e-learning situations, it could improve the pedagogical objectives of the courses.

Current researches in the context-aware system area could also bring improvements to the framework. For now, the client application is the same on every kind of terminal. We could imagine adapting the multimedia user interface depending on different criteria such as the terminal, available bandwidth, geo-location, etc.

Of course current research on streaming over HTTP are also trends and could improve the framework. The goal of streaming over HTTP is to propose technologies that use the natural opened port for internet connections, as opposed to other technologies such as RTSP that uses by default the 554 port. Such ports are also often blocked by routers, firewalls or proxies. The basic idea behind HTTP streaming is to split files into segments and present these segments in a playlist file. HTTP streaming protocols can be used using HTTP Live Streaming from Apple or Dynamic Adaptive Streaming over HTTP (DASH) from the MPEG committee.

Beware of bugs in the above code; I have only proved it correct, not tried it.

Donald Knuth

6

Spectrum Analyzer implementation : FREQ mode

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Ivi.SpecAn.Interop;
using RohdeSchwarz.RsSpecAn.Interop;

namespace IVISpectrumAnalyzer
{
    class IVIFreq
    {
        public IIVI_SpecAn_Frequency _freq;

        /*
         * Constructor
         */
        public IVIFreq(IRsSpecAn rs)
        {
            IIVI_SpecAn specan = (IIVI_SpecAn)rs;
            _freq = specan.Frequency;
        }
    }
}
```

```
/*
 * getStartFreq
 */
public double getStartFreq()
{
    return _freq.Start;
}

/*
 * getStopFreq
 */
public double getStopFreq()
{
    return _freq.Stop;
}

/*
 * getFreqOffset
 */
public double getFreqOffset()
{
    return _freq.Offset;
}

/*
 * setFreqOffset
 */
public double getFreqOffset(double offset)
{
    return _freq.Offset = offset;
}

/*
 * getCenterFreq
 */
public double getCenterFreq()
{
    return (_freq.Start+_freq.Stop)/2;
}

/*
 * setCenterFreq
 */
public double setCenterFreq(double center)
{
    double x = Math.Abs(getStopFreq() - 2*center);
    setStartFreq(x);
    double returnValue = (getStartFreq() + getStopFreq()) / 2;
}
```

```
        return returnValue;
    }

    /*
     * setStartFreq
     */
    public double setStartFreq(double start)
    {
        _freq.ConfigureStartStop(start, _freq.Stop);
        return _freq.Start;
    }

