Institut National Polytechnique de Grenoble
ENSIMAG

Université Joseph Fourier
U.F.R.
Informatique & Mathématiques Appliquées

I.M.A.G.

DEA D'INFORMATIQUE

Projet présenté par :
François Bérard

Vision par Ordinateur pour la Réalité Augmentée : Application au Bureau Numérique

Effectué au laboratoire :  IMAG
Laboratoire de Génie Informatique

Date : le 22 juin 1994

Présidents du Jury : **Augustin Lux** (LIFIA-IMAG)
Michel Riveill (Bull-IMAG)

Membres du Jury : **Didier Aubert** (ITMI)
Joëlle Coutaz (LGI-IMAG)
James L. Crowley (LIFIA-IMAG)

Je tiens à remercier Joëlle Coutaz pour le temps, la confiance et les moyens qu'elle a mis à ma disposition ; Jim Crowley pour ses excellents conseils ; ainsi que Laurence Nigay pour son aide précieuse ;

Je remercie également Messieurs Augustin Lux et Michel Riveill de présider mon jury ainsi que Monsieur Didier Aubert d'avoir accepter d'en être membre ;

Enfin, je remercie Sébastien, Jean-Pascal, Daniel, Stéphane, Éric, Philippe, Francis, Gilles, Nathalie, Martine, et Laurent pour leur aide, leur bonne humeur, leurs coups de gueule, bref, tout ce qui fait la vie de l'équipe IHM.

SOMMAIRE

TABLE DES FIGURES

Chapitre 5
Suivi par corrélation **47**

Chapitre 6
Suivi par contour actif (Snake) **57**

INTRODUCTION

Le sujet

Cette étude a pour thème l'apport de la vision par ordinateur dans les Interfaces Homme-Machine appliquées à la "réalité augmentée". La réalité augmentée vise à créer une synergie performante entre l'électronique et les pratiques non informatisées du quotidien. Son objectif est d'estomper la barrière entre les objets informatiques (par exemple, un document présenté à l'écran) et les objets et activités du monde réel (tel le livre ouvert sur la table de travail). Nous étudions l'utilisation de la vision par ordinateur pour un exemple innovant de réalité augmentée : le bureau numérique. Dans ce cadre, nous observons le suivi d'objets réels au moyen de la vision par ordinateur.

Le contexte

Ce travail s'inscrit dans le contexte général des "interfaces nouvelles". L'innovation dans les interfaces tient à la disponibilité sur les stations de travail usuelles, de mécanismes d'interaction sophistiqués de reconnaissance et de synthèse : vision par ordinateur, parole, génération automatique de textes et d'images réalistes, vidéo, etc. L'innovation tient également à la possibilité pour l'utilisateur et le système de combiner ces différentes formes de techniques en fonction de la tâche, des préférences ou des intentions communicationnelles.

La multiplicité des possibilités offertes par les interfaces nouvelles peut se voir comme un facteur de souplesse avec, sa compagne immédiate, la complexité. Il convient alors de s'interroger sur l'utilité et l'utilisabilité de ces nouvelles interfaces, deux aspects qui relèvent de la psychologie et de l'ergonomie cognitives. En psychologie, on visera à développer ou à étendre des théories existantes pour étudier, c'est-à-dire prédire ou expliquer, les performances sensori-motrices et cognitives de l'utilisateur mis en situation d'interaction innovante. Les travaux actuels dans ce domaine restent cependant prospectifs ([Barnard93]) et font souvent appel à l'approche expérimentale du Magicien

d'Oz ([Mignot93], [Coutaz93]). Bien que la psychologie ou les règles d'usage ergonomiques ne puissent, à l'heure présente, fournir des réponses opérationnelles au cas des nouvelles interfaces, il convient néanmoins d'être vigilant sur l'utilisabilité de telles interfaces. Nos objectifs se situent dans ce contexte général de la Communication Homme-Machine : concevoir et mettre en œuvre des interfaces nouvelles utiles et utilisables.

Les interfaces nouvelles désignent à la fois les interfaces multimodales ([Nigay94]), la réalité virtuelle ([Krueger91]), et la réalité augmentée. Nous retenons ce dernier thème.

Les objectifs

Dans le contexte que nous venons de fixer, nous visons quatre objectifs :

1) comprendre et cerner les propriétés de la réalité augmentée,

2) identifier l'apport de la vision par ordinateur dans la réalité augmentée,

3) définir les contraintes techniques que la réalité augmentée exige de la vision par ordinateur,

4) mener une étude comparative de techniques de vision répondant à ces contraintes.

Ces quatre objectifs adoptent le bureau numérique comme canevas applicatif intégrateur.

L'organisation du rapport

La structure de ce rapport reflète la logique de nos objectifs. Elle comprend deux parties : la première est centrée sur les préoccupations "IHM" ; la seconde sur les solutions techniques de la vision par ordinateur.

Dans la partie "IHM", nous définissons au chapitre 1 le concept de réalité augmentée. Nous prenons soin de le situer par rapport à la réalité virtuelle qui, on le sait, fait l'objet d'attentions particulières. Ensuite, nous présentons au chapitre 2 un exemple de réalité augmentée : le bureau numérique qui sert d'application pilote à ce projet. Nous montrons alors l'intérêt de l'usage de la caméra comme dispositif d'acquisition et son rôle dans la symbiose entre mondes électronique et physique. Nous relevons les premières contraintes auxquelles il convient de satisfaire. Cette étude technique fait l'objet de la seconde partie du rapport.

Dans la partie "vision par ordinateur", nous étudions au chapitre 3 l'un des problèmes relevés dans notre analyse "IHM" du domaine : le suivi d'objets. Nous identifions les critères de qualité qu'un système de vision par ordinateur doit satisfaire pour remplir cette fonction. Au chapitre 4, nous présentons notre banc d'essai, FingerPaint, une application qui illustre le suivi d'un objet particulier : le doigt. Cette application permet à l'utilisateur de dessiner sur une feuille de papier réelle au moyen d'un doigt ou d'un objet physique approchant qu'il aurait plongé dans de la peinture électronique. Aux chapitres 5 et 6 nous présentons deux techniques de vision applicables au suivi d'objet : les

contours actifs (ou snakes) et la corrélation d'images. Ces deux techniques sont évaluées comparativement au moyen de FingerPaint en s'appuyant sur les critères de qualité du chapitre 3.

En conclusion, nous soulignons les points contributifs de nos travaux et présentons les perspectives possibles.

Chapitre 1

RÉALITÉ AUGMENTÉE : JUSTIFICATION ET DÉFINITION

Le concept de Réalité Augmentée est le dernier né de l'évolution galopante des Interfaces Homme-Machine (voir le numéro spécial de CACM, juillet 1993, vol. 36, no7). Ce chapitre a pour but d'en expliquer les concepts fondateurs et les objectifs. Nous procéderons comme suit :

• au paragraphe 1, une réflexion sur la manipulation directe, les interfaces multimodales et la réalité virtuelle nous permettra de justifier cette nouvelle tendance. Ces modèles d'interaction, de même que la réalité augmentée, obéissent tous au principe directeur énoncé dans le chapitre introductif de ce rapport : améliorer l'utilité et l'utilisabilité des systèmes ;

• pour clarifier la terminologie, nous présentons au paragraphe 2 notre cadre taxonomique qui permet de distinguer le monde réel des réalités virtuelles et augmentées. Ce cadre offre aussi la possibilité de distinguer plusieurs classes de réalités augmentées. Nous illustrons la discussion au moyen de systèmes existants.

1.1. De la manipulation directe à la réalité augmentée

Nous distinguerons trois étapes successives dans les démarches de recherche en Communication Homme-Machine : la manipulation directe, les interfaces multimodales et la réalité virtuelle.

1.1.1. La manipulation directe

Avec l'objectif d'améliorer l'utilisabilité des systèmes, la recherche en IHM a vu dans la manipulation directe une avancée prometteuse. Au lieu de décrire une action dans un langage, l'utilisateur agit directement sur l'objet sans intermédiaire linguistique.

Cette distinction entre action et langage se retrouve dans l'espace de conception des interfaces de D. Frohlich [Frohlich91]. Frohlich distingue deux "modes" d'interaction : le mode langage et le mode action. La notion de mode recouvre chez Frohlich les deux métaphores essentielles de l'interaction homme-machine présentées par Hutchins et ses co-auteurs dans [Hutchins86]. Il distingue le *mode action* qui correspond à la métaphore du monde modèle ("model world metaphor") et le *mode langage* pour désigner la métaphore conversationnelle ("conversation metaphor").

Une interface qui répond au *mode action* présente à l'utilisateur un monde sur lequel il lui est possible d'agir directement. Il n'y a pas d'intermédiaire entre l'utilisateur et le monde présenté. L'interface *est* le monde et ce monde change d'état sous l'effet des manipulations de l'utilisateur. Lorsque le principe de cette métaphore est appliqué avec soin, l'utilisateur est conduit à assimiler le monde perçu aux objets du domaine de la tâche. La métaphore du bureau d'Apple [Apple88] est l'une des premières applications réussies de ce principe. B. Laurel qualifie d'interfaces "à la première personne" ces systèmes qui offrent le sentiment d'engagement direct dans l'action sans relais intermédiaire [Laurel86].

Dans le mode langage ou métaphore conversationnelle, l'interface est un support linguistique qui permet au système et à l'utilisateur de converser à propos du monde supposé des objets de la tâche. Les objets du domaine ne sont plus représentés explicitement et ne sont manipulables que par un intermédiaire : l'interface. L'utilisateur décrit les actions à appliquer sur les objets. Il ne les réalise pas directement lui-même. Il les fait réaliser. Pour marquer la présence d'un intermédiaire entre l'utilisateur et les objets de la tâche, B. Laurel parle d'interfaces "à la deuxième personne" [Laurel86]. Si l'on reprend la terminologie des actes de langage d'Austin, le mode langage conduit l'utilisateur à "faire-savoir" au système ou à "faire-faire" via l'interface [Austin62]. Inversement, dans le mode action, l'utilisateur "fait".

La plupart des systèmes informatiques offrent, en vérité, un curieux mélange de ces interfaces à la première et à la deuxième personne. Par exemple, les éditeurs de texte adoptent tous l'action pour ce qui est de la saisie de texte mais ils imposent le conversationnel pour l'activation des commandes. Certes, les menus, les formulaires et les boutons, font l'objet de manipulations directes mais ils servent d'intermédiaires entre l'utilisateur et leurs signifiés, c.-à-d. les concepts du domaine, qui sont alors manipulés par indirection et selon des conventions langagières. Il convient donc d'être prudent sur la qualification "interface à manipulation directe". Il faut savoir que la présence d'objets de présentation comme les menus et les formulaires, ne suffit pas à garantir l'action directe sur les concepts du domaine.

La figure 1.1 illustre la distinction entre mode langage et mode action avec l'exemple de la copie d'un fichier : la première méthode (en haut de la figure) impose à l'utilisateur de connaître le langage compris par l'ordinateur (ici DOS) ; la seconde méthode (illustrée en bas de la figure)

représente de manière graphique le concept de fichier au moyen d'une icône et la copie s'exprime par l'application d'un geste de transfert sur l'icône représentative.

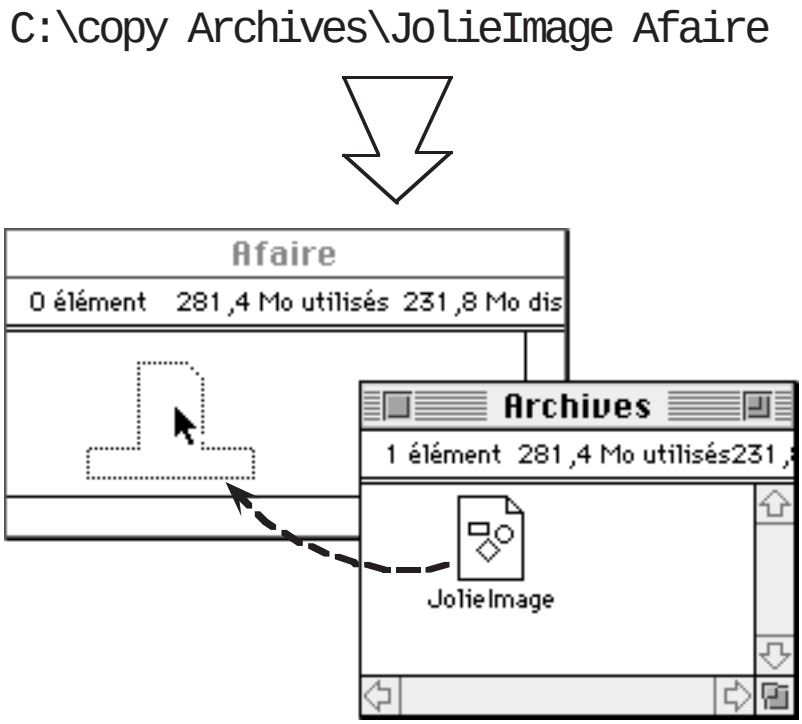


Figure 1.1 : Métaphores conversationnelles et métaphore du monde réel : cas de la copie du fichier "JolieImage" dans le répertoire "Afaire".

Si la manipulation directe a amélioré l'utilisabilité des systèmes, il faut constater que les interfaces graphiques sont confinées aux ressources de la station de travail : l'écran, le clavier et la souris ([Wellner93a]). Ces dispositifs de communication sont trop pauvres pour exploiter les canaux de communication de l'Homme et pour imiter ou observer de manière réaliste le monde qui l'entoure. Ce double constat a donné naissance à deux tendances de recherche : les interfaces multimodales et la réalité virtuelle.

1.1.2. Les interfaces multimodales

L'objectif des interfaces multimodales est d'étendre les facultés sensori-motrices des ordinateurs en vue de répondre aux moyens de communication naturels du sujet humain. Cette extension se manifeste sous différentes formes, depuis la création de nouveaux dispositifs physiques d'interaction jusqu'à la définition de systèmes de représentation symbolique des informations échangées via ces dispositifs. L'éventail des possibilités et la nouveauté de l'entreprise expliquent la variété des termes, tels "multimédia" et "multimodal".

Dans [Nigay94], nous trouvons une étude approfondie sur les problèmes actuels de terminologie dans ce domaine. Nous reprenons à notre compte l'analyse de Nigay : "multimédia" et

“multimodal” se déclinent différemment selon les perspectives et les disciplines de recherche. Typiquement, les définitions s’opposent entre le point de vue de l’utilisateur qui va de pair avec la psychologie, et le point de vue système qui va de pair avec les préoccupations de mise en œuvre informatique. Succintement, si l’on adopte le point de vue de l’utilisateur, le média désigne un véhicule d’information et une modalité correspond à une faculté sensorielle : toucher, audition, olfaction, etc. Si l’on adopte la perspective système,

- un système multimédia permet de manipuler des types de données tels le son, l’image et la vidéo que véhiculent différents médias. Un exemple est le système de courrier électronique de NeXT qui permet d’envoyer des messages audios, textuels et graphiques ;

- un système multimodal va plus loin que le multimédia : dans une interface multimodale, les informations véhiculées font sens. Une interface de ce type permet l’usage combiné de la parole, du geste, de la langue naturelle écrite, et/ou les capacités d’un système de vision qui comprendrait la scène observée. Ainsi, le système de courrier électronique de NeXT serait multimodal si le texte saisi en langue naturelle et les messages oraux étaient compris de la machine. Il est en vérité multimédia puisque le contenu des messages électroniques n’est pas interprété mais encapsulé à bas niveau d’abstraction comme un tout hermétique à remettre à un destinataire humain qui, lui, se chargera de l’interprétation.

MATIS (Multimodal Airline Travel Information System) offre au contraire une interface multimodale pour l’expression de requêtes sur des horaires d’avion. L’utilisateur a plusieurs possibilités. Il peut :

- faire usage de la souris et spécifier sa requête par manipulation directe,
- utiliser le clavier et spécifier sa requête en langue naturelle ou en remplissant un formulaire,
- énoncer sa requête oralement. Par exemple : <Show me the Usair flights from Pittsburgh to Boston>,
- utiliser de manière synergique les trois modes de communication précédemment cités en les combinant. Par exemple :
  <Parole : *Usair flights from this city* >
  + <Sélection avec la souris d'une ville dont le nom est affiché à l’écran>
  + <Parole : *arriving at*>
  + <Saisie d'un horaire en utilisant le clavier>

Les récents progrès techniques des systèmes de reconnaissance de la parole, de la langue naturelle, de la vision par ordinateur et des dispositifs à retour d’effort ([Cadoz94]) rendent possibles l’intégration de ces modalités. L’intégration peut se voir comme une simple cohabitation. Dans ce cas, seul l’usage concurrent ou exclusif des modalités serait permise. Dans MATIS, cela signifierait qu’à un instant donné, une requête pourrait être construite en utilisant soit la parole, soit la manipulation

directe, soit le remplissage de formulaire au clavier. Le but poursuivi par les chercheurs est l'intégration synergique qui autorise les coréférences entre modalités (comme le dernier exemple de MATIS : flights from *this* city + désignation à la souris de la ville en question). L'usage synergique est une pratique dans la communication humaine. Il convient maintenant d'en étudier son transfert en IHM tant sur le plan ergonomique ([Balbo93], [Salber93]) que sur le plan technique (modèles d'architecture idoines comme PAC-Amodeus et des moteurs de fusion comme celui développé par L. Nigay [Nigay94]).

1.1.3. La Réalité Virtuelle

En réaction au constat réducteur des interfaces graphiques à imiter le monde réel, certains auteurs, tel [Krueger91], ont imaginé immerger l'utilisateur dans un monde complètement simulé au moyen de dispositifs d'interaction évolués comme les gants numériques et les casques de visualisation. La figure 1.2 extraite de [Krueger91] présente de tels dispositifs.

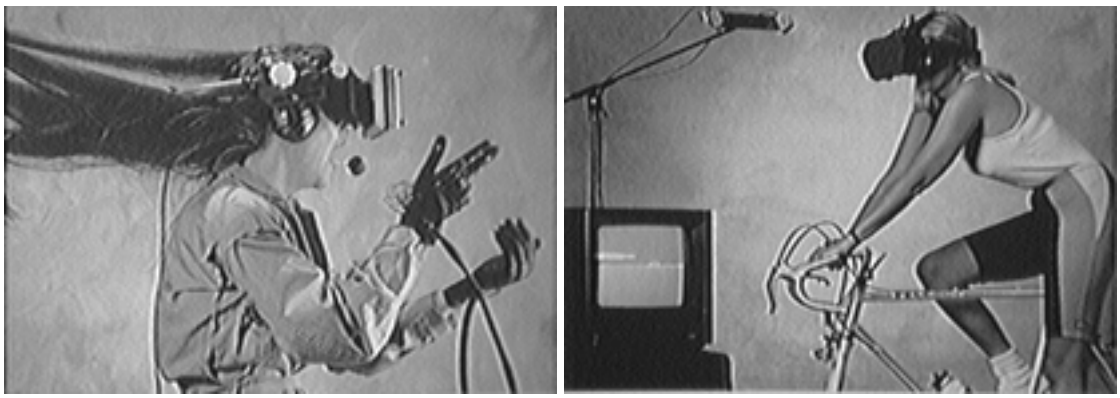


Figure 1.2 : Dispositifs d'interaction pour la réalité virtuelle ([Krueger91]).

Ainsi est né le concept de Réalité Virtuelle : les objets du monde réel sont reproduits aussi fidèlement que possible et éventuellement amplifiés de nouvelles fonctions grâce au support de l'informatique. Depuis les premières expériences de Krueger, le concept de Réalité Virtuelle a connu un succès grandissant : nous trouvons de nos jours de nombreuses applications dans des domaines aussi divers que la médecine, le divertissement (systèmes utilisés dans les salles de jeux publiques), les arts ou la bureautique.

La figure 1.3 montre un exemple d'application au domaine médical. Ce système, qui s'utilise avant une opération neurochirurgicale, permet de visualiser sur l'écran différentes coupes du cerveau du patient [Hinckley94]. Dans une interface graphique usuelle, le chirurgien spécifierait les paramètres de la coupe au moyen d'un formulaire. Ici, la définition de la coupe se fait en manipulant une boule (dispositif physique à six degrés de liberté qui figure le crâne du patient) et une plaque transparente qui concrétise le plan. La photo de la figure 1.3 illustre le dispositif de saisie. Dans ces conditions, le chirurgien spécifie la coupe recherchée en positionnant la plaque de verre par rapport à

la boule. Ces dispositifs permettent au praticien de concentrer l'attention sur la tâche à réaliser (la coupe du cerveau à obtenir) et non sur la façon de spécifier le plan de la coupe.

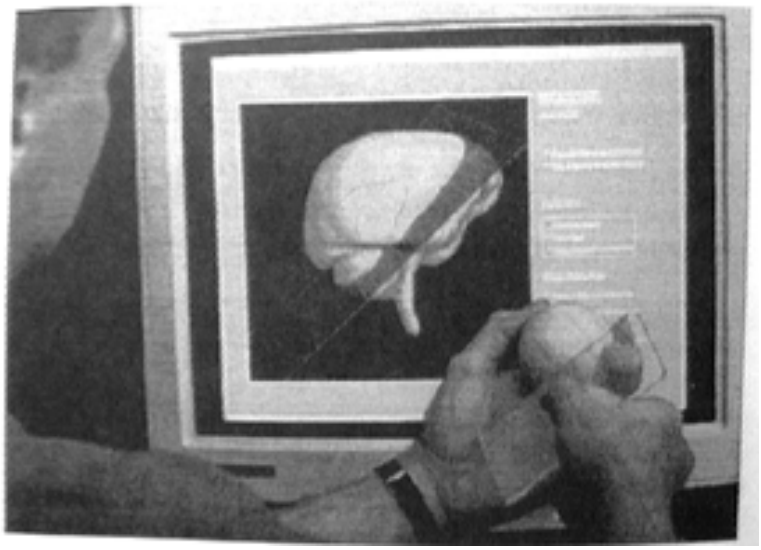


Figure 1.3 : Définition d'un plan de coupe virtuel à l'aide d'objets réels ([Hinckley94]).

Un autre exemple, illustre comment la manipulation directe a progressivement évolué vers la Réalité Virtuelle. Ce système définit un environnement virtuel distribué et interactif (DIVE : Distributed Interactive Virtual Environment) [Fahlén93]. Utilisant la métaphore d'une pièce, l'utilisateur évolue dans un espace où sont disposés plusieurs outils de bureautique : un tableau, une imprimante, etc. Dans cet espace virtuel, un utilisateur est représenté par un cube. Un document est matérialisé par un tableau noir sur lequel l'utilisateur peut écrire ou dessiner. Pour utiliser le tableau noir, l'utilisateur doit se trouver à côté du tableau. Pour commencer à dessiner par exemple, il doit prendre un stylo en sélectionnant un des boutons affichés à gauche du tableau. Ce système permet aussi de travailler à plusieurs sur un même document par exemple. Ainsi plusieurs utilisateurs sont regroupés virtuellement autour d'un même tableau noir.

Suivant la même approche de transcription dans le virtuel, des travaux se sont attachés à transférer dans un espace en trois dimensions les menus et barres de menus : nous trouvons alors des solutions comme les menus en trois dimensions et les cubes et même les SuperCubes [Waterworth94]. Par exemple, chaque facette d'un cube correspond à une commande. Pour accéder à la commande de la facette cachée du cube, l'utilisateur doit pivoter le cube.

Comme dans le VIDEOPLACE de Krueger [Krueger91], ALIVE, développé au Media Lab du MIT, gère une interaction avec tout le corps de l'utilisateur [Maes94]. Mais à l'inverse du VIDEOPLACE qui est 2D (extraction de contours), ALIVE utilise une interaction 3D : une caméra CCD capte des images couleur de l'utilisateur. L'image vidéo est ensuite intégrée dans une scène graphique 3D dans laquelle évoluent des créatures graphiques. L'ensemble est projeté sur un mur

vidéo de 3 mètres sur 4 environ : c'est le miroir magique. La tâche de l'utilisateur est simple : il s'agit de nourrir les créatures (avec une nourriture virtuelle placée dans une assiette virtuelle). Celles-ci viennent à lui dès qu'il pénètre dans la scène. Ce système est un bel exemple d'intégration de recherches en vision par ordinateur, en graphique et en architecture multi-agent (au sens de l'I.A. distribuée).