    /*
     * setStopFreq
     */
    public double setStopFreq(double stop)
    {
        _freq.ConfigureStartStop(_freq.Start, stop);
        return _freq.Stop;
    }
}
}
```


Rules of Optimization :

Rule 1 : *Don't do it*

Rule 2 (*for experts only*) : *Don't do it yet.*

Michael A. Jackson

7

Features and descriptors graphs

This annex shows the performance results of the extraction of the four feature descriptors BRIEF, ORB, SIFT and SURF for ten devices that are available at Télécom Saint-Étienne and in the LT2C laboratory.



FIGURE 7.1 – Agilent E4407B.

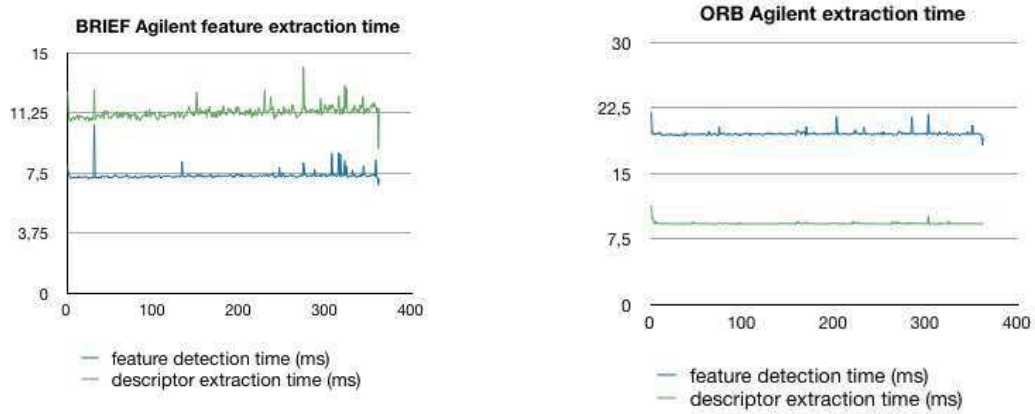


FIGURE 7.2 – Agilent E4407B : BRIEF and ORB extraction.

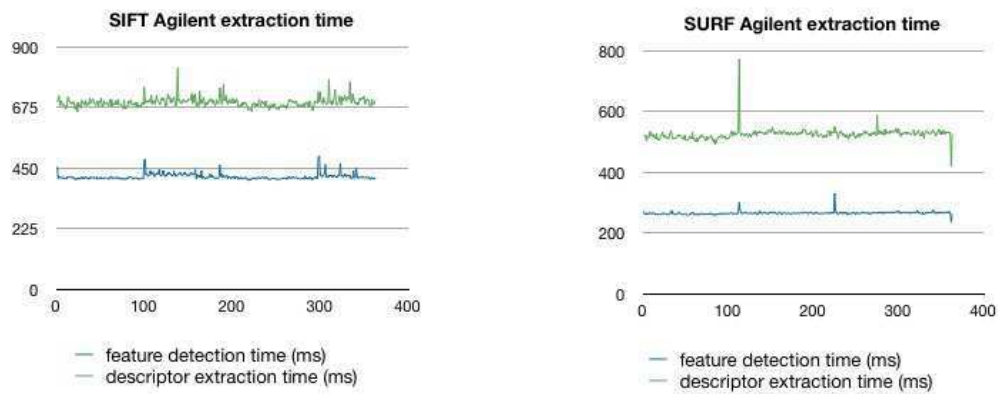


FIGURE 7.3 – Agilent E4407B : SIFT and SURF extraction.

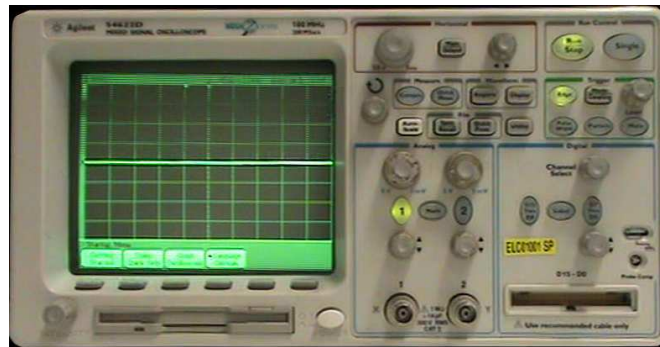


FIGURE 7.4 – Agilent 54622D.

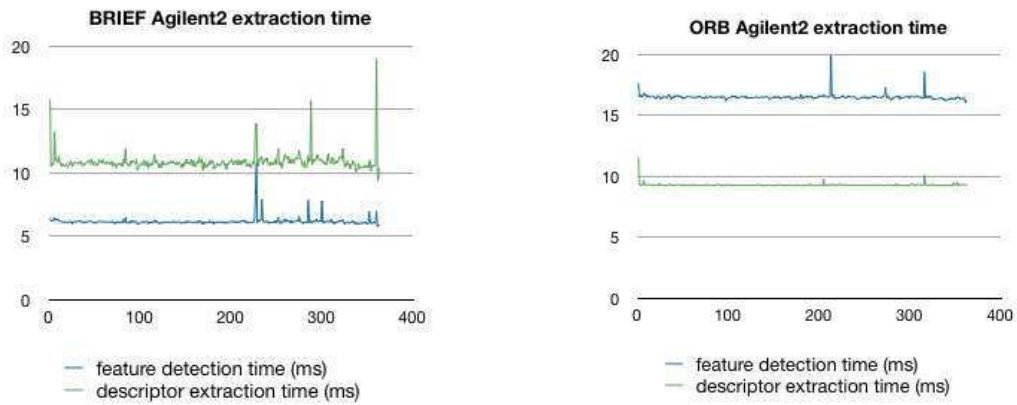


FIGURE 7.5 – Agilent 54622D : BRIEF and ORB extraction.

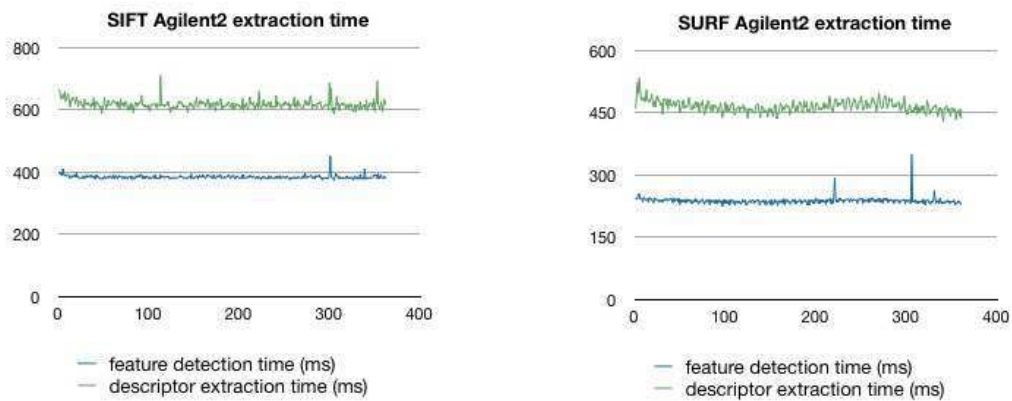


FIGURE 7.6 – Agilent 54622D : SIFT and SURF extraction.



FIGURE 7.7 – Anritsu 37397A.

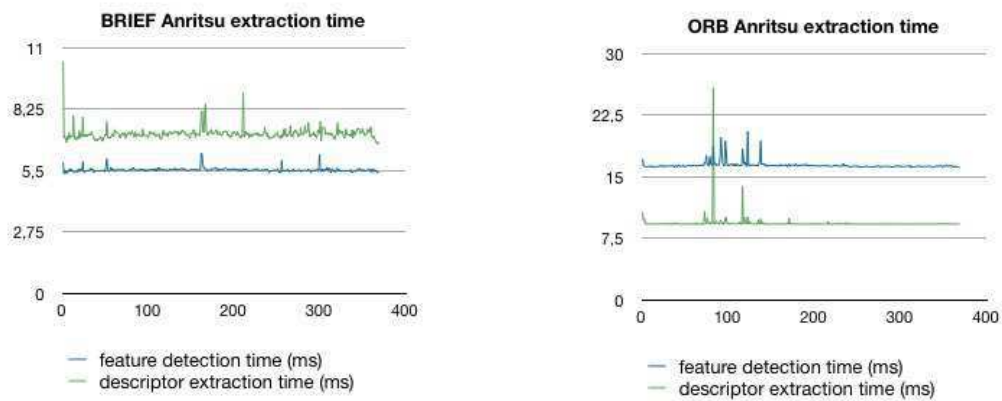


FIGURE 7.8 – Anritsu 37397A : BRIEF and ORB extraction.

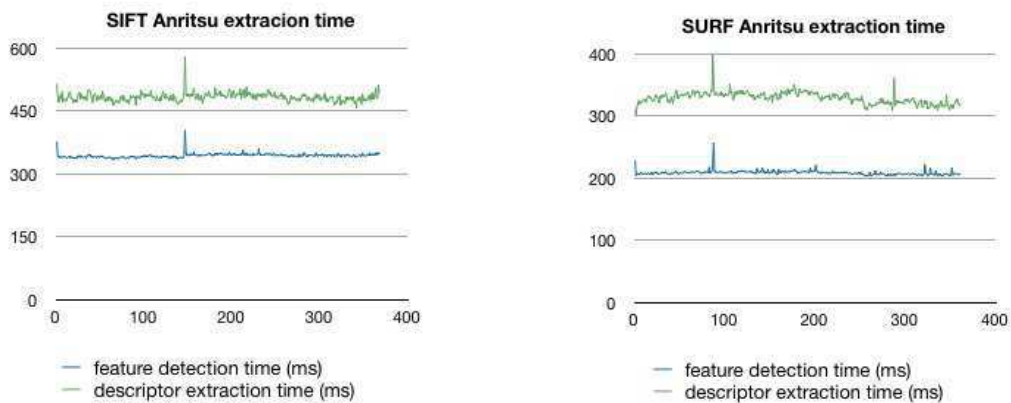


FIGURE 7.9 – Anritsu 37397A : SIFT and SURF extraction.

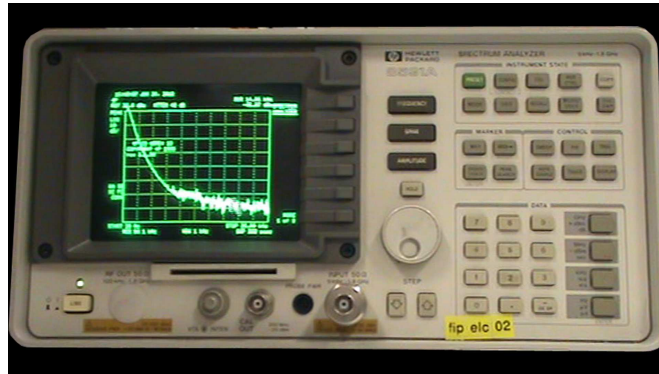


FIGURE 7.10 – HP 8561A.

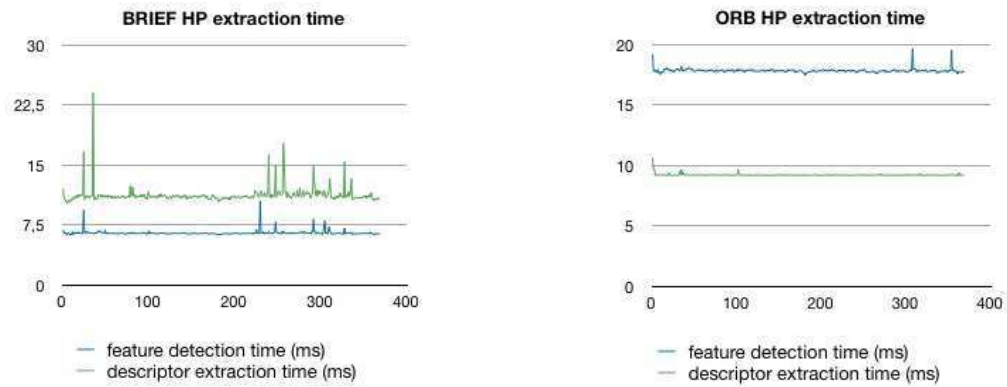


FIGURE 7.11 – HP 8561A : BRIEF and ORB extraction.

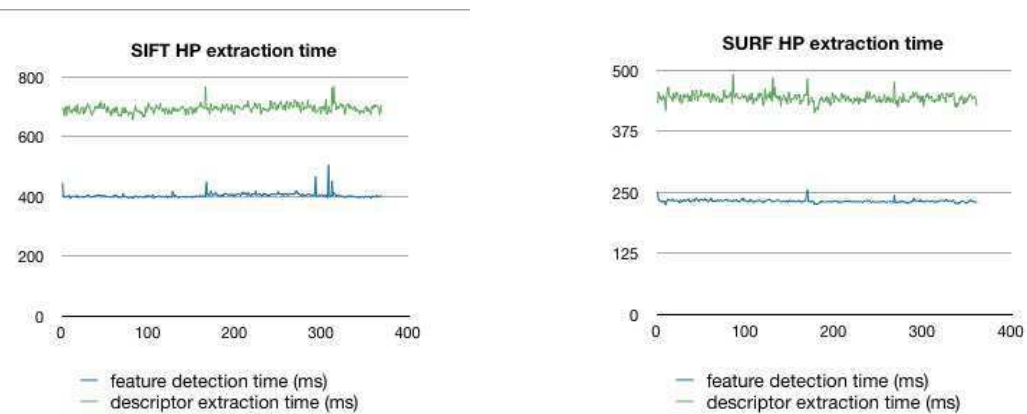


FIGURE 7.12 – HP 8561A : SIFT and SURF extraction.



FIGURE 7.13 – HP 54510A.

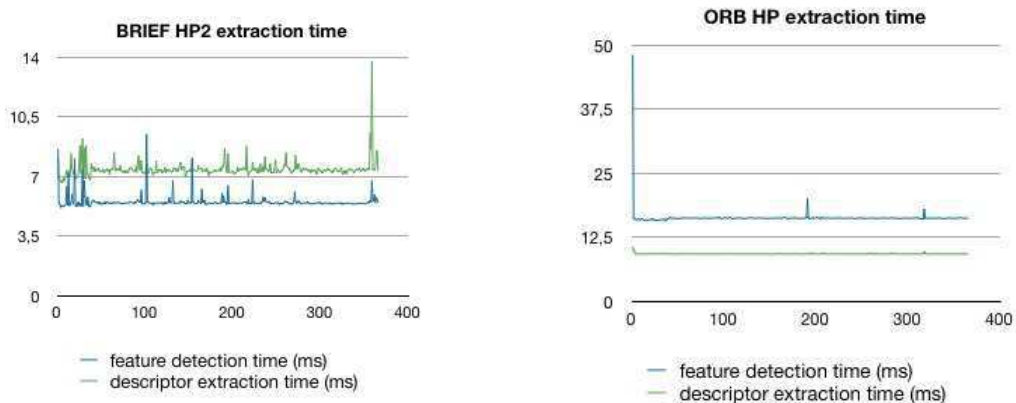


FIGURE 7.14 – HP 54510A : BRIEF and ORB extraction.

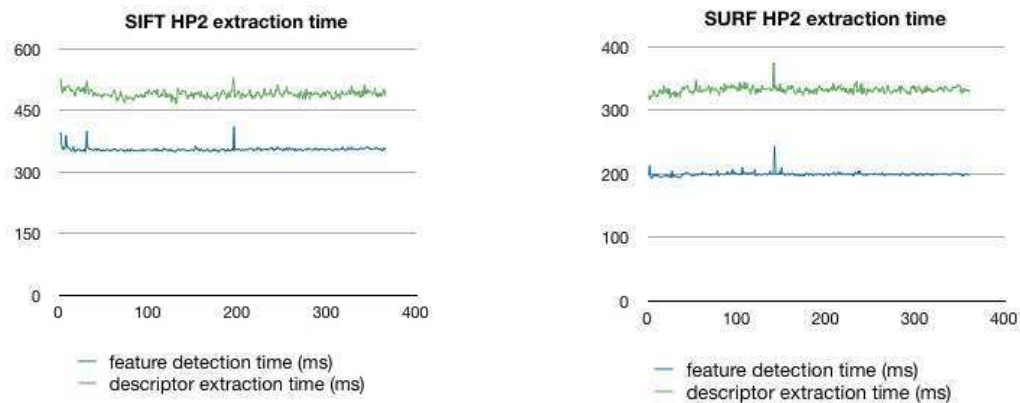


FIGURE 7.15 – HP 54510A : SIFT and SURF extraction.

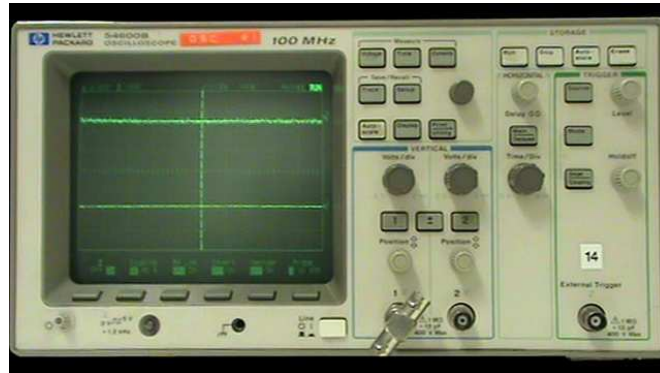


FIGURE 7.16 – HP 54600B.

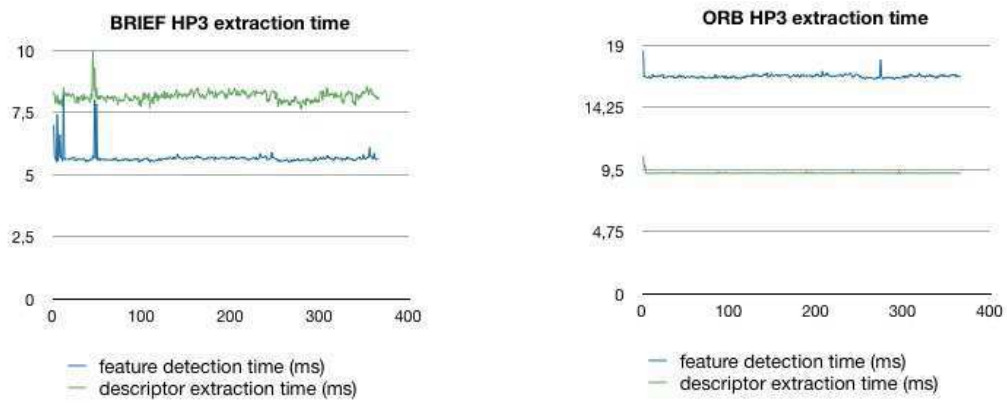


FIGURE 7.17 – HP 54600B : BRIEF and ORB extraction.

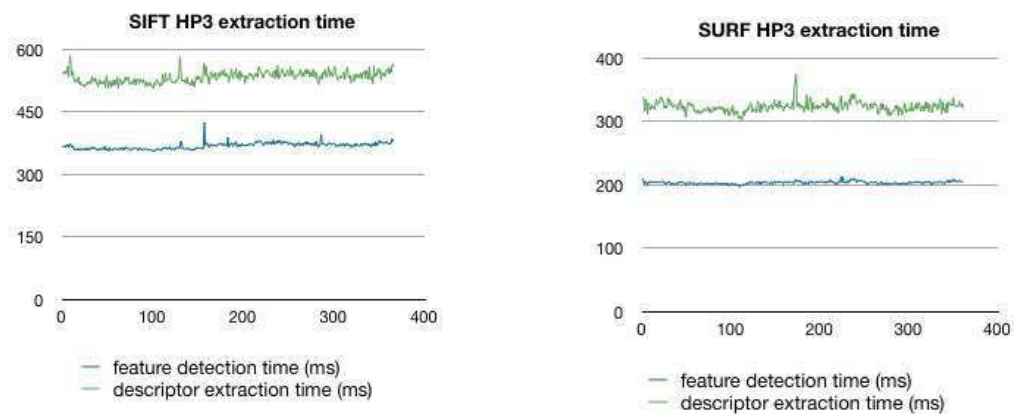


FIGURE 7.18 – HP 54600B : SIFT and SURF extraction.

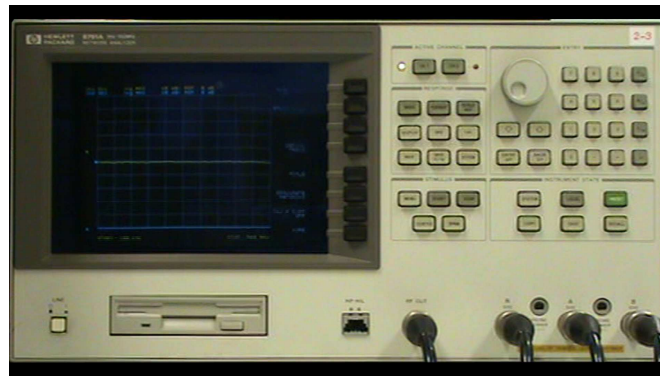


FIGURE 7.19 – HP E5060A.

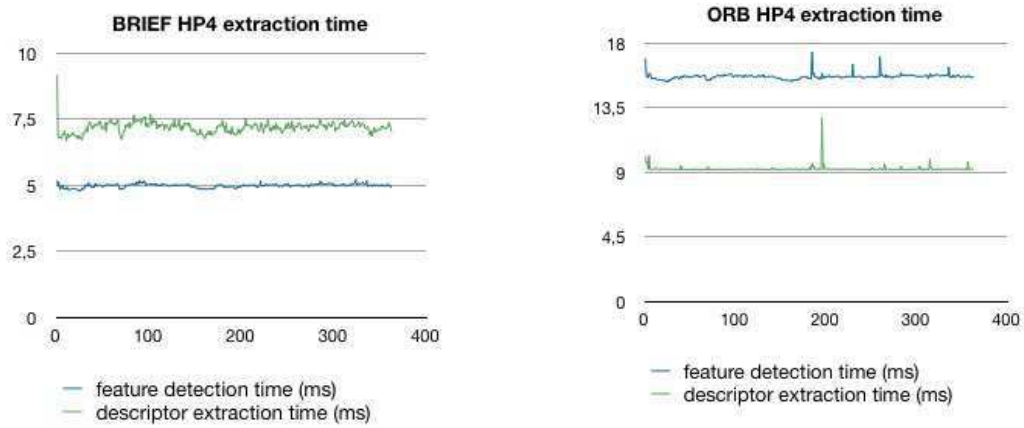


FIGURE 7.20 – HP E5060A : BRIEF and ORB extraction.

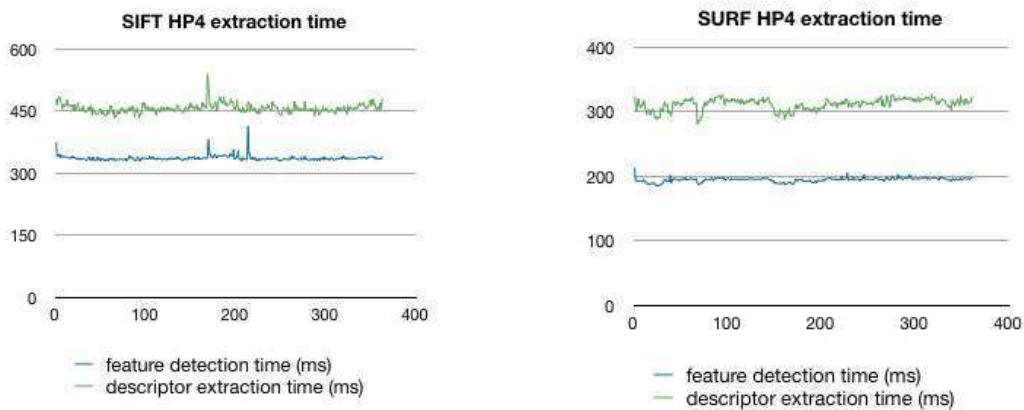


FIGURE 7.21 – HP E5060A : SIFT and SURF extraction.

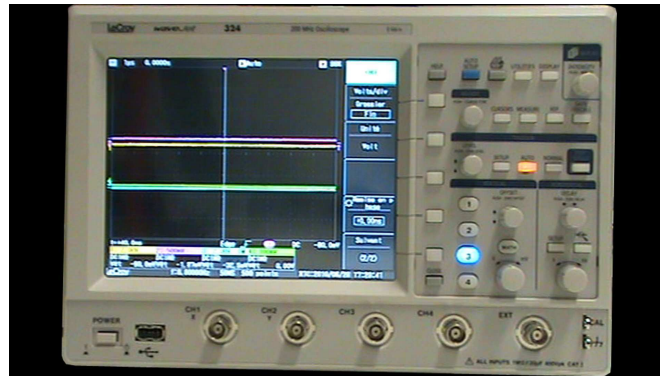


FIGURE 7.22 – LeCroy WaveJet 324A.

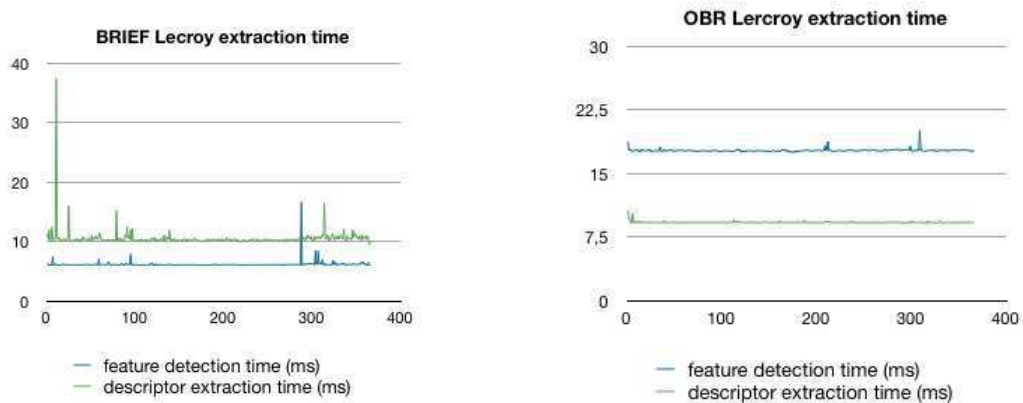


FIGURE 7.23 – LeCroy WaveJet 324A : BRIEF and ORB extraction.

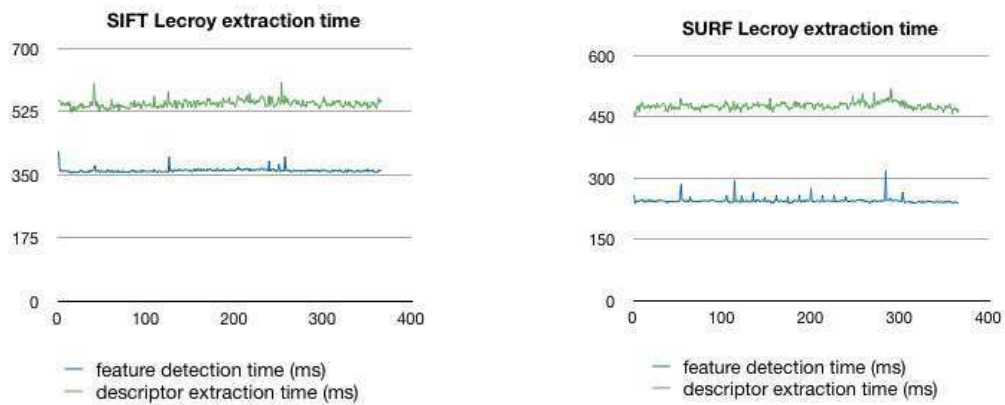


FIGURE 7.24 – LeCroy WaveJet 324A : SIFT and SURF extraction.

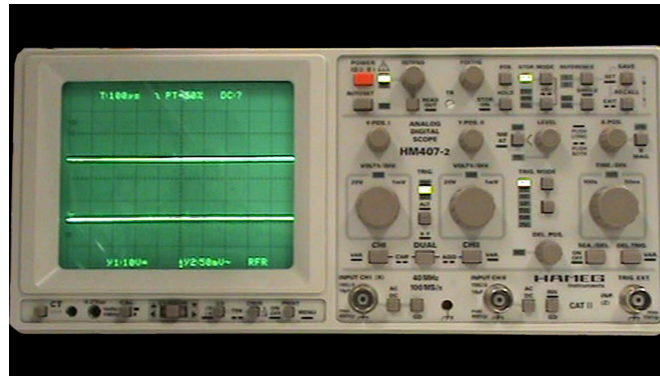


FIGURE 7.25 – Hameg HM407-2.

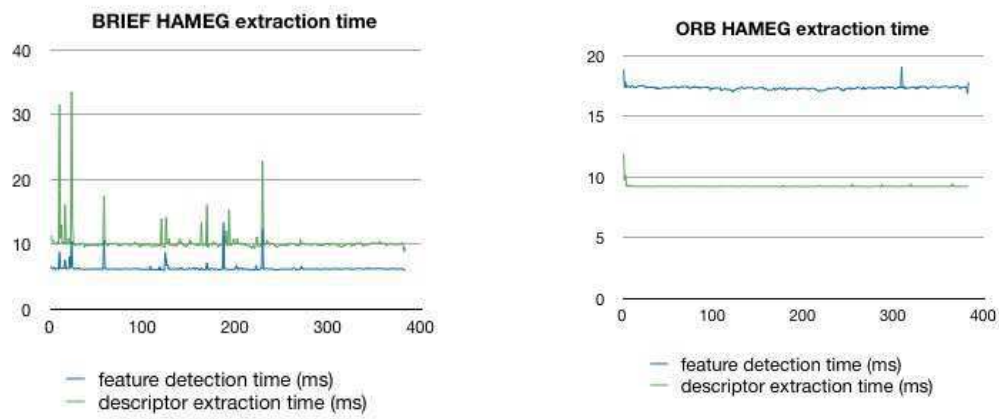


FIGURE 7.26 – Hameg HM407-2 : BRIEF and ORB extraction.

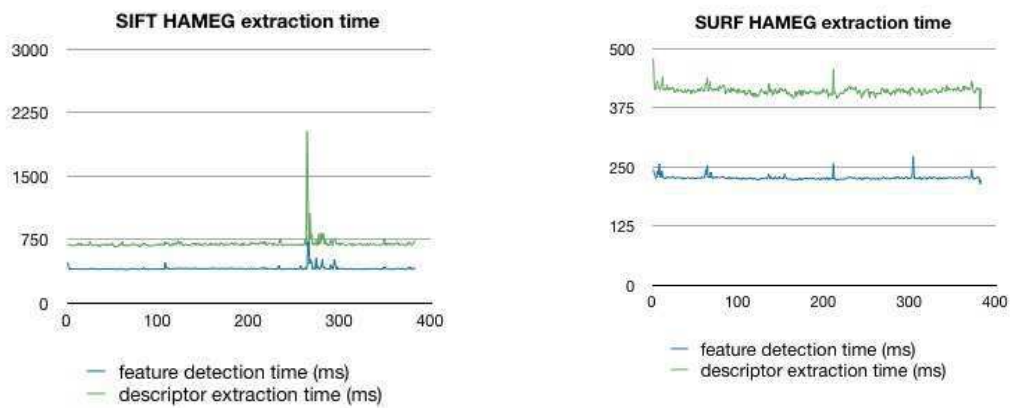


FIGURE 7.27 – Hameg HM407-2 : SIFT and SURF extraction.

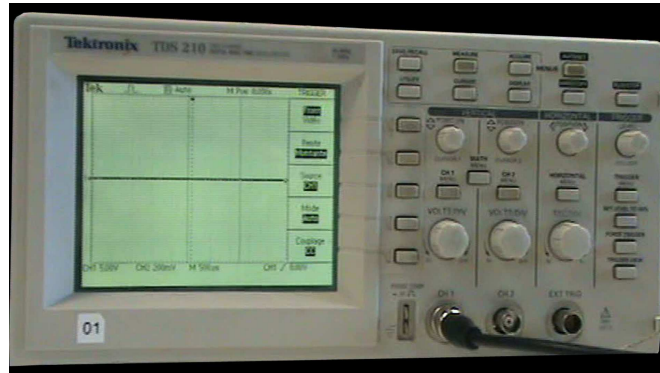


FIGURE 7.28 – Tektronix TDS 210.

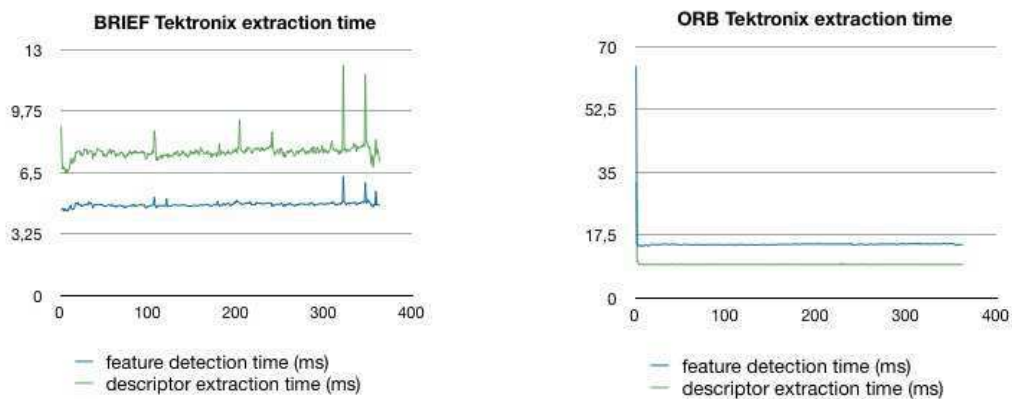


FIGURE 7.29 – Tektronix TDS 210 : BRIEF and ORB extraction.

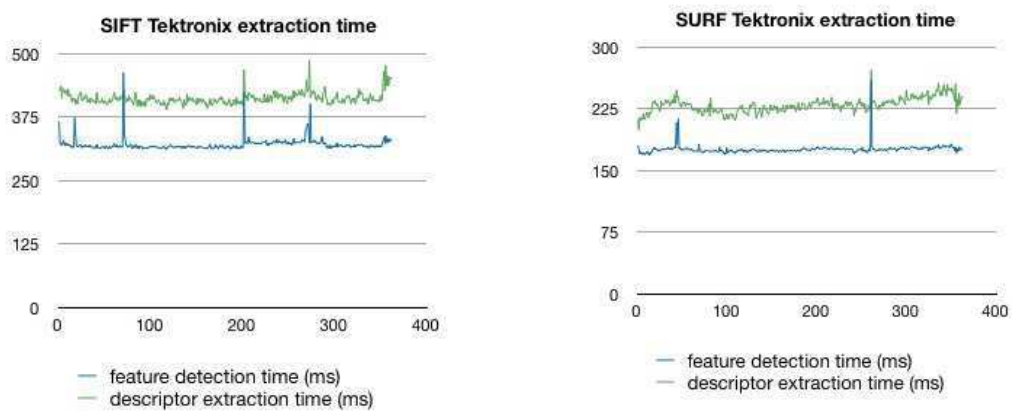


FIGURE 7.30 – Tektronix TDS 210 : SIFT and SURF extraction.

Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.

Rick Osborne

8

BF and FLANN matching graphs for SURF features descriptors

This annex shows the performance results of the brute force matching against the FLANN matching for SURF feature descriptors for ten devices that are available at Télécom Saint-Étienne and in the LT2C laboratory.



FIGURE 8.1 – Agilent E4407B.

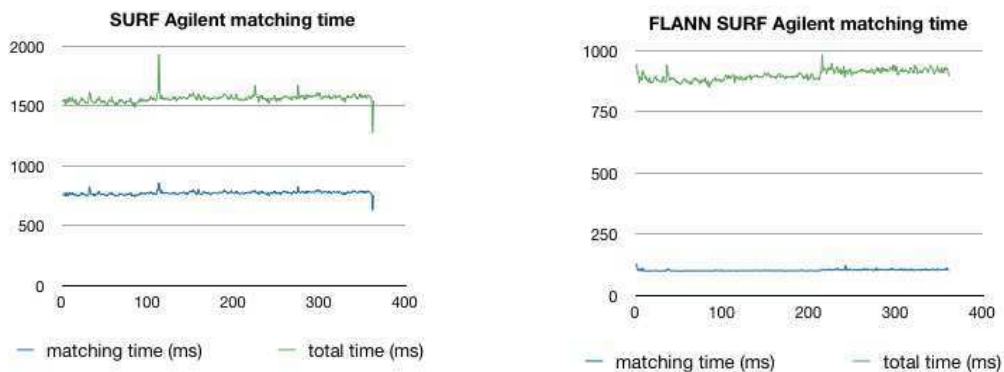


FIGURE 8.2 – Agilent E4407B : brute force and FLANN matching.

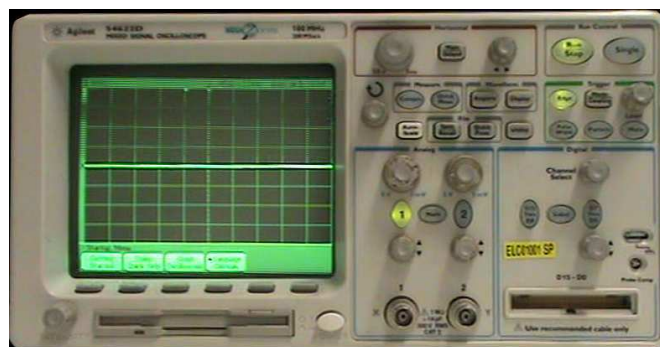


FIGURE 8.3 – Agilent 54622D.

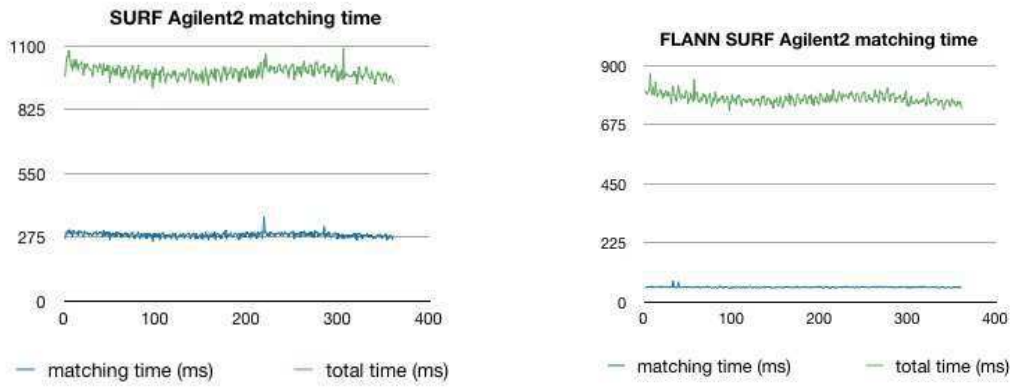


FIGURE 8.4 – Agilent 54622D : brute force and FLANN matching.



FIGURE 8.5 – Anritsu 37397A.

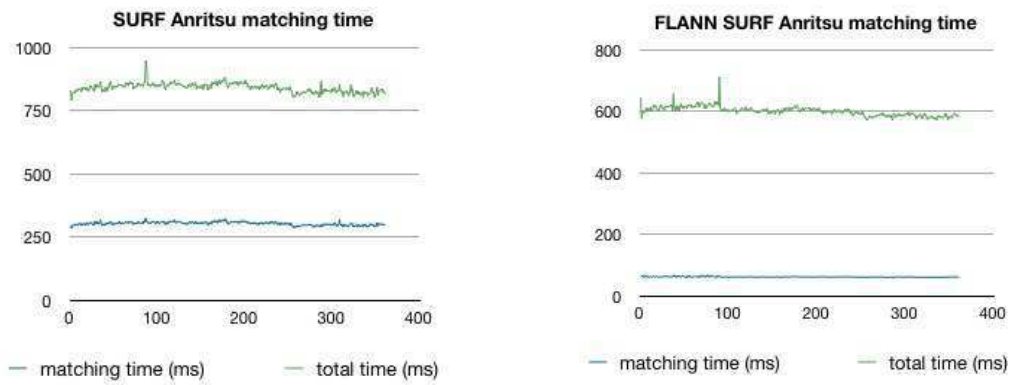


FIGURE 8.6 – Anritsu 37397A : brute force and FLANN matching.

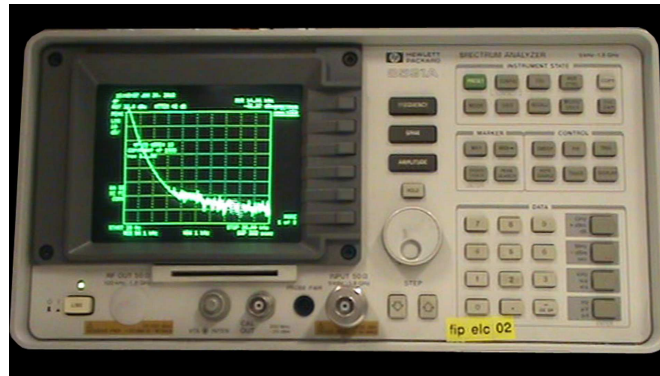


FIGURE 8.7 – HP 8561A.

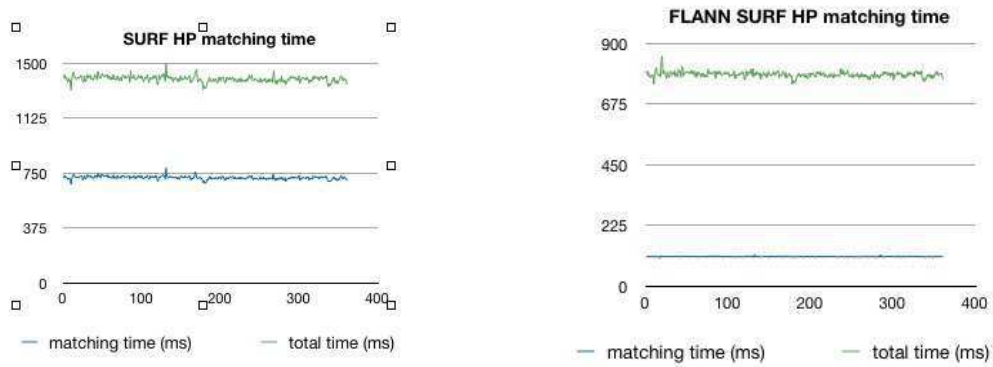


FIGURE 8.8 – HP 8561A : brute force and FLANN matching.



FIGURE 8.9 – HP 54510A.

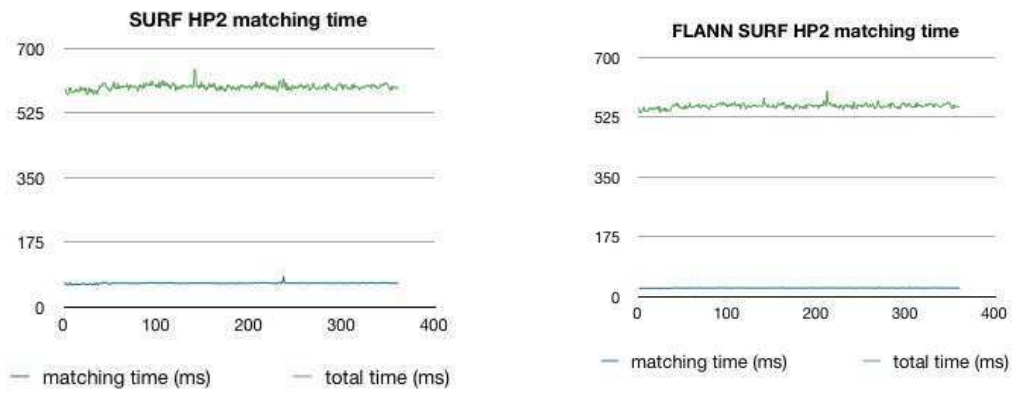


FIGURE 8.10 – HP 54510A : brute force and FLANN matching.

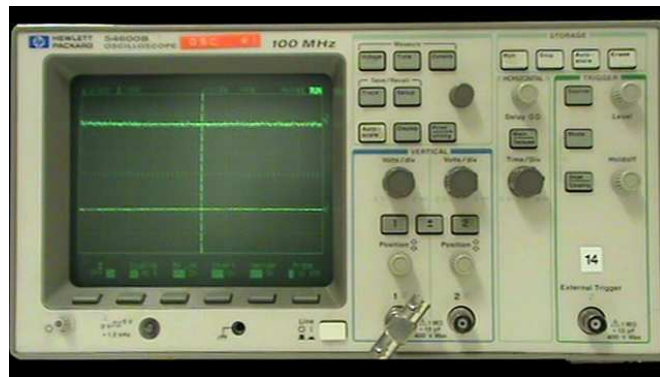


FIGURE 8.11 – HP 54600B.

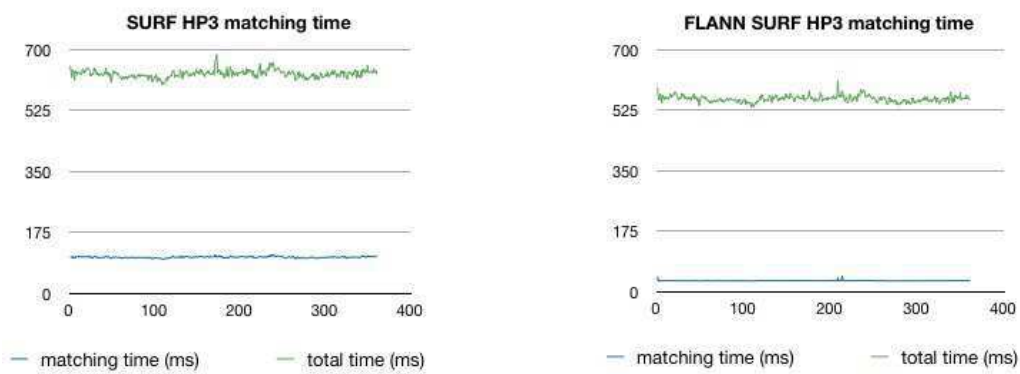


FIGURE 8.12 – HP 54600B : brute force and FLANN matching.

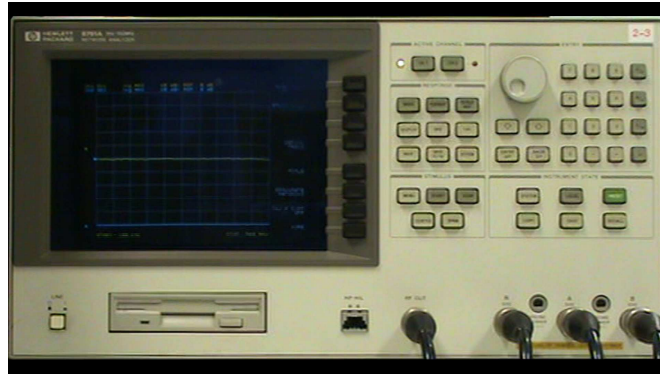


FIGURE 8.13 – HP E5060A.

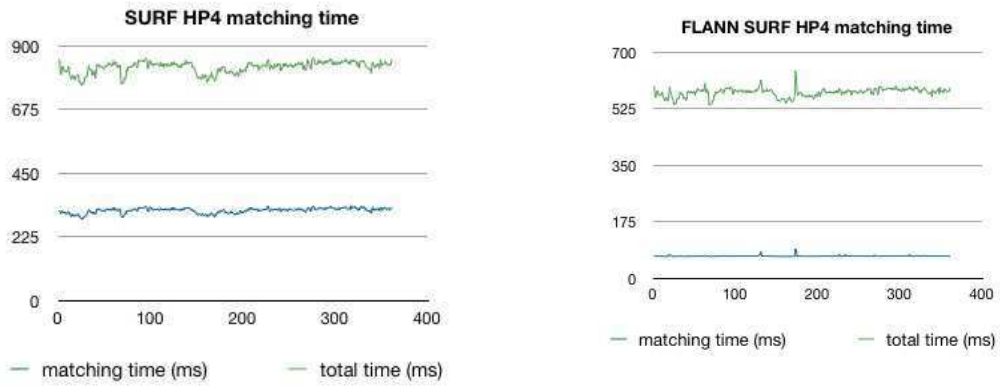


FIGURE 8.14 – HP E5060A : brute force and FLANN matching.

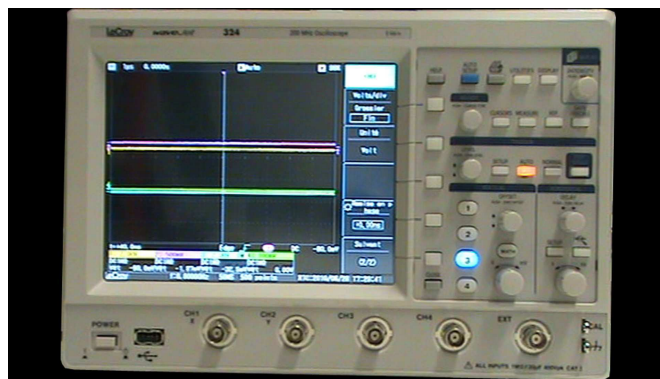


FIGURE 8.15 – LeCroy WaveJet 324A.

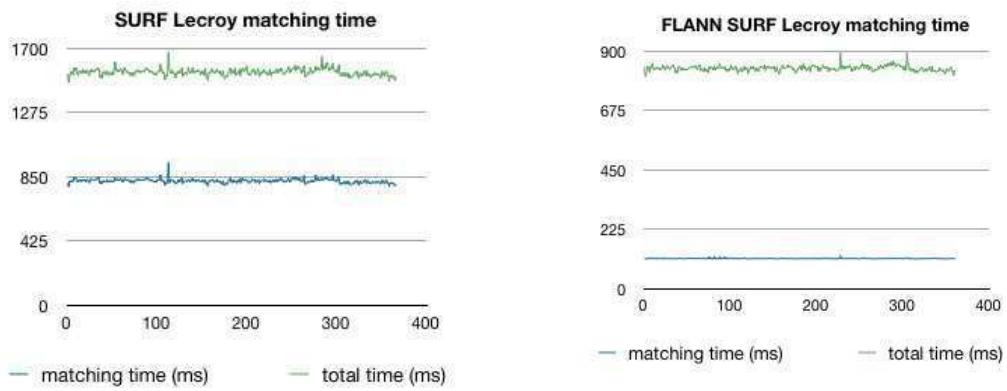


FIGURE 8.16 – LeCroy WaveJet 324A : brute force and FLANN matching.

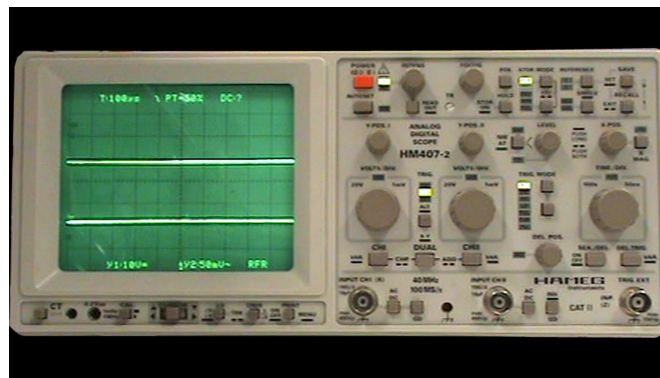


FIGURE 8.17 – Hameg HM407-2.

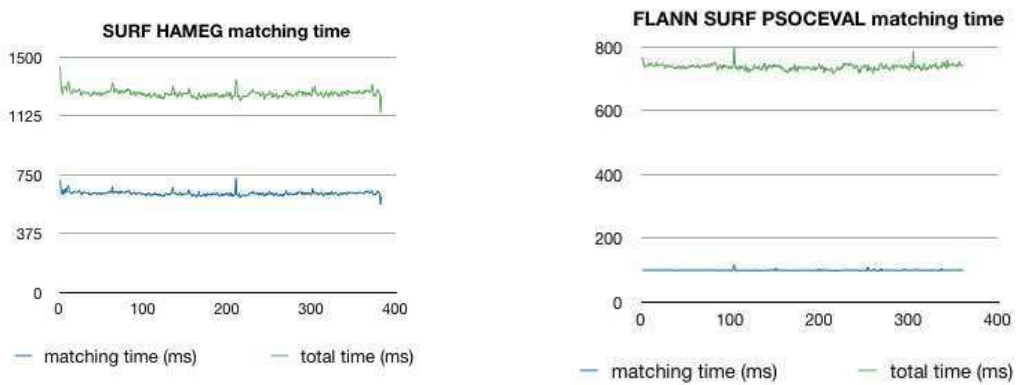


FIGURE 8.18 – Hameg HM407-2 : brute force and FLANN matching.