Au vu de la diversité des applications que nous venons de présenter, en raison aussi de la nouveauté du concept, il est difficile de définir avec précision le terme Réalité Virtuelle : la terminologie n'est pas fixée et l'expression Réalité Virtuelle est fréquemment employée de manière galvaudée. Les définitions trouvées dans la littérature sont trop souvent dirigées par la technique et sont symptomatiques du malaise actuel : ainsi C. Shaw définit les interfaces de Réalité Virtuelle [Shaw92] comme donnant à son utilisateur la possibilité de manipuler des objets en trois dimensions en utilisant des dispositifs physiques d'entrée à trois degrés de liberté au moins. De même J. Waterworth [Waterworth94] souligne que les travaux relevant de la Réalité Virtuelle ont essentiellement porté sur le rendu réaliste d'espace et sur la manipulation d'objets représentés en trois dimensions.

Au contraire dans [Burdea93], nous trouvons une définition qui se détache des aspects techniques et qui par là-même est plus ambitieuse. Nous nous appuierons sur cette définition générale : *"Un système de Réalité Virtuelle est une interface qui implique de la simulation en temps réel et des interactions via de multiples canaux sensoriels. Ces canaux sont ceux de l'homme : vision, audition, toucher, odorat, goût."*. On trouve dans cette définition toutes les notions qui constituent l'attrait de la Réalité Virtuelle : la simulation en temps réel et l'interaction multimodale (au sens de la psychologie). L'objectif final est l'immersion de l'utilisateur dans un espace où il puisse interagir comme il le fait naturellement dans le monde réel.

En résumé, un système de Réalité Virtuelle est une interface multisensorielle temps réel à manipulation directe qui permet à un ou plusieurs utilisateurs d'évoluer dans un espace purement électronique. Cet objectif relève encore de l'utopie. Bien que la définition de Burdea suscite notre intérêt, elle met aussi en évidence la difficulté de réaliser des systèmes comportant de telles capacités. Malgré les progrès effectués au niveau de la simulation du toucher [Cadoz94] et même si l'on accepte la contrainte de porter un gant à retour de forces, on imagine mal comment l'ordinateur peut reproduire la sensation d'écrire avec un stylo sur du papier. Les objets du monde réel ont des propriétés qu'il est difficile, voire non souhaitable, de transposer dans le virtuel.

Si la création d'un monde entièrement virtuel trouve des applications dans certains domaines, il en existe d'autres comme la bureautique où il nous semble plus judicieux et efficace de revenir aux objets matériels. Ce retour au monde réel correspond à l'évolution de la réalité virtuelle à la réalité augmentée. Celle-ci est présentée au paragraphe suivant.

1.2. La réalité augmentée

Nous commencerons par justifier le concept de réalité augmentée avant d'en fournir une définition. Les illustrations sont présentées en 1.3 selon les dimensions de notre cadre taxonomique.

1.2.1. Motivation

Le but de la réalité augmentée est de réunir le meilleur des deux mondes : réel et virtuel. Pour harmoniser la cohabitation de ces deux mondes, la réalité augmentée doit briser cette frontière brutale que l'informatique a installée entre l'électronique et le physique. L'une et l'autre a ses atouts.

Les objets électroniques ont l'avantage d'être manipulables à façon. Par exemple, les documents électroniques ont la capacité d'être reproductibles avec une qualité parfaite ; grâce aux réseaux, ils peuvent être transmis de manière quasi-instantanée d'un bout à l'autre de la planète ; ils possèdent des outils de recherche de mots, de correction orthographique et de traduction dans une autre langue [Newman92].

Cependant, par comparaison à nos actions sur des objets réels comme le papier, les crayons et la gomme, la manipulation (même directe) des documents électroniques est limitée par les capacités des dispositifs électroniques d'entrée-sortie [Wellner93b]. La manipulation des objets du monde réel profite de notre habileté naturelle et de notre habitude. Sans même en être conscients, nous sommes capables de manipuler plusieurs objets en parallèle en tirant partie de nos bras et de nos doigts. De plus, les documents papier ont aussi des propriétés qui ne sont pas reproductibles virtuellement : leur lecture est moins fatiguante que devant un écran ; ils sont acceptés par tous sans contrainte de compatibilité ou de format. Selon Hinckley [Hinckley94], la manipulation d'objets physiques familiers est une technique d'interaction efficace et importante qu'il convient de conserver.

Cette brève analyse indique que choisir entre l'un des deux mondes, abandonnant par là les possibilités de l'un au profit de l'autre, n'est pas nécessairement satisfaisant.

1.2.2. Définition

La réalité augmentée vise à concilier les deux mondes réel et virtuel en ayant recours au virtuel uniquement de manière ciblée pour réaliser des fonctions qui n'existent pas dans le monde réel et donc pour augmenter la réalité. Plutôt que de transposer le monde physique dans celui de l'électronique, la réalité augmentée adopte le choix symétrique : amplifier la réalité par des solutions informatiques.

Nous présentons maintenant une définition plus formelle qui nous permettra de situer les réalités virtuelles et augmentées.

1.3. Notre cadre taxonomique pour les réalités virtuelles et augmentées

Ce cadre est le résultat de notre réflexion personnelle sur une distinction de fond entre réalités virtuelles et augmentées. Nous le présentons en 1.3.1. Ce cadre simpliste est imparfait mais suffit aux besoins de notre discussion. Nous nous proposons ensuite de l'appliquer à l'analyse de systèmes relevant de la réalité augmentée en distinguant interface d'entrée et interface de sortie. Des exemples illustratifs sont alors fournis pour étayer la discussion.

1.3.1. Les dimensions de l'espace

Un système peut se modéliser comme un ensemble d'opérations (ou primitives). Une opération se définit par un opérateur et des opérandes. Nous proposons ici un espace de classification des systèmes en fonction de l'opérateur et des opérandes d'une opération. Comme le montre la figure 1.3, l'espace s'organise selon deux axes intitulés "Action" et "Objet". L'axe "Action" correspond à l'opérateur d'une opération tandis que l'axe "Objet" en désigne les opérandes. Chacun des axes admet deux valeurs : réel et virtuel ou, si l'on préfère, physique et électronique.

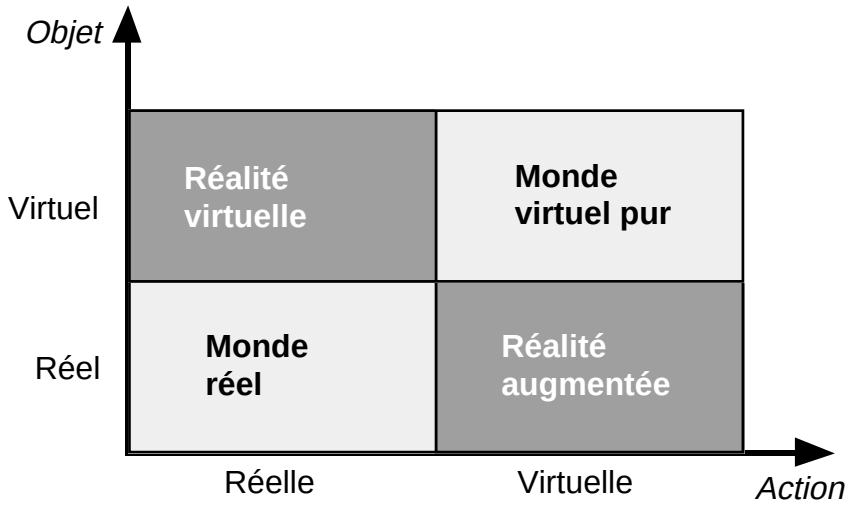


Figure 1.4 : Mondes réel et virtuel, réalités virtuelle et augmentée.

L'intérêt majeur de notre classification est de situer dans un même espace la réalité augmentée et la réalité virtuelle par rapport au monde réel. Dans le monde réel, objets et actions sont physiques. A l'extrême, dans le monde purement électronique, objets et actions sont virtuels. L'un et l'autre sont des représentations électroniques d'entités éventuellement équivalentes dans le monde physique. Par exemple, dans un système de traitement de texte, un fichier Word est la représentation électronique d'un document et les actions possibles (couper, coller, rechercher, remplacer, etc.) sont également électroniques. En édition, le document électronique est fonctionnellement plus riche que sa contrepartie physique mais il est entièrement virtuel. Remarquons que la plupart des systèmes actuels produisent des mondes purement virtuels.

Les deux cas qui nous restent à envisager dans la figure 1.4 correspondent aux situations mixtes : les réalités virtuelle et augmentée. Nous observons que l'une et l'autre impliquent une composante du monde réel (d'où leur nom). Pour la réalité virtuelle, l'action est physique et l'objet de l'action est virtuel : l'utilisateur évolue dans un monde simulé. Par exemple, dans les figures 1.2 et 1.3, l'utilisateur agit (il choisit une coupe, il fait du vélo) et ses actes ont un effet sur un monde simulé (le crâne du patient, un paysage californien).

Dans la réalité augmentée, l'option est inverse à celle de la réalité virtuelle : l'objet est physique alors que l'action est virtuelle. Les propriétés des objets du quotidien sont alors amplifiées de fonctions électroniques. Notons ici que ces fonctions peuvent également exister dans le monde réel. Par exemple, le système offre à l'utilisateur plusieurs options pour activer une même fonction : actions physiques réelles sur les objets ou activation d'une fonction de l'ordinateur. Offrir le choix de l'activation d'une fonction répond au critère de qualité de souplesse des interfaces. Selon le contexte, l'utilisateur choisira l'une ou l'autre des options offertes. Nous retrouvons ce concept de choix dans la méthode de classification UOM (Usage Option et Multiplicité) des systèmes multimodaux [Nigay94].

Nous visons maintenant à appliquer notre canevas général en distinguant le cas de l'interface en entrée (de l'utilisateur vers le système) de celui de l'interface en sortie (du système vers l'utilisateur). Ces deux plans d'analyse d'un système ont été originellement introduits par Norman dans sa théorie de l'action [Norman86] et largement utilisés dans la littérature et notamment par L. Nigay avec son modèle pipe-line pour la classification des interfaces multimodales [Nigay94]. Nous adoptons une démarche d'analyse similaire en l'appliquant aux réalités augmentées.

1.3.2. Interface augmentée en entrée

Nous sommes maintenant en mesure d'avancer une définition plus précise de la réalité augmentée du point de vue des entrées :

Interface augmentée en entrée :

Dans une interface augmentée en entrée, les objets manipulés sont réels et l'utilisateur active des services électroniques.

Chameleon est un système de réalité augmentée en entrée. Il se présente sous la forme d'un écran miniature "conscient" de sa position (6 degrés de liberté) [Fitzmaurice93]. Cette connaissance lui permet de réagir en fonction des objets physiques qui lui sont voisins. Par exemple, en supposant qu'il soit relié à une base de données cartographiques, Chameleon affiche les informations qui concernent le lieu près duquel il est placé. La figure 1.5 montre l'écran miniature positionné sur une carte routière physique. L'écran joue alors le rôle de loupe électronique : le fait de l'approcher d'une ville provoque un zoom fluide corrélé au mouvement de la main.

Figure 1.5 : Ecran réagissant à sa position géographique ([Fitzmaurice93]).

La carte routière de la figure 1.5 est amplifiée d'un service de renseignement. Le concept de lentilles magiques (magic lenses) relève de la même idée ([Bier93], [Bier94]) : une lentille magique est un filtre de forme et de fonction arbitraires qui affecte l'apparence de l'objet sur lequel elle est placée. A la différence de Chameleon qui agit sur des objets physiques, une lentille magique opère sur des objets virtuels graphiques. Mais dans les deux cas, des objets existants, qu'ils soient virtuels ou non, sont étendus de nouvelles fonctions.

1.3.3. Interface augmentée en sortie

Interface augmentée en sortie :

Dans une interface augmentée en sortie, les informations fournies par le système doivent utiliser comme support du rendu le monde virtuel et le monde réel.

Il existe peu d'exemples pour illustrer notre définition. Nous constatons que les rares systèmes actuels exploitent uniquement le **rendu visuel** et surimposent des objets virtuels par-dessus le monde réel.

Par exemple, chez Boeing, un projet vise à fournir aux mécaniciens un écran miniature transparent placé devant un œil (comme une paire de lunettes dont un verre aurait éte remplacé par ce mini-écran LCD). Ce dispositif permet de lire les informations de l'écran tout en voyant le monde réel donc tout en continuant à travailler. Cette idée pourrait être étendue à d'autres domaines comme la visite des musées : l'écran miniature afficherait les informations relatives à l'œuvre d'art observée.

Dans le système KARMA [Feiner93], le monde virtuel vient en surimpression de la vue du monde réel par le biais de deux écrans "see-through". Ces deux écrans sont pilotés par un ordinateur capable de repérer la position et l'axe de vue de l'utilisateur ainsi que la position de certains objets de l'entourage. Des objets en trois dimensions et en fil de fer sont synthétisés sur les deux écrans et fournissent une information virtuelle plaquée sur les objets réels. La première application de cette idée

est un système d'aide à la maintenance d'imprimante laser. Grâce au casque de visualisation, les pièces à manipuler sont mises en évidence par leur contours en fil de fer. Des flèches sont affichées pour indiquer le sens de la manipulation qu'il convient d'effectuer (cf. figure 1.6).

Figure 1.6 : Surimposition d'une image informatique sur un objet réel ([Feiner93]).

A l'évidence, KARMA exploite à la fois le monde réel et virtuel en affichant des objets virtuels sur des objets réels. Le système peut également afficher les pièces internes de l'imprimante donnant ainsi l'impression de voir à travers le plastique. Cet exemple montre la pertinence de la réalité augmentée par rapport au "tout virtuel" : tout n'est pas recréé de manière informatique mais les objets réels sont amplifiés de capacités virtuelles (tel le plastique qui devient translucide).

1.4. Conclusion

Dans ce chapitre, nous avons cerné les signes distinctifs des réalités augmentées et virtuelles. En distinguant l'interface en entrée de l'interface en sortie, nous proposons deux définitions :

Interface augmentée en entrée :

interface telle que les objets manipulés sont physiques et les services activés électroniques.

Interface augmentée en sortie :

interface telle que les informations fournies par le système utilisent comme support du rendu à la fois le réel et le virtuel.

La distinction entre entrée et sortie souligne le fait qu'un système de réalité augmentée en entrée n'est pas nécessairement un système de réalité augmentée en sortie et vice-versa. En allant plus loin, il peut être augmenté en entrée et virtuel en sortie et vice-versa! Nous aboutissons aux mêmes conclusions que L. Nigay dans sa classification des interfaces multimodales : en IHM, entrée et sortie, bien que fortement couplées, ne sont pas nécessairement symétriques. Nous verrons au chapitre suivant que le bureau numérique obtient la symétrie grâce à la projection de l'écran sur la table de travail et l'usage de vision par ordinateur.

La vision par ordinateur devrait jouer un rôle privilégié dans les "réalités" à venir. Nous voyons à cela deux raisons essentielles :

1) Concernant la manipulation d'objets virtuels ou réels, la caméra fait moins intrusion que le gant numérique. Elle permet donc une manipulation plus naturelle. Elle conserve la précision du geste (la main n'est pas habillée) et l'utilisateur n'a pas de "fil à la patte" (voir la notion de "dispositif écologique" développée par S. Maury [Maury94]).

2) La caméra peut observer autre chose que le geste, comme par exemple l'outil (stylo, gomme) que manipule l'utilisateur. Ainsi la vision permet l'utilisation des objets du quotidien (au contraire des dispositifs spécifiques comme la boule du système médical présenté ci-dessus ou l'écran miniature du système Chameleon).

La vision, en tant que technique d'interaction d'entrée, est donc une source d'informations extrêmement riche. Inversement, l'extraction des informations est un processus complexe. L'objectif de ce projet de DEA est de préciser les besoins en vision par ordinateur pour la réalité augmentée et d'identifier des solutions possibles. Cette tâche étant cependant trop ambitieuse pour le cadre d'un DEA, nous nous restreindrons à une application particulière de la réalité augmentée, le bureau numérique, et à une famille de techniques de vision indispensables à sa réalisation : le suivi d'objet. Nous poursuivons donc au chapitre suivant en présentant en détail les concepts véhiculés par le bureau numérique.

Chapitre 2

LE BUREAU NUMÉRIQUE

Le concept de bureau numérique (ou digital desk) nous vient de RXRC (ex EuroPARC) [Wellner93b]. Il se veut une réponse aux ambitions irréalistes de la bureautique qui clamait dès le départ, que l'objectif “zéro-papier” serait atteint en quelques années. La solution du tout électronique était un choix extrême que la pratique a rapidement mis en cause : de source sérieuse, on note que le travail sur papier dans les bureaux a été multiplié par 6 depuis 1970 et qu’il progresse chaque année de 20% environ ([Wellner93b]). C'est pourquoi le concept du bureau numérique, qui prône une coopération harmonieuse entre documents papiers et électroniques, semble prometteur.

Après avoir présenté le concept de bureau numérique en 2.1, nous l’illustrons en 2.2 au moyen des premières applications qui en ont été faites. Nous orientons ensuite notre analyse vers les problèmes techniques de mise en œuvre : en 2.3, nous identifions les dispositifs d’affichage et de capture possibles. Nous retenons et justifions le choix de la vision par ordinateur et évoquons en 2.4, les problèmes qui l’accompagnent dans la réalisation du bureau numérique : le calibrage, la correction d’image et le suivi d’objets.

2.1. Présentation du concept

Le concept de bureau numérique a été introduit au centre de recherche de Xerox à Cambridge avec les projets “MARCEL” et “Digital Desk” ([Newman92], [Wellner93b]). Le principe directeur de ces projets est le retour au bureau réel avec le papier, le jeu de crayons préférés, la gomme et ... la tasse de café. La métaphore du “desktop” devient inutile puisque l'espace de travail *est* le bureau.

Comme le montre la figure 2.1, les services informatiques sont intégrés au bureau réel via une caméra et un projecteur fixés verticalement à la surface de travail. Les actions de l'utilisateur sont fournies à la machine par l'intermédiaire de la caméra alors que la projection de l'écran sur le bureau procure dès le départ l'équivalent d'une interface graphique, avec en plus la possibilité de fusionner

les objets réels (ceux qui sont posés sur la table) avec les objets virtuels (ceux qui sont projetés sur la table).

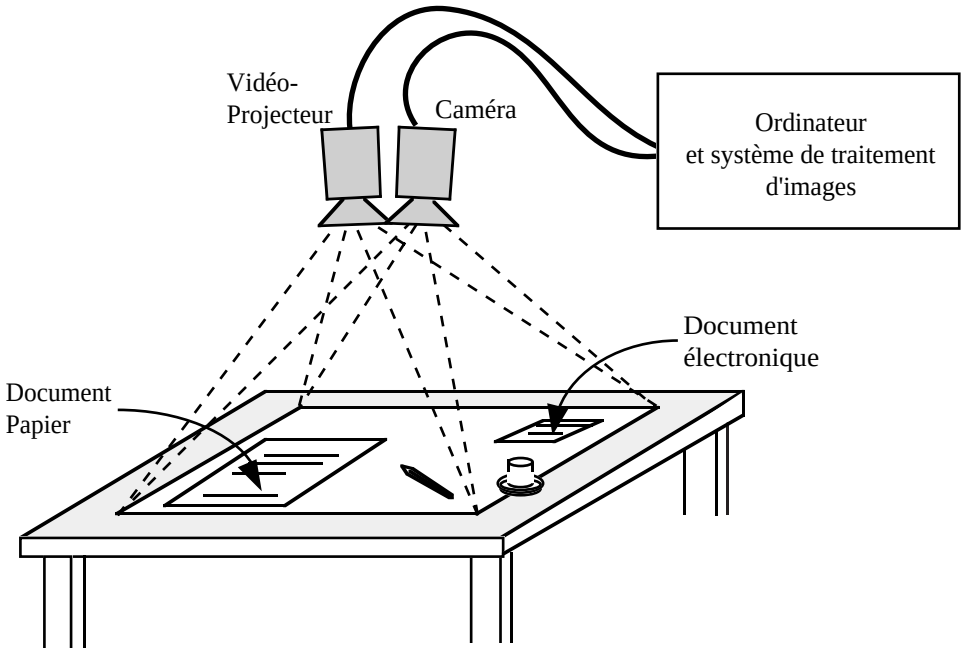


Figure 2.1 : Le DigitalDesk

Cette installation satisfait trois caractéristiques essentielles :

- une image est projetée sur le plan de travail mais aussi sur les documents qui y sont entreposés ;
- l'utilisateur dispose du clavier et de la souris mais conserve ses outils du monde réel : stylos, gomme, doigts ;
- les documents posés sur le bureau peuvent être “lus” par l’utilisateur mais aussi par la machine.

La combinaison de ces caractéristiques montre que le bureau numérique est un exemple de réalité augmentée au sens du chapitre 1 et que la combinaison du réel et du virtuel est présente aussi bien en entrée qu’en sortie. De fait, le bureau numérique ouvre un vaste champ d’applications dont les premières sont présentées au paragraphe suivant.

2.2. Applications

La première application qu’il est raisonnable d’envisager est l’amplification des documents physiques par les services électroniques que nous apprécions aujourd’hui dans les traitements de texte. Ainsi, le système MARCEL fournit un dictionnaire Français/Anglais : l’utilisateur désigne le mot à traduire, puis l’endroit du bureau sur lequel la traduction doit porter [Newman92]. Aux paragraphes qui suivent nous présentons les premiers développements réalisés dans le cadre du projet

DigitalDesk : une calculatrice (inspiré des feuilles de calcul des tableurs), “PaperPaint” (un ensemble d'outils pour le dessin) et le “DoubleDigitalDesk” (ou comment le papier sert de base au collecticiel).

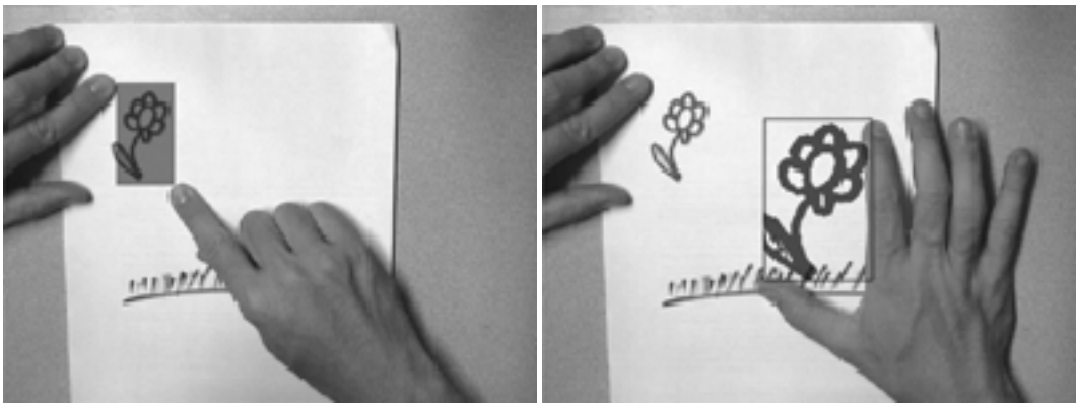
2.2.1. La calculatrice

La calculatrice de poche est un outil dont on se sert fréquemment, conservée à portée de main sur le bureau. Une manipulation classique consiste à saisir des nombres dans la machine, effectuer une opération et recopier le résultat sur papier. Ce va-et-vient est fastidieux et comporte des risques d'erreur à la saisie comme à la transcription du résultat.

Avec le bureau numérique, une calculatrice virtuelle est projetée sur la table de travail. La saisie se fait rapidement et sans risque d'erreur en sélectionnant avec les doigts des nombres présents sur la table. On peut imaginer aller plus loin en proposant à l'utilisateur de prononcer “additionne”, puis de taper du doigt sur les deux nombres et enfin de projeter le résultat à un endroit désigné de nouveau par le doigt. N'importe quel papier gagne ici les capacités d'une calculatrice.

2.2.2. PaperPaint

A moins de s'armer de ciseaux, de colle et d'une photocopieuse, le “couper-coller” adopté par tous les logiciels d'édition, manque cruellement à nos documents papiers. Voici un exemple de manipulation que permet “PaperPaint” :



a b

Figure 2.2 : Copier-coller sur le bureau numérique

Avec son crayon préféré, l'utilisateur dessine un motif sur un papier réel . Il sélectionne ensuite le dessin par deux de ses doigts (figure 2.2a). Le système traduit la sélection en projetant sur le papier un pavé virtuel sur le motif. L’utilisateur peut alors “coller” une projection de ce motif dont il aura spécifié la taille par l’écart de ses doigts (figure 2.2b). Il peut aussi utiliser la fonction de remplissage pour peindre une surface avec un motif.