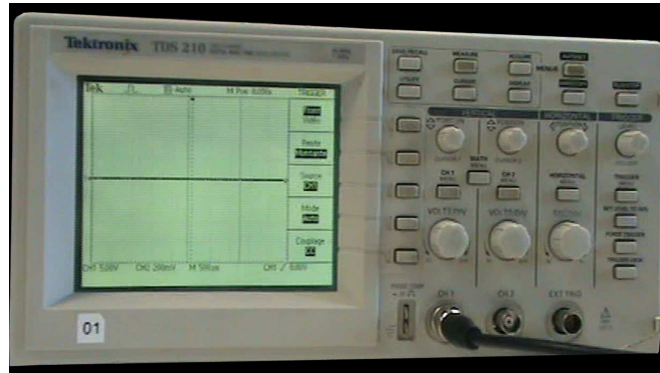


FIGURE 8.19 – Tektronix TDS 210.

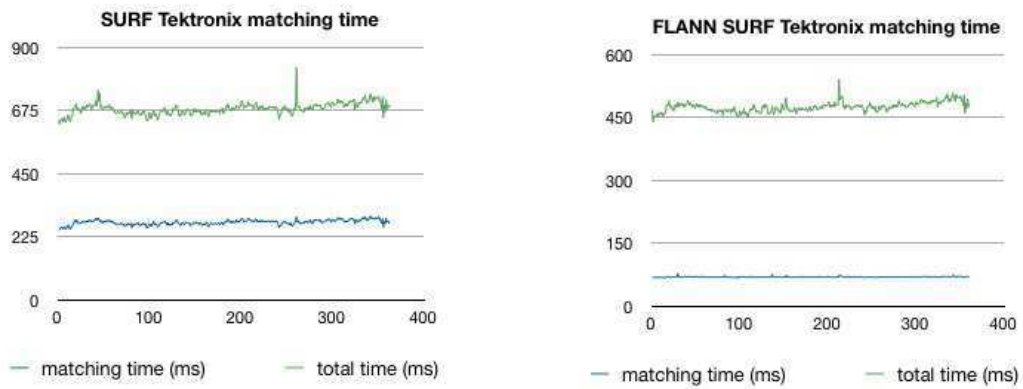


FIGURE 8.20 – Tektronix TDS 210 : brute force and FLANN matching.

Glossary

- AJAX** Asynchronous Javascript and XML. 3
- API** Application Programming Interface. 59, 99
- CAM** Computer-Assisted Maintenance. 5, 10
- CD-ROM** Compact Disc - Read Only Memory. 35
- Collaboratories** [a] center without walls, in which the nation's researchers can perform their research without regard to geographical location, interacting with colleagues, accessing instrumentation, sharing data and computational resources, and accessing information in digital libraries. 8
- COLLADA** COLLABorative Design Activity. 37
- CSCW** Computer Supported Collaborative Work. 5
- DoG** Difference of Gaussian. 23
- ES** Elementary Streams. 79
- GPS** Global Positioning Systems. 22
- HCI** Human Computer Interface. 4
- HTML** HyperText Markup Language. 38
- HTTP** Hypertext Transfer Protocol. 79
- HTTPS** HyperText Transfer Protocol Secure. 95
- IETF** Internet Engineering Task Force. 99
- IVI** Interchangeable Virtual Instrument. 59
- JEE** Java Enterprise Edition. 89
- JMS** Java Messaging Service. 72
- JOnAS** Java Open Application Server. 90
- JORAM** Java Open Reliable Asynchronous Messaging. 90
- JSON** JavaScript Object Notation. 73

-
- KB** Knowledge Base. 75
- LDAP** Lightweight Directory Access Protocol. 94
- MOM** Message-Oriented Middleware. 60
- MOS** Mean Opinion Scale. 114
- MPEG-4 BIFS** MPEG-4 BInary Format for Scene. 38
- MPEG-4 LAsER** MPEG-4 Lightweight Application Scene Representation. 39
- MPEG-4 XMT** MPEG-4 Textual format. 39
- NFC** Near Field Communication. 22
- OD** Object Descriptors. 79
- OOM** Object-Oriented Middleware. 60
- ORB** Oriented FAST and Rotated BRIEF. 26
- QoE** Quality of Experience. 114
- REST** REpresentation State Transfer. 105
- RFID** Radio Frequency IDentification. 22
- RIA** Rich Internet Application. 3
- RMI** Remote Method Invocation. 72
- RPC** Remote Procedure Control. 72
- RTP** Real-time Transport Protocol. 82
- RTSP** Real Time Streaming Protocol. 82
- SDP** Session Description Protocol. 82
- SIFT** Scale-Invariant Feature Transform. 24
- SMIL** Synchronized Multimedia Integration Language. 36
- SPARQL** SPARQL Protocol and RDF Query Language. 105
- SPARUL** SPARQL Update. 105
- SSL** Secure Sockets Layer. 95
- SURF** Speeded-Up Robust Feature. 24
- SVG** Scalable Vector Graphics. 36
- TCP** Transmission Control Protocol. 82
- UDP** User Datagram Protocol. 82
- ULF** Université Libano-Française. 113

URL Uniform Resource Locator. 79

VISA Virtual Instrument Software Architecture. 59

VRML Virtual Reality Modeling Language. 35

W3C World Wide Web Consortium. 99

Widget Contraction of window and gadget.. 95

X3D eXtensible 3D. 36

Bibliography

- [4rs06] 4rs. *4 Store Reasoning.*, 2006. 35
- [Adobe12] Adobe. *Flash Professional*, 2012. 38
- [Alusi97] G. H. Alusi, A. C. Tan, A. D. Linney, K. Raof & A. Wright. *Three dimensional tracking with ultrasound for augmented reality applications in skull base surgery*. In Proceedings of the First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Computer-Assisted Surgery, CVRMed-MRCAS '97, pages 511–517, London, UK, UK, 1997. Springer-Verlag. 29
- [Andujar11] Jose Manuel Andujar, Andrés Mejías & Marco Antonio Marquez. *Augmented Reality for the Improvement of Remote Laboratories : An Augmented Remote Laboratory*. IEEE Trans. Education, vol. 54, no. 3, pages 492–500, 2011. xi, 47, 52
- [Back10] Maribeth Back, Don Kimber, Eleanor Rieffel, Anthony Dunningan, Bee Liew, Sagar Gattepally, Jonathan Foote, Jun Shingu & James Vaughan. *The virtual chocolate factory : mixed reality industrial collaboration and control*. In Proceedings of the international conference on Multimedia, MM '10, pages 1505–1506, New York, NY, USA, 2010. ACM. 48
- [Bafoutsou02] Georgia Bafoutsou & Gregoris Mentzas. *Review and functional classification of collaborative systems*. Int. J. Inf. Manag., vol. 22, no. 4, pages 281–305, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, August 2002. 31
- [Bay08] Herbert Bay, Andreas Ess, Tinne Tuytelaars & Luc Van Gool. *Speeded-Up Robust Features (SURF)*. Comput. Vis. Image Underst., vol. 110, no. 3, pages 346–359, Elsevier Science Inc., New York, NY, USA, June 2008. 24
- [Bazire05] Mary Bazire & Patrick Brézillon. *Understanding Context Before Using It*. In 5th International and Interdisciplinary Conference on Modeling and Using Context, pages 29 – 40, 2005. 32

- [Bernstein96] Philip A. Bernstein. *Middleware : a model for distributed system services*. Commun. ACM, vol. 39, no. 2, pages 86–98, ACM, New York, NY, USA, February 1996. 8, 62
- [Bohringer09] David Bohringer, Sabina Jeschke & Thomas Richter. *LiLa : A European Project on Networked Experiments*. In Proceedings of the 6th International Conference on Remote Engineering and Virtual Instrumentation. Kassel University Press, Bridgeport, CT, June 2009. 47