Cette application montre un exemple d'opérateur virtuel (le couper-coller) sur un opérande réel (le motif papier). L'inverse est également représenté dans PaperPaint : le dessin projeté (opérande virtuel) peut être effacé au moyen d'une gomme (action réelle). Ici, le savoir-faire universel qui consiste à prendre un crayon pour dessiner et une gomme pour effacer est utilisé en remplacement du mode de travail artificiel qu'imposent tous les logiciels de dessin.

2.2.3. Le Double DigitalDesk

Les documents électroniques ont la propriété de voyager à la vitesse de la lumière, c'est ce qui a permis d'ouvrir la voie des collecticiels. D'où l'idée de connecter deux DigitalDesks par réseau. L'image capturée par la caméra de l'un est projetée sur le bureau de l'autre et vice-versa. Deux personnes distantes peuvent alors travailler simultanément sur le même document. Le don d'ubiquité est donné au papier par le biais du réseau informatique.

Nous allons à présent nous intéresser aux problèmes que soulève la réalisation du bureau numérique. Nous commençons par l'analyse des moyens matériels nécessaires.

2.3. Mise en œuvre

Le problème est d'attribuer au bureau physique le potentiel d'une station de travail. Cette amplification commence par l'étude des dispositifs d'affichage et d'acquisition des informations.

2.3.1. Dispositif d'affichage

L'objectif est de restituer en symbiose les documents réels et électroniques. Deux solutions répondent à priori à ce besoin : la projection d'une image électronique par-dessous le bureau au travers d'une surface translucide, ou bien la projection de cette image par-dessus à l'aide d'un vidéo projecteur.

La première solution présente des avantages certains : elle est compacte et intégrée. De ce fait, elle règle la plupart des problèmes de calibrage de la caméra. Ensuite l'image affichée est d'excellente qualité, même si le bureau est touché par une forte source lumineuse. Elle élimine également le problème de l'ombre due au bras ou à la tête de l'utilisateur. Cependant, la projection de l'image sous le bureau interdit le dépot d'objets opaques (papier, livre) qui masqueraient l'écran. De plus, cette technique rend impossible la projection d'information virtuelle sur les documents réels. Dans le contexte de la réalité augmentée, cette incapacité exclut le choix de l'affichage par-dessous. En conséquence, il convient de retenir la projection par-dessus.

Le problème le plus préoccupant avec la projection par-dessus est celui de l'ombre créée par le bras ou le corps de l'utilisateur. Il semblerait toutefois que ce soit un faux problème. Les concepteurs du DigitalDesk ont effectué des tests d'utilisation informelle avec des personnes habituées à travailler

sur des stations de travail classiques. La réaction générale est que le DigitalDesk est agréable à utiliser. En particulier aucun d'eux n'a dit avoir été gêné par les ombres ; il apparaît que les gens sont habitués à “faire avec” l'ombre créée par une lampe au-dessus d'eux et que ce comportement se retrouve avec l'image affichée sur le bureau par le projecteur. Le choix d'une image projetée est donc celui que nous retenons pour notre projet.

2.3.2. Dispositif d'entrée

En dispositif d'entrée, il est possible d’utiliser une table à digitaliser comme dans le système MARCEL [Newman92], ou bien une caméra façon DigitalDesk.

La table à digitaliser procure vitesse et précision mais comme pour l'affichage rétroprojeté, le contact est perdu lorsque le travail s'effectue sur un livre. Elle impose aussi l’usage d'un stylo spécial. Une table tactile lève cette dernière contrainte mais elle interdit de poser les mains sur le bureau. La vision par ordinateur n’a pas ces limitations.

La vision par ordinateur lève les contraintes du point de vue facteurs humains mais l’interprétation des flots d'image est complexe. Les algorithmes sont coûteux en temps de calcul alors que notre objectif est d'obtenir une interprétation en temps réel, c’est-à-dire en accord avec l’attente de l’utilisateur. Néanmoins l'information susceptible d'être extraite est plus riche que la position de pointage d'une table numérique ou d'un écran tactile. Au paragraphe suivant on tente de préciser les capacités requises d’un système de vision pour la mise en œuvre d’applications relevant du bureau numérique.

2.3.3. Capture d'information par caméra

Un problème crucial dans la mise en œuvre du bureau numérique est la capture d'image. Avoir la possibilité de lire un document papier implique que ce document soit fourni en haute résolution à un algorithme de reconnaissance de caractères (Optical Character Recognition). Sur ce point, plusieurs voies ont été explorées :

Dans l’application “Paper User Interface” ([Johnson93]), le scanner est le seul dispositif d'entrée utilisé. L'utilisateur code l'opération à réaliser sur des formulaires qu'il fournit en entrée du système. Il n'y a ensuite aucune difficulté pour effectuer l'OCR sur des zones prédéfinies du document.

Le système MARCEL exploite deux versions du même document [Newman92] : une version haute résolution pré-scannée, et une version basse résolution obtenue par la caméra à la verticale du bureau. Lorsque l’utilisateur sélectionne un paragraphe (par exemple, avec ses doigts), la position du mouvement de sélection est captée et une corrélation est effectuée entre l'image basse résolution du document fournie par la caméra et l'image haute résolution stockée. La zone sélectionnée est alors fournie à un logiciel d'OCR du commerce.

Les solutions que nous venons de présenter ne conviennent pas au contexte dans lequel nous nous sommes placés. Toutes deux forcent l'utilisateur à savoir à priori sur quel document papier il va travailler et lui imposent de les scanner : une manipulation qui fait resurgir une contrainte informatique que, précisément, nous cherchons à éliminer.

La voie empruntée par le DigitalDesk conserve la propriété de non intrusion en utilisant une caméra focalisée sur une zone réduite du bureau. La résolution obtenue est suffisante pour l'OCR mais s'applique sur une zone bien trop réduite pour traiter une page entière. On serait alors tenté d'utiliser une caméra haute résolution mais celles-ci sont encore trop encombrantes et le volume des images obtenues est trop important pour autoriser un traitement en temps réel.

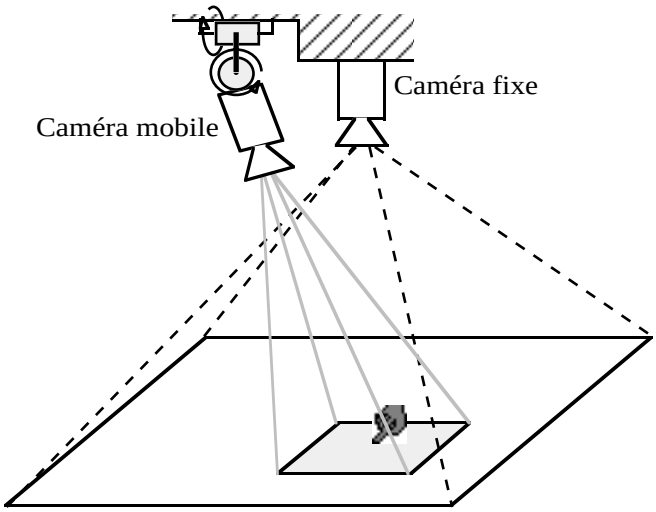


Figure 2.3 : Une installation pour "Vision active".

Il reste une voie à explorer : la vision active. Un système de vision par ordinateur relève de la vision active lorsqu'il contrôle le mouvement et les focales des caméras d'acquisition dans le but de simplifier et d'accélérer la perception visuelle [Crowley94]. Nous appliquons ce principe au cas du bureau numérique. Comme le montre la figure 2.3, un dispositif de type vision active consisterait à monter la caméra mobile "zoom" du DigitalDesk sur une tourelle que piloterait la caméra fixe de "basse résolution". Le système de vision pourrait ainsi obtenir une image haute résolution de n'importe quelle partie du bureau sous réserve qu'il sache recoller les morceaux. Equiper la caméra mobile d'un zoom autofocus controlable par le système de vision offrirait une solution souple au fin dosage entre niveau de résolution et taille de l'image.

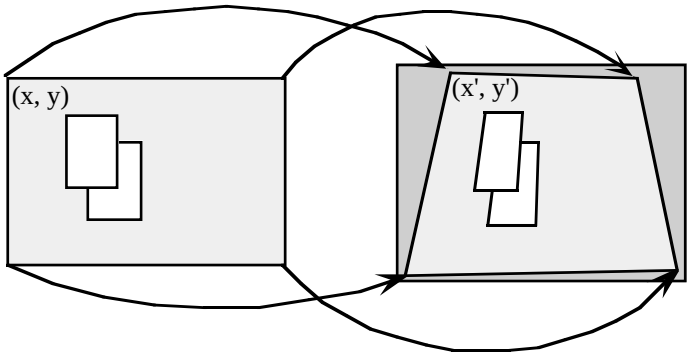
Au vu de notre analyse, nous constatons que la vision par ordinateur occupe une place déterminante dans la réalisation du bureau numérique. Dans le paragraphe suivant, nous énonçons les techniques mises en jeux et nous identifions les performances requises.

2.4. Les problèmes du bureau numérique pour la vision par ordinateur

Les problèmes de vision que soulève la réalisation du bureau numérique sont maintenant présentés : le calibrage, la correction d'image et la reconnaissance d'objet.

2.4.1. Calibrage

Le premier problème qui se pose est une mise en correspondance précise de la position d'un point capturé par la caméra avec sa position sur le bureau. Ce calibrage n'est malheureusement pas immédiat. Il y a deux raisons à cela : les objectifs de la caméra et du projecteur ont deux points de vue différents même si ces deux dispositifs sont rapprochés. Il en résulte deux images de forme différente. La seconde difficulté tient aux changements de configuration des dispositifs dûs aux vibrations ou à un souffle d'air sur le papier. En conséquence, il convient d'appliquer régulièrement un algorithme d'auto-calibrage qui soit rapide et qui ne nécessite pas l'intervention explicite de l'utilisateur.



$$x' = c1.x + c2.y + c3.xy + c4$$
$$y' = c5.x + c6.y + c7.xy + c8$$

Figure 2.4 : Calibrage de la caméra sur l'image projetée.

La figure 2.4 schématise le principe du calibrage du DigitalDesk. Il consiste à projeter quatre croix sur le bureau et à les localiser sur l'image captée par la caméra. Cette opération permet de résoudre le système d'équation de la figure 2.4 et d'en déduire les coefficients c1 à c8. Ensuite, une cinquième croix est localisée pour vérifier qu'elle correspond à la position prédite par le système. Cette technique suppose que l'on sache localiser dans une image une forme prédéfinie simple (par exemple, une croix) et ceci sur un fond quelconque.

L'auteur ne justifie pas les équations de la figure 2.4. Aussi, nous suggérons d'utiliser la forme précise suivante qui correspond à une transformation projective :

$$w.\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,1} & m_{3,2} & 1 \end{bmatrix}.\begin{bmatrix} x \\ y \\ 1 \end{bmatrix},$$

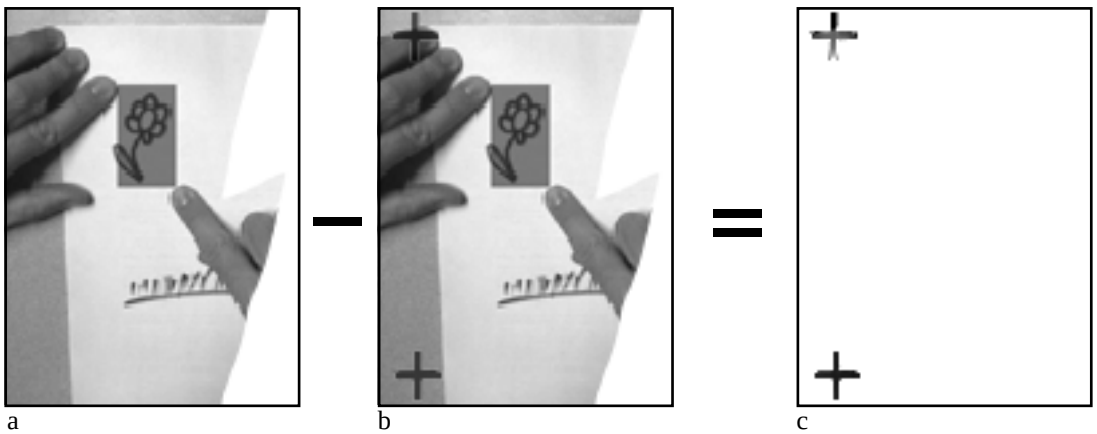
$$\text{soit } x' = \frac{m_{1,1}.x + m_{1,2}.y + m_{1,3}}{m_{3,1}.x + m_{3,2}.y + 1},\ y' = \frac{m_{2,1}.x + m_{2,2}.y + m_{2,3}}{m_{3,1}.x + m_{3,2}.y + 1}.$$

Une solution efficace pour implémenter cette transformation est de la calculer une fois pour toute dans une table pour chacun des points de l'image projetée. On remarque que les deux objectifs (caméra et projecteur) sont assez proches et que la distance des objectifs au bureau est largement supérieure à leur distance focale. On peut donc supposer qu'une transformation afine suffise pour atteindre la précision désirée et l'on pourra tester le calibrage avec la forme simplifiée suivante :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \\ 0 & 0 & 1 \end{bmatrix}.\begin{bmatrix} x \\ y \\ 1 \end{bmatrix},$$

$$\text{soit } x' = m_{1,1}.x + m_{1,2}.y + m_{1,3},\ y' = m_{2,1}.x + m_{2,2}.y + m_{2,3}.$$

Quant à la localisation des symboles pour la détermination des coefficients, l'utilisation d'une image de différence semble judicieuse : on capture une image du bureau sans les croix (figure 2.5a), puis le plus rapidement possible on affiche les croix et on capture une nouvelle image (figure 2.5b). Si l'intervalle entre les deux prises est assez court, seuls les symboles appaîtront sur la différence des deux images (figure 2.5c).



a b c

Figure 2.5 : Localisation de repères sur le bureau pour calibration.

2.4.2. Correction d'image

Pour exploiter les informations contenues dans les documents papier, il faut pouvoir exécuter un logiciel d'OCR sur une image de ces documents. Les trois étapes schématisées par la figure 2.6 devront être effectuées.

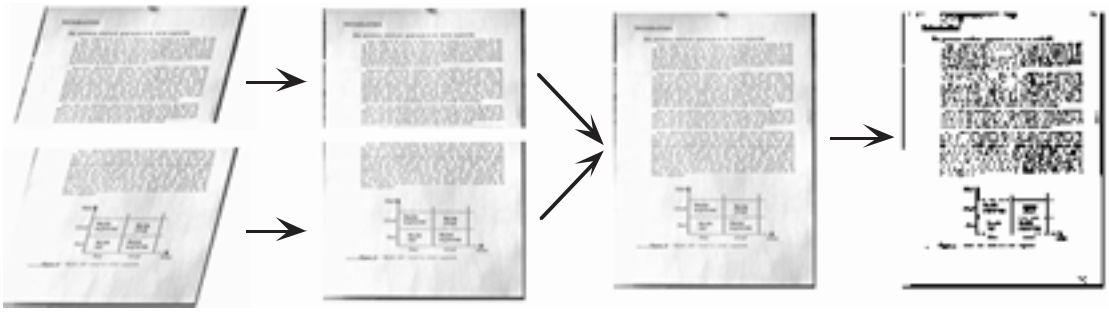


Figure 2.6 : Traitements de l'image pour la reconnaissance optique

L'inclinaison des images due à l'angle que forment le projecteur et la caméra par rapport à la verticale doit être corrigée afin de reconstituer une image complète du document. On effectuera ensuite un seuil adaptatif pour obtenir l'image binaire nécessaire à la reconnaissance optique des caractères. La plupart des algorithmes évoqués ici font encore l'objet de recherche et sortent du cadre de ce projet de DEA.

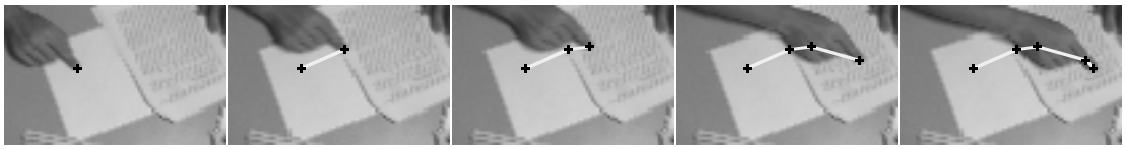
2.4.3. Reconnaissance d'objet

Le scénario du "PaperPaint" présenté précédemment implique que le système de vision soit capable de reconnaître le type d'outil emprunté par l'utilisateur. Le problème de la reconnaissance en vision est un vaste domaine de recherche mais les besoins sont ici cernés par le nombre d'objets susceptibles d'être identifiés : l'utilisateur aura l'occasion de se servir de ses doigts, de stylos ou de gommes. Le processus de reconnaissance peut être encore simplifié si l'on accepte le fait que l'utilisateur se sert globalement toujours des mêmes outils et qu'une phase d'apprentissage est demandée à l'initialisation du système.

Après avoir évoqué quelques problèmes généraux, nous consacrons le chapitre suivant au sujet que nous avons choisi d'étudier en détail dans le cadre de ce DEA : le suivi d'objets sur image 2D.

Chapitre 3

LE SUIVI D'OBJETS



Le suivi d’objets fait parti des thèmes essentiels de la recherche en vision par ordinateur. Il est le sujet d’une multitude de publications majoritairement centrées sur les problèmes canoniques : suivi en mono ([Meyer92], [Koller92]) ou stéréo vision ([Rehg93]) ; suivi d’un motif ([Meyer92]), d’un modèle géométrique ([Harris92], [Koller92], [Rehg93]), ou d’un contour actif ([Kass87], [Terzopoulos92], [Williams90], [Uedea92]). Certaines recherches portent tout de même sur des applications précises du suivi comme le contrôle du trafic automobile ([Koller92], [Koller93]), la reconnaissance des gestes et même le suivi des mains ([Rehg93]) comme outil d’interaction en IHM. Cependant, à l’heure actuelle, il n’existe aucune étude sur l’utilisation du suivi pour le bureau numérique. C’est précisément l’objet de ce chapitre qui définit avec précision les besoins de ce type d’application et les capacités du suivi requises pour y répondre.

Notre objectif est de connaître en temps réel la position sur le bureau d’un objet de pointage, essentiellement un doigt, un stylo ou une gomme. On cherche à modifier au minimum l’environnement naturel de l’utilisateur : l’emploi de gants ou de marquages spécifiques sur les objets à suivre est proscrit. Notre seule source d’information est un flot d’images en 256 niveaux de gris provenant d’une caméra.

Nous procédons comme suit : le choix de la restriction du suivi à deux dimensions est d’abord évoqué (paragraphe 3.1). Ensuite, au paragraphe 3.2, les contraintes de temps réel sont précisées en fonction du rôle de l’outil d’interaction que l’on cherche à concevoir. Puis, en 3.3 et 3.4, nous analysons deux points des techniques de suivi que nous jugeons pertinents dans le contexte du bureau

numérique : la zone de recherche et l'initialisation. En 3.5, nous identifions les critères qui nous permettront d'évaluer les diverses techniques étudiées.

3.1. Choix entre un suivi 2D ou 3D

Lorsque la tâche à réaliser s'effectue dans l'espace, le suivi de la main en trois dimensions par stéréovision ou l'utilisation d'une "flying mouse" permettent de désigner directement un point 3D. Cependant, l'utilisateur doit rester bras tendu et sans support pendant toute la durée de la manipulation ce qui a évidemment tendance à le fatiguer. De plus, la désignation est moins précise que si la main repose sur un bureau. Cette remarque nous vient de Krueger ([Krueger91]). C'est pourquoi il propose avec son *VideoDesk* de commencer par définir un plan ayant n'importe quelle orientation dans l'espace. Le travail s'effectue ensuite dans ce plan où une corrélation est faite entre la position du curseur et celle de la main qui repose sur le bureau.

Outre cela, nous avons à réaliser un suivi qui interviendra dans le contexte du bureau. Que celui-ci soit réel ou virtuel, l'espace de travail est essentiellement 2D. Tout ceci nous amène à restreindre notre étude aux techniques de suivi fonctionnant en deux dimensions, ce qui n'est pas vraiment une contrainte puisque la plupart des techniques de suivi 3D s'adaptent assez directement au 2D. Les problèmes spécifiques à la stéréovision tels que l'occlusion, la mise en correspondance de segments, et le traitement de deux flots d'image ne seront pas à traiter. En conséquence, nous concentrons l'effort sur la vitesse du suivi, sa précision et sa robustesse.

Cependant le choix du 2D pose le problème de la détection du contact et de la levée du contact avec la surface du bureau. En particulier comment reproduire le clic de sélection ? L'idée qui vient naturellement à l'esprit est d'exploiter une tape du doigt sur le bureau : pour lancer une application, il suffit comme en manipulation directe de taper deux fois rapidement sur la projection de son icône sur le bureau. Or ce geste intervient principalement dans la dimension verticale au bureau, donc de manière invisible à une caméra placée à la verticale. Plusieurs idées ont été émises pour tenter de résoudre ce problème.

Dans la réalisation actuelle du *DigitalDesk*, un micro est installé sous le bureau qui détecte le bruit émis par le doigt lorsqu'il est en contact avec la table de travail. Nous avons vérifié que ce procédé est efficace en l'améliorant par l'utilisation d'une caisse de résonnance : le micro est collé à l'intérieur d'une boîte sur laquelle on simule un double clic, un simple clic, et le fait de "tirer"[1] une icône.

[1] À la souris, on tire un objet en cliquant dessus et en laissant le bouton enfoncé pendant qu'on le déplace.

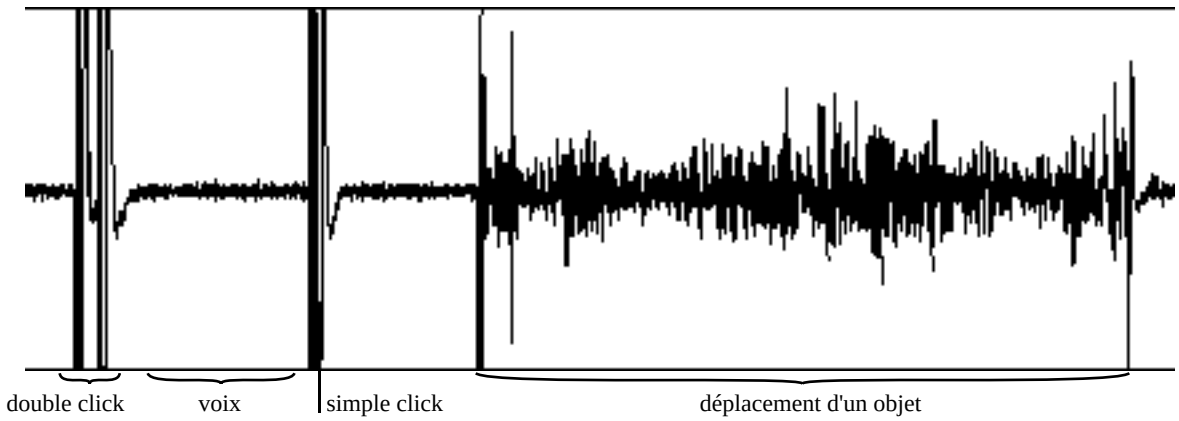


Figure 3.1 : Signal capté par un micro "intégré au bureau".

La figure 3.1 montre le graphe du niveau du signal capté. Les clics (simples ou doubles) sont effectivement mis en évidence, de même que les périodes durant laquelle le doigt se déplace en contact avec le "bureau". De plus, le fait de parler pendant la manipulation n'a absolument pas été gênant, ceci grâce à la caisse de raisonnance qui joue en même temps le rôle d'isolant du monde extérieur. Cette solution présente cependant deux faiblesses : en premier lieu, les bruits provoqués par la pose d'objets lourds sur le bureau peuvent interférer avec les clics. Ensuite, le signal est perdu lorsque l'utilisateur se sert de son doigt sur un livre posé sur le bureau. C'est pourquoi il est nécessaire de trouver un meilleur équivalent au clic de la souris.

Lorsqu'il a conçu son *VideoDesk*, Krueger a pris conscience de ce problème et a imaginé que le clic serait remplacé par une pression du pouce sur l'index. Un simple suivi du bout de l'index comme curseur ne serait plus suffisant et c'est tout une partie de la forme de la main qu'il faudrait analyser. De plus, il s'agit d'un geste conventionnel qu'il faudrait apprendre.

En fait, la solution est peut-être d'utiliser une deuxième caméra sans pour autant réaliser un véritable suivi en trois dimensions. Son rôle est très limité : donner la position haut/bas de l'objet suivi. La résolution d'image nécessaire est faible car il n'est pas nécessaire de connaître précisément la hauteur du doigt. Un algorithme simple basé sur les images de différences est envisageable puisque lever ou baisser le doigt est un mouvement dynamique.