- [Bottecchia08] Sébastien Bottecchia, Jean-Marc Cieutat & Jean-Pierre Jessel. *T.A.C : Système de réalité augmentée pour la Télé-Assistance Collaborative dans le domaine de la maintenance*. In Association Française de Réalité Virtuelle, page 133, Bordeaux, France, October 2008. 48, 49
- [Boudoin08] Pierre Boudoin, Christophe Domingues, Samir Otmame, Nasima Ouramdane & Malik Mallem. *Towards multimodal human-robot interaction in large scale virtual environment*. In Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction, HRI '08, pages 359–366, New York, NY, USA, 2008. ACM. 49
- [Bruns05] Friedrich Wilhelm Bruns, Heinz-Hermann Erbe & Martin Faust. *Engineering Future Laboratories*. In Dieter Muller, editeur, impuls : MARVEL Mechatronics Training in Real and Virtual Environments, Vol. 18, pages 83–91, 2005. (conference, reviewed). 46
- [Buschiazzo09] P. Buschiazzo & A. M. Scapolla. *A service oriented framework to set up flexible and distributed data acquisition and control systems*. In Proceedings of the International Conference on Management of Emergent Digital EcoSystems, MEDES '09, pages 85 :510–85 :515, New York, NY, USA, 2009. ACM. 46
- [Buschiazzo10] Paolo Buschiazzo, Davide Leoncini, Rodolfo Zunino & Anna Marina Scapolla. *A Web-Based Laboratory for Digital Signal Processing*. iJOE, vol. 6, no. S1, pages 6–11, 2010. 46
- [Callaghan07] Michael J. Callaghan, Jim Harkin, T. Martin McGinnity & Liam P. Maguire. *Paradigms in Remote Experimentation*. iJOE, vol. 3, no. 4, 2007. 47
- [Calonder10] M. Calonder, V. Lepetit, C. Strecha & P. Fua. *BRIEF : Binary Robust Independent Elementary Features*. In European Conference on Computer Vision, September 2010. 24
- [Calonder11] Michael Calonder, Vincent Lepetit, Mustafa Ozuysal, Tomasz Trzcinski, Christoph Strecha & Pascal Fua. *BRIEF : Computing*

- a Local Binary Descriptor very Fast*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 99, no. PrePrints, IEEE Computer Society, Los Alamitos, CA, USA, 2011. vii, 25
- [Cerf93] Vinton G. Cerf. National laboratories : Applying information technology for scientific research. The National Academies Press, 1993. 8
- [Chen11a] Alfred Chen, Chen-Yuan Kao, Ying-Hsiu Chen & Wen-Cheng Wang. *Establishing a Museum Display Platform by Using Combination of Reflection Holograms and Tangible Augmented Reality*. Transactions on Computational Science, vol. 13, pages 157–168, 2011. 28
- [Chen11b] Ruei-Jia Chen, Hung-Wei Lin, Yeong-Hwa Chang, Chieh-Tsai Wu & Shih-Tseng Lee. *Development of an Augmented Reality Force Feedback Virtual Surgery Training Platform*. International Journal of Automation and Smart Technology, pages 1–11, Ausmt, 2011. 29
- [Cheok04] Adrian David Cheok, Kok Hwee Goh, Wei Liu, Farzam Farbiz, Siew Wan Fong, Sze Lee Teo, Yu Li & Xubo Yang. *Human Pac-man : a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing*. Personal Ubiquitous Comput., vol. 8, no. 2, pages 71–81, Springer-Verlag, London, UK, UK, May 2004. vii, 28
- [Chien10] Jong-Chih Chien, Hoang-Yang Lu, Yi-Sheng Wu & Li-Chang Liu. *ARToolkit-based augmented reality system with integrated 1-D barcode : combining colorful markers with remote servers of 3D data for product promotion purposes*. In Proceedings of the Second international conference on Computational collective intelligence : technologies and applications - Volume Part III, ICC-CI'10, pages 200–209, Berlin, Heidelberg, 2010. Springer-Verlag. 27
- [Cho99] Youngkwan Cho, Jongweon Lee & Ulrich Neumann. *A multi-ring fiducial system and an intensity-invariant detection method for scalable augmented reality*. In Proceedings of the international workshop on Augmented reality : placing artificial objects in real scenes : placing artificial objects in real scenes, IWAR '98, pages 147–165, Natick, MA, USA, 1999. A. K. Peters, Ltd. 22
- [Concolato07] Cyril Concolato. *Descriptions de scènes multimédia : représentations et optimisations*. In Doctoral Dissertation, Ecole Nationale Supérieure des Télécommunications, Paris, France. July 2007. viii, 36

- [Concolato09] Cyril Concolato, Giovanni Cordara & Kyungmo Park. *Overview of the ISO/IEC 23007 MPEG-U (Rich Media User Interface).*, 2009. 102
- [Cooper02] Martyn Cooper & Alexis Donnelly. *Remote Controlled Experiments for Teaching over the Internet : A Comparison of Approaches Developed in the PEARL Project.* In 19th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education, pages 119–128, 2002. 45
- [Daniel11] Daniel Burnett Daniel & Paul Coulton. *Using Infra-Red Beacons as Unobtrusive Markers for Mobile Augmented Reality.* In 1st Workshop on Mobile Augmented Reality : Design Issues and Opportunities (Mobile HCI 2011), 2011. 22
- [Deaky12] Bogdan-Alexandru Deaky, Danilo Garbi Zutin & Philip H. Bailey. *The First Android Client Application for the iLab Shared Architecture.* iJOE, vol. 8, no. 1, pages 4–7, 2012. 43