En conclusion, il semble suffisant pour le bureau numérique de réaliser un suivi en deux dimensions (ou bien doit-on parler de 2D 1/2 en cas d'utilisation d'une deuxième caméra ?). Il est alors plus crédible d'atteindre le "temps-réel", objectif qui va être précisé dans le paragraphe suivant.

3.2. Le facteur vitesse

La finalité du suivi est la réalisation d'un outil de pointage d'une interface homme-machine. Sa vitesse de fonctionnement doit donc respecter certains seuils au-dessous desquels le résultat est saccadé et inexploitable comme outil d'interaction. Afin de produire une estimation de la vitesse

requise pour le suivi, nous reprenons le modèle du processeur humain défini par [Card83] et présenté dans [Coutaz90]. La figure 3.2 extraite de [Card83] en montre les éléments essentiels. Nous nous intéressons en particulier aux temps de cycle des trois processeurs : perceptuel, cognitif et moteur. Pour chacun, trois estimations sont données sous la forme $\tau = x[y - z]$ avec $y \le x \le z$. La valeur x correspond à un temps de traitement moyen, y au maximum et z au minimum.

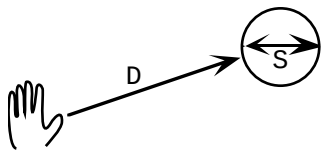
3.2.1. Fréquence de fonctionnement

La première estimation que l'on obtient grâce à ce modèle est la fréquence de rafraîchissement requise. Le temps d'exécution du cycle du processeur perceptuel est estimé en moyenne à τ_p = 100 ms = 0,1 s . Il faut donc une fréquence supérieure à $1/\tau_p$ = 10 Hz afin de parvenir à l'effet de l'animation. En pratique, il est préférable d'utiliser le temps de cycle minimum du processeur perceptuel égal à 50 ms (soit une fréquence de 20 Hz) pour que le mouvement apparaisse comme réellement fluide.

3.2.2. Vitesse de la cible

Ensuite, il nous faut estimer la vitesse maximale de déplacement de la main afin de connaître les performances requises pour l'algorithme de suivi. Dans le modèle du processeur humain, le déplacement global est en fait la somme de plusieurs petits déplacements correspondant à un cycle : observation de la position courante (processeur perceptif), calcul de la correction de la trajectoire pour atteindre la cible (processeur cognitif), déplacement effectif (processeur moteur). À chaque cycle la distance à la cible est réduite d'un facteur ε estimé à 0,07.

De ce modèle du déplacement, on peut dériver le temps nécessaire pour déplacer la main d'une distance D afin d'atteindre une cible de taille S. Cette formule, connue sous le nom de la loi de Fitts, peut s'écrire sous la forme :



$$T = I_M \log_2(2D / S), \text{ avec}$$

$$I_M = -\frac{\tau_P + \tau_C + \tau_M}{\log_2(\varepsilon)}$$

À l'aide des temps donnés par le modèle du processeur humain, on calcule $I_M = 63[27 - 122]$ ms. L'estimation de la vitesse de déplacement de la main est donnée par :

$$v = \frac{D}{I_M . \log_2(2D / S)}$$

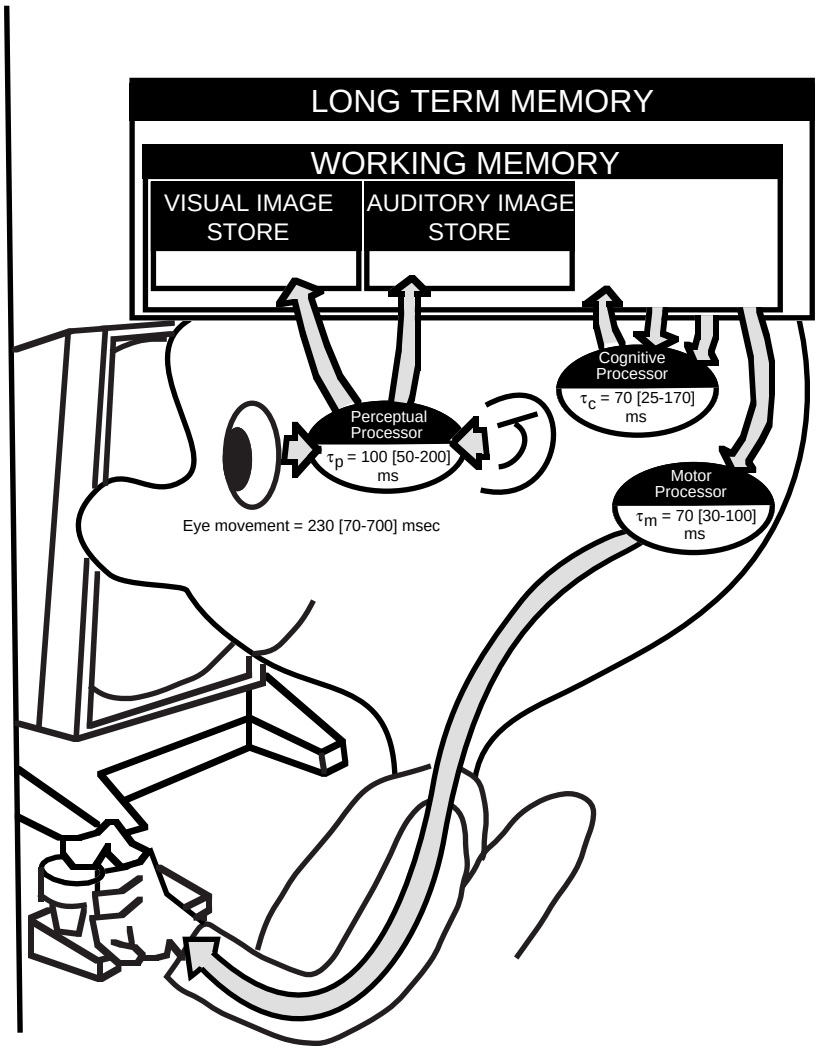


Figure 3.2 : Estimation du temps de cycle des processeurs perceptuel, cognitif et moteur dans le modèle du processeur humain (d'après [Card 83]).

Celle-ci est d'autant plus grande que la cible est loin et petite. Dans le cas du bureau numérique, la diagonale capturée par la caméra mesure environ 70 cm. Les objets à manipuler (icônes) ont une taille de l'ordre de 2 cm. Ceci nous permet de calculer l'estimation de la vitesse de déplacement maximale de la main de l'utilisateur :

$$v = 1,81[0,94 - 4,22] \text{ m/s}$$

3.2.3. Validation expérimentale

Afin d'obtenir une validation expérimentale de ce résultat, nous avons réalisé une expérience simple qui consiste à filmer un mouvement rapide de la main sur le bureau (en l'occurence un va-et-vient rapide), puis à déterminer "manuellement" sur chacune des images du film la position du doigt dont on peut ainsi calculer vitesse et accélération. Le film est au format QuickTime [Apple93] qui a l'avantage de garantir une fréquence régulière de prise de vue. Dans le cas de l'expérience, les images

ont une taille de 192x144 (quart d'image au format PAL) ce qui permet d'enregistrer à la fréquence de 24 images par secondes, soit un intervalle entre les images de $\Delta_t = 1/24 = 41,7\text{ms}$. La taille des pixels est connue en mesurant sur le bureau la hauteur (32 cm) et la largeur (40 cm) de la zone capturée par la caméra.

On calcule alors

$$x_{cm} = 40/192 * (x_{pixels} - 80)$$
$$y_{cm} = 32/144 * (y_{pixels} - 410)$$

Ce qui permet de tracer l'échantillonnage de la position du doigt présenté dans la figure 3.3. La norme de la vitesse est calculée par :

$$v_i = \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{\Delta_t}$$

et la norme de l'accélération par :

$$\gamma_i = \frac{v_i - v_{i-1}}{\Delta_t}$$

La tableau de la figure 3.4 résume les résultats de cette expérience.

Nous constatons que dans le cas du mouvement de la main sur le bureau, le modèle du processeur humain a tendance à surestimer la vitesse du doigt puisque lors de l'expérience, le geste a été effectué de manière très rapide et sans la contrainte d'atteindre une cible. Or la vitesse maximale atteinte est de 1,39 m/s, là où la loi de Fitts prédit des vitesses de l'ordre de 1,8 m/s pour un individu moyen et de l'ordre de 4,2 m/s pour un individu rapide.

Dans la figure 3.4, nous mettons en évidence les fortes accélérations (jusqu'à 17 $m.s^{-2}$) que le doigt peut effectuer lors d'un changement brutal de direction. Ces résultats ont un impact direct sur la stratégie des algorithmes de suivi que nous évoquons au paragraphe suivant.

3.3. Zone de recherche

L'heuristique principale des algorithmes de suivi affirme que la position de l'objet traqué dans une nouvelle image est "proche" de sa position dans l'image précédente. C'est la conséquence directe de la continuité du mouvement des objets suivis. La proximité des positions entre deux images consécutives de la cible est fonction de deux facteurs : les positions sont d'autant plus proches que la vitesse de déplacement apparente de l'objet est faible et que la fréquence de prise d'images est élevée.

3.3.1. Vitesse

Supposons que l'on puisse borner à la fois la vitesse de déplacement maximum de l'objet suivi (soit V_m cette borne supérieure) et le temps maximum Δ_t entre deux prises de vue. Alors on peut

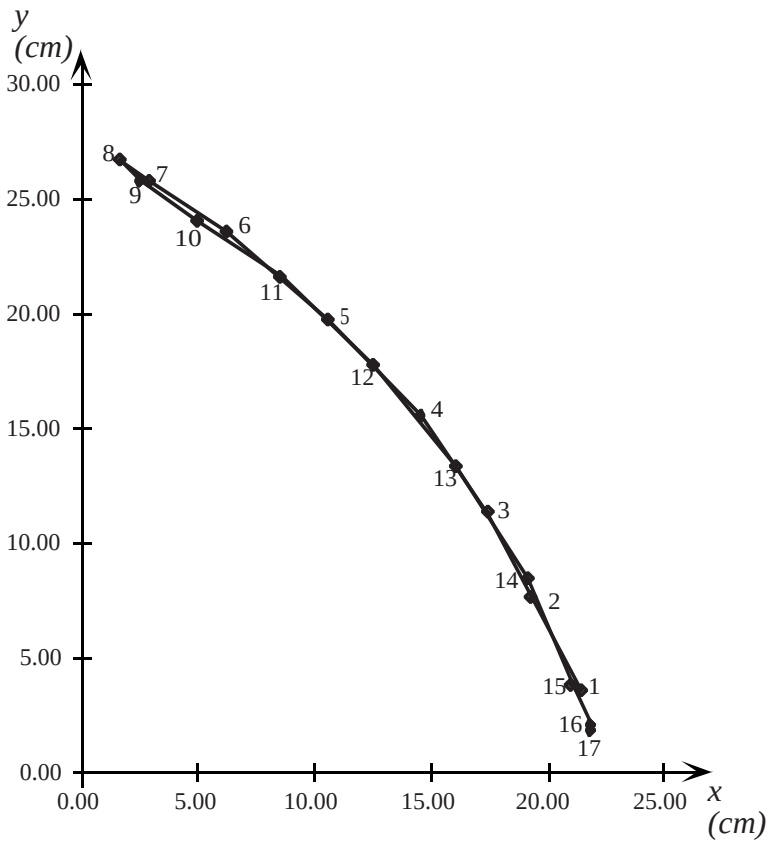


Figure 3.3 : Echantillonnage des positions du doigt dans un geste de va-et-vient rapide sur le bureau.