- [Dey00] A.K. Dey & G.D. Abowd. *Towards a better understanding of context and context-awareness.* In CHI 2000 workshop on the what, who, where, when, and how of context-awareness, volume 4, pages 1–6. Citeseer, 2000. viii, 32
- [Didier05] J.-Y. Didier, D. Roussel, M. Mallem, S. Otmane, S. Naudet, Q.-C. Pham, S. Bourgeois, C. Mégard, C. Leroux & A. Hocquard. *AMRA : Augmented Reality assistance in train maintenance tasks.* In Workshop on Industrial Augmented Reality (ISMAR'05), Vienna (Austria), October 5th 2005. 49
- [Drummond06] Nicholas Drummond, Alan Rector, Robert Stevens, Georgina Moulton, Matthew Horridge, Hai Wang & Julian Sedenberg. *Putting OWL in Order : Patterns for sequences in OWL.* Athens Georgia, 2006. 79
- [Dufourd05] Jean-Claude Dufourd, Olivier Avaro & Cyril Concolato. *An MPEG Standard for Rich Media Services.* IEEE MultiMedia, vol. 12, no. 4, pages 60–68, IEEE Computer Society Press, Los Alamitos, CA, USA, October 2005. 40
- [Erifiu11] Alexander Erifiu & Gerald Ostermayer. *Hardware Sensor Aspects in Mobile Augmented Reality.* In EUROCAST (2), pages 536–543, 2011. 23
- [Eugster03] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui & Anne-Marie Kermarrec. *The many faces of publish/subscribe.* ACM Comput. Surv., vol. 35, no. 2, pages 114–131, ACM, New York, NY, USA, June 2003. xvii, 74

- [Fayolle98] Jacques Fayolle, Christophe Ducottet & Jean-Paul Schon. *Application of multiscale characterization of edges to motion determination*. IEEE Transactions on Signal Processing, vol. 46, no. 4, pages 1174–1178, 1998. 24
- [Fayolle00] Jacques Fayolle, Laurence Riou & Christophe Ducottet. *Robustness of a multiscale scheme of feature points detection*. Pattern Recognition, vol. 33, no. 9, pages 1437–1453, 2000. 23
- [Fayolle10] J. Fayolle, C. Gravier & B. Jailly. *Collaborative remote laboratory in virtual world*. In Proceedings of the 10th WSEAS international conference on applied informatics and communications, and 3rd WSEAS international conference on Biomedical electronics and biomedical informatics, AIC'10/BEBI'10, pages 396–402, Stevens Point, Wisconsin, USA, 2010. World Scientific and Engineering Academy and Society (WSEAS). 73
- [Fischler81] Martin A. Fischler & Robert C. Bolles. *Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography*. Commun. ACM, vol. 24, no. 6, pages 381–395, ACM, New York, NY, USA, June 1981. 70
- [Forgy82] Charles Forgy. *Rete : A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem*. Artificial Intelligences, vol. 19, no. 1, pages 17–37, 1982. 35
- [Franks97] J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink & L. Stewart. *An Extension to HTTP : Digest Access Authentication*, 1997. 96
- [Friz98] Harald Friz. *Design of an Augmented Reality User Interface for an Internet based Telerobot using Multiple Monoscopic Views*. In Diploma Thesis, Institute for Process and Production Control Techniques, Technical University of Clausthal Clausthal-Zellerfeld, Germany. September 1998. 44
- [Gelenbe03] Erol Gelenbe, Khaled Hussain & Varol Kaptan. *Enabling Simulation with Augmented Reality*. In MASCOTS Tutorials, pages 290–310, 2003. 29
- [Gillet11] Denis Gillet. *Remote Laboratory at EPFL, 10 years of exploitation*, 2011. 46
- [Gravier06] Christophe Gravier & Jacques Fayolle. *E-Instrumentation : A Collaborative and Adaptive Intelligent HCI for Remote Control of Devices*. In ICEIS (5), pages 87–93, 2006. 73
- [Gravier08] Christophe Gravier, Jacques Fayolle & Bernard Bayard. *Coping with collaborative and competitive episodes within collaborative remote laboratories*. In Proceedings of the International Conference

- on Remote Engineering and Virtual Instrumentation (REV'08), page 5, Duesseldorf, Allemagne, June 2008. Conseil Général de la Loire. 76
- [Gravier11] Christophe Gravier, Martin O'Connor, Jacques Fayolle & Jeremy Lardon. *Adaptive System for Collaborative Online Laboratories*. IEEE Intelligent Systems, vol. 99, no. PrePrints, IEEE Computer Society, Los Alamitos, CA, USA, 2011. 60, 73
- [Gros09] Benjamin Grosf, Mike Dean & Michael Kifer. *The silk system : Scalable higher-order defeasible rules*. In 3rd International Rule Challenge at RuleML, page 2, Las Vegas, Nevada, USA, 2009. 110
- [Gutwin04] Carl Gutwin, Reagan Penner & Kevin Schneider. *Group Awareness in Distributed Software Development*. In Proceedings of the ACM Conference on Computer-Supported Cooperative Work, 2004. xvii, 30, 54, 73
- [Halaschek-Wiener06] Christian Halaschek-Wiener, Bijan Parsia & Evren Sirin. *Description Logic Reasoning with Syntactic Updates*. In OTM Conferences (1), pages 722–737, 2006. 110, 111
- [Harris88] C. Harris & M. Stephens. *A combined corner and edge detector*. In Proc. Fourth Alvey Vision Conference, pages 147–151, 1988. vii, 23
- [Harward04] VJ Harward, J Del Alamo, V Choudhary, K DeLong, J L Hardison, S R Lerman, J Northbridge, D Talavera, C Varadharajan, S Wang, K Yehia & D Zych. *iLab : A scalable architecture for sharing online experiments*. International, 2004. ix, 9, 43
- [Henderson11] Steven Henderson & Steven Feiner. *Exploring the benefits of augmented reality documentation for maintenance and repair*. IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 10, pages 1355–68, 2011. 48
- [Hoolet06] Hoolet. *Hoolet Reasoner*, 2006. 35
- [HP09] HP. *Joseki - A SPARQL Server for Jena*, 2009. 110
- [Hutchison10] David Hutchison & John C Mitchell. *Medical imaging and augmented reality*, volume 6326. Springer Berlin Heidelberg, 2010. 29
- [IETF11] IETF. *The WebSocket Protocol*, 2011. xxi, 102
- [IRIS06] IRIS. *IRIS Reasoner*, 2006. 35
- [ISO02] ISO. International Standardization Organization (ISO) and International Electrotechnical Commission (IEC) ISO/IEC 14496-1 :2002, information technology ? coding of audio-visual objects – MPEG-4 part 1. 2002. 99

- [ISO/IEC-1449608] ISO/IEC-14496. *ISO/IEC 14496-11 :2005/Cor 5 :2008*, 2008. 40
- [ISO/IEC-1477297] ISO/IEC-14772. *The Virtual Reality Modeling Language*, 1997. 36
- [Iung09] Benoît Iung, Eric Levrat, Adolfo Crespo Marquez & Heinz Erbe. *Conceptual framework for e-Maintenance : Illustration by e-Maintenance technologies and platforms*. *Annual Reviews in Control*, vol. 33, no. 2, pages 220–229, 2009. 33
- [Jonassen91] David H Jonassen. *Objectivism versus constructivism ? Do we need a new philosophical paradigm ?* *Educational Technology Research & Development*, vol. 39, no. 3, pages 5–14, Springer, 1991. 30, 54
- [Julier00] Db Julier & L Rosenblum. *BARS : Battlefield Augmented Reality System*. *Virtual Reality*, vol. 7525, no. Code 5580, pages 1–7, Citeseer, 2000. vii, 29
- [Kato99] Hirokazu Kato & Mark Billinghurst. *Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System*. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, IWAR '99*, pages 85–, Washington, DC, USA, 1999. IEEE Computer Society. 21
- [Khronos08] Khronos. *COLLADA - 3D Asset Exchange Schema*, October 2008. 37
- [Khronos11] Khronos. *WebGL Specification*, 2011. 39
- [Kock11] Mirjam Kock & Alexandros Paramythis. *Activity sequence modeling and dynamic clustering for personalized e-learning*. *User Modeling and User-Adapted Interaction*, vol. 21, no. 1-2, pages 51–97, Kluwer Academic Publishers, Hingham, MA, USA, April 2011. 33
- [Landragin06] Jean-Pierre Landragin & Philippe Schwartz. *L'instrumentation PXL*, 2006. 7, 61
- [Lang07] Daniela Lang, Christoph Mengelkamp, Reinhold S Jaeger, Didier Geoffroy, Michel Billaud & Thomas Zimmer. *Pedagogical evaluation of remote laboratories in eMerge project*. *European Journal of Engineering Education*, vol. 32, no. 1, pages 57–72 ST – Pedagogical evaluation of remote labor, 2007. 43
- [LeFeuvre11] Jean LeFeuvre, Cyril Concolato, Jean-Claude Dufourd, Romain Bouqueau & Jean-Claude Moissinac. *Experimenting with multimedia advances using GPAC*. In *Proceedings of the 19th ACM international conference on Multimedia, MM '11*, pages 715–718, New York, NY, USA, 2011. ACM. 95
- [Liarokapis10] Fotis Liarokapis & Sara De Freitas. *A Case Study of Augmented Reality Serious Games*, 2010. 28

- [Lindstaedt09] Stefanie N. Lindstaedt, Günter Beham, Barbara Kump & Tobias Ley. *Getting to Know Your User — Unobtrusive User Model Maintenance within Work-Integrated Learning Environments*. In Proceedings of the 4th European Conference on Technology Enhanced Learning : Learning in the Synergy of Multiple Disciplines, EC-TEL '09, pages 73–87, Berlin, Heidelberg, 2009. Springer-Verlag. 33
- [Lowe04] David G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. Int. J. Comput. Vision, vol. 60, no. 2, pages 91–110, Kluwer Academic Publishers, Hingham, MA, USA, November 2004. vii, 23, 24
- [Lowe09] David Lowe, Chris Berry, Steve Murray & Euan Lindsay. *Adapting a Remote Laboratory Architecture to Support Collaboration and Supervision*. iJOE, vol. 5, no. S1, pages 51–56, 2009. 73
- [Lukosch01] Stephan Lukosch & Jörg Roth. *Reusing Single-User Applications to Create Multi-user Internet Applications*. In Proceedings of the International Workshop on Innovative Internet Computing Systems, IICS '01, pages 79–90, London, UK, UK, 2001. Springer-Verlag. 50
- [Machotka10] Jan Machotka, Zorica Nedic, Andrew Nafalski & Ozdemir Gol. *1st WIETE Annual Conference on Engineering and Technology Education*. In 17th IFAC (International Federation of Automatic Control) World Congress, 2010. 46
- [Marshall11] Tommy Marshall. *Moving the museum outside its walls : An Augmented Reality Mobile Experience*, 2011. vii, 28
- [Milgram94] P. Milgram & F. Kishino. *A Taxonomy of Mixed Reality Visual Displays*. IEICE Trans. Information Systems, vol. E77-D, no. 12, pages 1321–1329, December 1994. xxxi, 20
- [Moravec80] Hans Moravec. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. In tech. report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University & doctoral dissertation, Stanford University, numéro CMU-RI-TR-80-03. September 1980. 23
- [Muja09] Marius Muja & David G. Lowe. *Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration*. In VISAPP (1), pages 331–340, 2009. xvi, 68
- [Muller08] A. Muller, A. Crespomarquez & B. Iung. *On the concept of e-maintenance : Review and current research*. Reliability Engineering & System Safety, vol. 93, no. 8, pages 1165–1187, August 2008. 34

- [Muller09] Dieter Muller & Frederico Schaf. *A Low Cost Learning Environment for Collaborative Engineering*. In Proc.6th International Conference on Remote Engineering and Virtual Instrumentation (REV'09), Bridgeport, USA, 2009. 45
- [Murray08] S. Murray, D. Lowe, E. Lindsay, V. Lasky & D. Liu. *Experiences with a hybrid architecture for remote laboratories*. Frontiers in Education, Annual, vol. 0, pages F1B-15-F1B-19, IEEE Computer Society, Los Alamitos, CA, USA, 2008. 44
- [Naimark02] Leonid Naimark & Eric Foxlin. *Circular Data Matrix Fiducial System and Robust Image Processing for a Wearable Vision-Inertial Self-Tracker*. In Proceedings of the 1st International Symposium on Mixed and Augmented Reality, ISMAR '02, pages 27-, Washington, DC, USA, 2002. IEEE Computer Society. 22
- [Navab99] N. Navab, E. Cubillo, B. Bascle, J. Lockau, K.-D. Kamsties & M. Neuberger. *Cylicon : A Software Platform for the Creation and Update of Virtual Factories*. In Proc. 7th IEEE Int'l Conf. Emerging Technologies and Factory Automation, ETFA 99, pages 459-463. IEEE Press, 1999. 21
- [Ng09] I. Ng, P. Hwang, D. Kumar, C. Lee, R. Kockro & Y. Sitoh. *Surgical planning for microsurgical excision of cerebral arterio-venous malformations using virtual reality technology*. Acta Neurochir (Wien), 2009. 29
- [O'Connor05] Martin J. O'Connor, Holger Knublauch, Samson W. Tu, Benjamin N. Grosz, Mike Dean, William E. Grosso & Mark A. Musen. *Supporting Rule System Interoperability on the Semantic Web with SWRL*. In International Semantic Web Conference, pages 974-986, 2005. 35
- [Odeh09] Salaheddin Odeh. *Remote experimentation using augmented reality*. Ubiquitous Computing and Communication, vol. 4, no. 1, pages 7-15, 2009. xi, 47, 52
- [OpenCV12] OpenCV. *Open Source Computer Vision*, 2012. 106
- [Orduna11] Pablo Orduna, Javier Garcia-Zubia, Jaime Irurzun, Diego Lopez de Ipina & Luis Rodriguez-Gil. *Enabling mobile access to Remote Laboratories*. In International Conference on Engineering Education, 2011. 45
- [Ozdenizci11] Busra Ozdenizci, Kerem Ok, Vedat Coskun & Mehmet N. Aydin. *Development of an Indoor Navigation System Using NFC Technology*. In Proceedings of the 2011 Fourth International Conference on Information and Computing, ICIC '11, pages 11-14, Washington, DC, USA, 2011. IEEE Computer Society. 22

- [Pellet12] Pellet. *Pellet Reasoner.*, 2012. 35
- [Pereira09] Carlos Eduardo Pereira & Frederico Menine Schaf. *Integrating Mixed-Reality Remote Experiments Into Virtual Learning Environments Using Interchangeable Components*. IEEE Transactions on Industrial Electronics, vol. 56, pages 4776–4783, 2009. 45, 73
- [Press12] Oxford University Press. *Oxford Dictionaries*, 2012. 32
- [Ross11] R. G Ross & S. W. Lam. *Building business solutions : Business analysis with business rules*. Business Rule Solutions. LLC, 2011. 34
- [Rosten05] Edward Rosten & Tom Drummond. *Fusing points and lines for high performance tracking*. In IEEE International Conference on Computer Vision, volume 2, pages 1508–1511, October 2005. vii, 23
- [Ruble11] Ethan Rublee, Vincent Rabaud, Kurt Konolige & Gary Bradski. *ORB : An efficient alternative to SIFT or SURF*. Computer Vision, IEEE International Conference on, vol. 0, pages 2564–2571, IEEE Computer Society, Los Alamitos, CA, USA, 2011. vii, 25
- [Salzmann08] Christophe Salzmann, Denis Gillet, Peter Scott & Kevin Quick. *Remote lab : online support and awareness analysis*. In 17th IFAC (International Federation of Automatic Control) World Congress, 2008. 46
- [Schaf09] Frederico M. Schaf, Dieter Müller, F. Wilhelm Bruns, Carlos E. Pereira & Heinz-Hermann Erbe. *Collaborative learning and engineering workspaces*. Annual Reviews in Control, vol. 33, no. 2, pages 246–252, 2009. 45
- [Scherp05] Ansgar Scherp & Susanne Boll. *Paving the Last Mile for Multi-Channel Multimedia Presentation Generation*. In in Proceedings of the 11th Multimedia Modeling Conference, pages 190–197. ACM Press, 2005. 36
- [Schilit94] B. Schilit, N. Adams & R. Want. *Context-aware computing applications*. Workshop on Mobile Computing Systems and Applications, pages 85–90, IEEE Comput. Soc. Press, 1994. 32
- [Schmidt08] Kay-Uwe Schmidt, Jörg Dörflinger, Tirdad Rahmani, Mehdi Sahbi, Ljiljana Stojanovic & Susan Marie Thomas. *An user interface adaptation architecture for rich internet applications*. In Proceedings of the 5th European semantic web conference on The semantic web : research and applications, ESWC'08, pages 736–750, Berlin, Heidelberg, 2008. Springer-Verlag. 33
- [Signes00] J. Signes, Y. Fisher & A. Eleftheriadis. *MPEG-4's binary format for scene description*. vol. 15, no. 4-5, pages 321–345, January 2000. 82

- [Sutherland68] Ivan E. Sutherland. *A head-mounted three dimensional display*. In Proceedings of the December 9-11, 1968, fall joint computer conference, part I, AFIPS '68 (Fall, part I), pages 757–764, New York, NY, USA, 1968. ACM. 19, 22
- [Suzuki10] Naoki Suzuki, Asaki Hattori, Kazuo Tanoue, Satoshi Ieiri, Kozo Konishi, Morimasa Tomikawa, Hajime Kenmotsu & Makoto Hashizume. *Scorpion shaped endoscopic surgical robot for NOTES and SPS with augmented reality functions*. In Proceedings of the 5th international conference on Medical imaging and augmented reality, MIAR'10, pages 541–550, Berlin, Heidelberg, 2010. Springer-Verlag. vii, 29
- [Swan05] J. Edward Swan & Joseph L. Gabbard. *Survey of User-Based Experimentation in Augmented Reality*. In Proceedings of 1st International Conference on Virtual Reality, HCI International 2005, July 2005. 19, 65
- [Takacs11] Barnabas Takacs. *Immersive interactive reality : Internet-based on-demand VR for cultural presentation*. *Virtual Real.*, vol. 15, no. 4, pages 267–278, Springer-Verlag, London, UK, UK, November 2011. 28
- [Tawfik12] Mohamed Tawfik, Elio San Cristóbal, Sergio Martín, Charo Gil, Alberto Pesquera, Pablo Losada, Gabriel Díaz, Juan Peire, Manuel Castro, Javier García Zubía, Unai Hernández, Pablo Orduña, Ignacio Angulo, Maria C. Costa-Lobo, Maria A. Marques, Maria C. Viegas & Gustavo R. Alves. *VISIR : Experiences and Challenges*. *iJOE*, vol. 8, no. 1, pages 25–32, 2012. 44
- [Tran04] SM Tran, Marius Preda, FJ Preteux & K Fazekas. *New proposal for enhancing the interactive capability in MPEG-4*. 2004 IEEE 6th Workshop on Multimedia Signal Processing, pages 415–418, 2004. 81
- [Trevelyan04] James Trevelyan. *Lessons Learned from 10 Years Experience with Remote Laboratories*. *iCEER 2004 International Conference on Engineering Education and Research Progress Through Partnership*, no. 2001, pages 687–697, 2004. 44
- [Tuddenham10] Philip Tuddenham & Peter Robinson. *Coordination and Awareness in Remote Tabletop Collaboration*. In *Tabletops*, pages 407–434. 2010. 73
- [Tuytelaars08] Tinne Tuytelaars & Krystian Mikolajczyk. *Local invariant feature detectors : A survey*. Now Publishers Inc., Hanover, MA, USA, 2008. vii, 23, 24
- [Ukkonen92] Esko Ukkonen. *Constructing Suffix Trees On-Line in Linear Time*. In Proceedings of the IFIP 12th World Computer Congress on

- Algorithms, Software, Architecture - Information Processing '92, Volume 1 - Volume I, pages 484–492, Amsterdam, The Netherlands, The Netherlands, 1992. North-Holland Publishing Co. 111
- [Vargas-Vera08] Maria Vargas-Vera & Miltiadis Lytras. *Personalized Learning Using Ontologies and Semantic Web Technologies*. Emerging Technologies and Information Systems for the Knowledge Society, vol. 5288, pages 177–186, Springer Berlin Heidelberg, 2008. 34
- [Vezhnevets06] V. Vezhnevets. *Opencv calibration object detection.*, 2006. 69
- [W3C04a] W3C. *Resource Description Framework.*, 2004. 35
- [W3C04b] W3C. *Resource Description Framework Schema.*, 2004. 35
- [W3C08] W3C. *Synchronized Multimedia*, December 2008. 37
- [W3C11a] W3C. *HTML 5, A vocabulary and associated APIs for HTML and XHTML*, 2011. 39
- [W3C11b] W3C. *Synchronized Multimedia*, August 2011. 37
- [W3C12] W3C. *The WebSocket API.*, 2012. xxi, 103
- [Yamabe11] Tetsuo Yamabe, Takahiro Iwata, Takahiro Shichinohe & Tatsuo Nakajima. *Prototyping Augmented Traditional Games : Concept, Design and Case Studies*. In GPC, pages 105–116, 2011. 28
- [Zha02] Visual marker detection and decoding in AR systems : a comparative study, 2002. 21
- [Zhang00] Zhengyou Zhang. *A Flexible New Technique for Camera Calibration*. In IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 22, pages 1330–1334, 2000. xvi, 68
- [Zhang10] Zhengyou Zhang. *Estimating Projective Transformation Matrix (Collineation, Homography)*. Rapport technique MSR-TR-2010-63, Microsoft Research, May 2010. xvi, 68
- [Zhang11] J. Zhang, S.K. Ong & A.Y.C. Nee. *RFID-assisted assembly guidance system in an augmented reality environment*. International Journal of Production Research, vol. 49, no. 13, pages 3919–3938, 2011. 22
- [Zhou08] Feng Zhou, Henry Been-Lirn Duh & Mark Billinghurst. *Trends in augmented reality tracking, interaction and display : A review of ten years of ISMAR*. In Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR '08, pages 193–202, Washington, DC, USA, 2008. IEEE Computer Society. 22

- [Zubía12] Javier García Zubía, Ignacio Angulo, Pablo Orduña, Unai Hernández, Diego López de Ipiña, Luis Rodríguez, Olga Dziabenko & Veronica Canivell. *WebLab-Deusto-CPLD : A Practical Experience*. *ijOE*, vol. 8, no. S1, pages 17–18, 2012. 45
- [Zutin11] Danilo Garbi Zutin. *A VISIR Lab Server for the iLab Shared Architecture*. *ijOE*, vol. 7, no. S1, pages 14–17, 2011. 44