Image	x (pixels)	y (pixels)	x (cm)	y (cm)	dx (cm)	dy (cm)	v (m/s)	dv (m/s)	γ ($m.s^{-2}$)
1	183	426	21.46	3.56					
2	173	444	19.38	7.56	-2.08	4.00	1.08		
3	164	461	17.50	11.33	-1.88	3.78	1.01	-0.07	-1.68
4	150	480	14.58	15.56	-2.92	4.22	1.23	0.22	5.27
5	131	499	10.63	19.78	-3.96	4.22	**1.39**	0.16	3.78
6	110	516	6.25	23.56	-4.38	3.78	**1.39**	0.00	-0.04
7	94	526	2.92	25.78	-3.33	2.22	0.96	-0.43	-10.22
8	88	530	1.67	26.67	-1.25	0.89	0.37	-0.59	-14.24
9	92	526	2.50	25.78	0.83	-0.89	0.29	-0.08	-1.82
10	104	518	5.00	24.00	2.50	-1.78	0.74	0.44	10.65
11	121	507	8.54	21.56	3.54	-2.44	1.03	0.30	7.12
12	140	490	12.50	17.78	3.96	-3.78	1.31	0.28	6.73
13	157	470	16.04	13.33	3.54	-4.44	1.36	0.05	1.22
14	172	448	19.17	8.44	3.13	-4.89	**1.39**	0.03	0.69
15	181	427	21.04	3.78	1.88	-4.67	1.21	-0.19	-4.45
16	185	419	21.88	2.00	0.83	-1.78	0.47	-0.74	**-17.66**
17	185	418	21.88	1.78	0.00	-0.22	0.05	-0.42	-10.03

Figure 3.4 : Vitesse et accélération maximales exécutées par le doigt.

définir une zone de recherche maximum en dehors de laquelle il est impossible de trouver la cible : c'est un cercle centré sur la dernière position connue de la cible, de rayon $R = V_m.\Delta_t$ (cf. figure 3.5).

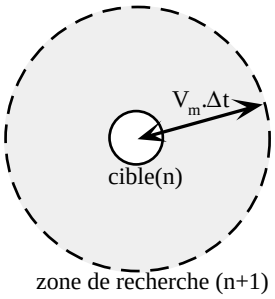


Figure 3.5 : Détermination de la zone de recherche en fonction de la position et de la vitesse maximale de la cible.

3.3.2. Accélération

On peut encore rétrécir la zone de recherche dans l'image numéro n+1 si on connaît dans l'image numéro n à la fois la position et le vecteur vitesse $\vec{V}(n)$ de la cible ainsi que l'accélération maximale γ_m qu'elle peut effectuer (cf. figure 3.6). Dans le pire des cas, entre les deux prises de vue la cible subit une accélération γ_m pendant Δ_t. Son déplacement peut être décomposé en un mouvement à accélération nulle (vitesse constante = $\vec{V}(n)$), puis en un mouvement départ arrêté à accélération constante (= γ_m). La recherche dans l'image numéro n+1 s'effectue dans le cercle dont le centre est le translaté de la cible dans l'image n par $\vec{V}(n).\Delta_t$ et le rayon est $\gamma_m.\Delta_t^2$.

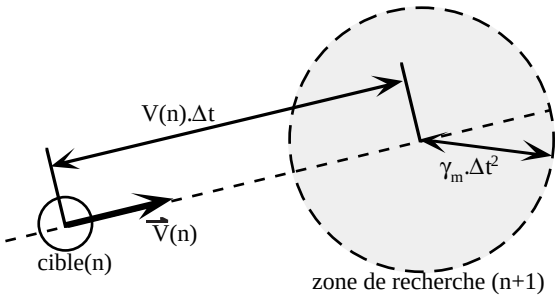


Figure 3.6 : Détermination de la zone de recherche en fonction des position, vecteur vitesse et accélération maximale de la cible.

3.3.2. Comparaison

Cependant, la prise en compte de l'accélération n'est avantageuse que dans le cas où le rayon du disque de recherche suivant le second principe ($\gamma_m.\Delta_t^2$) est plus petit que celui du premier principe ($V_m.\Delta_t$), c'est-à-dire lorsque :

$$\gamma_m.\Delta_t^2 < V_m.\Delta_t, \text{ soit}$$

$$\Delta_t < \frac{V_m}{\gamma_m}$$

Dans le cas du suivi de la main sur le bureau, en considérant les résultats du tableau de la figure 3.4, on trouve une vitesse maximale de 1,39 m/s et une accélération maximale de 17,7 m.s^{-2}. Prendre en compte l'accélération n'est valable que si

$$\Delta_t < \frac{1,39}{17,7} = 0,0786\,\text{s}.$$

Ce qui veut dire que la fréquence de l'algorithme de suivi doit être supérieure à $\frac{1}{0,0786} = 12,7$ Hz. Dans les différentes mises en œuvre qui vont suivre, nous utiliserons le premier principe fondé uniquement sur la vitesse : les fréquences de fonctionnement atteintes sont de l'ordre de 10 Hz. Or, prendre en compte l'accélération met en jeux des calculs qui auraient pour conséquence de faire baisser cette fréquence.

Nous venons de voir que pour déterminer la nouvelle position, le suivi s'appuie sur la position précédente de l'objet. Le paragraphe suivant pose le problème de l'initialisation où la position de l'objet à suivre est encore inconnue.

3.4. Initialisation

Les algorithmes de suivi d'objet permettent de trouver la nouvelle position à partir d'une position courante. Il est donc nécessaire à l'initialisation de leur fournir la première position de la cible. Le plus simple est de laisser l'utilisateur placer celle-ci à un endroit prédéfini connu de l'algorithme de suivi qui est ensuite déclenché "manuellement". C'est le choix adopté par exemple par [Rehg93].

Cette solution n'est pas acceptable dans certains contextes, et en particulier pour le bureau numérique : intrusion d'une tâche parasite dans l'activité de l'utilisateur. Il existe plusieurs méthodes de détection plus ou moins automatiques de la présence de la cible dans le champ de la caméra. La complexité et le coût en temps de calcul varient suivant les méthodes, et toutes ne sont pas adaptables à toutes les familles d'algorithme de suivi. Aussi leur faisabilité sera-t-elle discutée dans chacun des paragraphes sur le suivi par recherche de motif, par contour actif et par suivi d'un modèle. Toutefois nous évoquons ici leur principe.

3.4.1. Par détection localisée du mouvement

Un premier pas vers la détection automatique est possible en lançant l'algorithme de suivi lorsque le doigt de l'utilisateur fait irruption dans un cadre prédéfini. Une manière de réaliser ceci est d'effectuer en permanence une différence entre deux images consécutives de l'intérieur du cadre. Lorsque la somme des valeurs de pixels de ces images de différence dépasse un certain seuil, c'est que l'image a varié. On suppose que c'est le doigt qui a fait irruption dans le cadre et le suivi sur le

contenu du cadre peut être lancé. Nous avons expérimenté ce principe que nous évaluons dans le chapitre 5 sur le suivi par recherche de motif.

3.4.2. Par détection globale du mouvement

On peut aussi automatiser complètement l'initialisation du suivi en évaluant le mouvement sur la totalité des images capturées par la caméra. Si nous supposons que le bras de l'utilisateur est le seul objet en mouvement sur le bureau, alors c'est sa forme qui apparaît sur l'image calculée (cf. figure 3.7 pour un exemple de mise en évidence de la main par image de différence). Une phase d'analyse est encore nécessaire pour localiser le bout du doigt et lui attacher le suivi. Détecter le mouvement afin d'initialiser le suivi est utilisé par [Koller93] dans son suiveur multiple d'automobiles. La détection des objets en déplacement y est effectuée en analysant la différence entre l'image courante et une image de "fond" mise à jour en permanence pour prendre en compte les variations lumineuses.

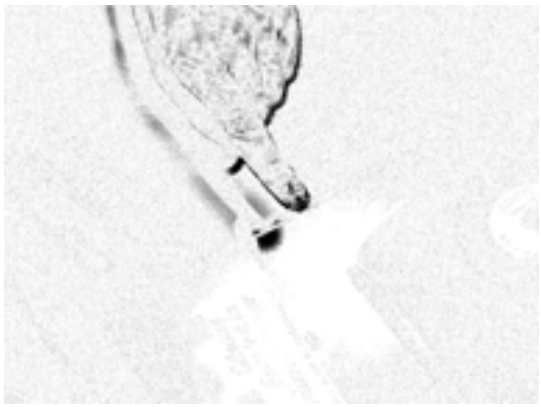


Figure 3.7 : Mise en évidence de la main en déplacement par images de différence.

Cependant, il est possible que l'automatisation totale de l'initialisation du suivi ne soit pas nécessaire : l'utilisateur voudra aussi se servir de ses mains sans pour autant les utiliser comme outil de pointage. Dans la première technique évoquée, le passage du doigt sur une zone prédéfinie indique de manière explicite que l'on veut utiliser le suivi. (On retrouve le même problème en reconnaissance de la parole et en papier électronique : il convient d'indiquer au système le déclenchement de l'interprétation. L'avantage pour le système est la fiabilté, l'inconvénient pour l'utilisateur est l'introduction d'une tâche parasite.)

3.4.3. Par extension de la zone de recherche

Il existe aussi une solution qui a l'avantage de ne pas requérir d'algorithme spécifique : seul l'algorithme du suivi est utilisé. Il s'agit à l'initialisation d'effectuer la recherche du doigt dans la totalité de l'image plutôt que dans une zone de recherche réduite. Ceci n'est possible que pour certains algorithmes de suivi possédant un modèle de la cible et n'utilisant sa position courante que pour

accélérer la recherche de la position suivante. Le temps de recherche sur la totalité de l'image étant relativement long (au minimum quelques secondes), il est fort probable que lorsque la recherche aboutit le doigt de l'utilisateur se soit largement éloigné de sa position initiale! Celle-ci ne présente alors plus d'intérêt pour réduire la zone de recherche du suivi. Pour résoudre ce problème, il suffit que l'utilisateur en soit conscient et laisse son doigt immobile dans le champ de la caméra jusqu'à ce qu'un signal (l'apparition d'un curseur par exemple) le prévienne que le suivi est opérationnel. On règle du même coup le problème du déclenchement explicite du suivi évoqué précédemment. Cette proposition mérite néanmoins d'être confirmée par des tests d'utilisabilité.

Jusqu'à présent, nous avons centré l'analyse sur la définition du suivi d'objet dans une optique d'utilisation pour le bureau numérique. Au paragraphe suivant, nous identifions les caractéristiques qui nous permettront de comparer les deux techniques de suivi que nous envisageons d'étudier.

3.5. Critères de qualité

Le choix d'une technique de suivi pour la réalisation du bureau numérique commence par l'analyse des solutions qu'elle apporte aux problèmes évoqués plus haut. A présent, nous étudions des "critères de qualité" servant d'éléments de comparaison entre les solutions offertes : robustesse, facilité d'adaptation à différentes formes, traitement des échecs et simplicité d'intégration.

3.5.1. Robustesse

Nous traduisons le critère de robustesse d'un algorithme de suivi d'objet par sa tolérance aux rotations, sa résistance aux variations de l'éclairage et sa tolérance à l'encombrement du bureau. Nous reprenons chacun de ces points.

La technique doit tolérer une totale liberté de rotation de l'objet suivi dans le plan du bureau (angle α de la figure 3.8) et des rotations de l'ordre de 70 degrés hors de ce plan (angle β de la figure 3.8). Un angle β supérieur entraîne l'occlusion de la cible par la main, donc la perte du suivi ; il n'est donc pas nécessaire de prévoir ce cas.

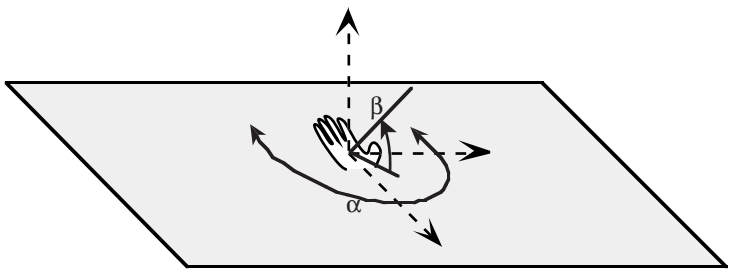


Figure 3.8 : Rotation du doigt dans le plan du bureau (α) et hors de ce plan (β).

Ensuite, nous ne voulons pas imposer d'environnement lumineux particulier autour du bureau. Il est fréquent de trouver de fortes variations lumineuses à sa surface en raison des ombres portées par les montants d'une fenêtre par exemple, ou par l'utilisation d'une lampe dont l'abat-jour crée une ombre franche. La luminosité de la cible va donc varier en fonction de sa position. Une tolérance du suivi par rapport aux variations de la luminosité est nécessaire.

Enfin, l'objet à suivre est destiné à se déplacer au-dessus d'un bureau nu ou encombré d'objets quelconques et de feuilles de papier au contenu indéfini. Le suivi doit donc autoriser l'utilisation d'un fond bruité.

3.5.2. Facilité d'adaptation à différentes formes

La technique de suivi à retenir doit pouvoir suivre la forme de l'extrémité d'un doigt, d'un stylo et d'une gomme. La structure de ces trois formes étant assez proche (cf. figure 3.9), ce critère devrait être satisfait assez aisément.

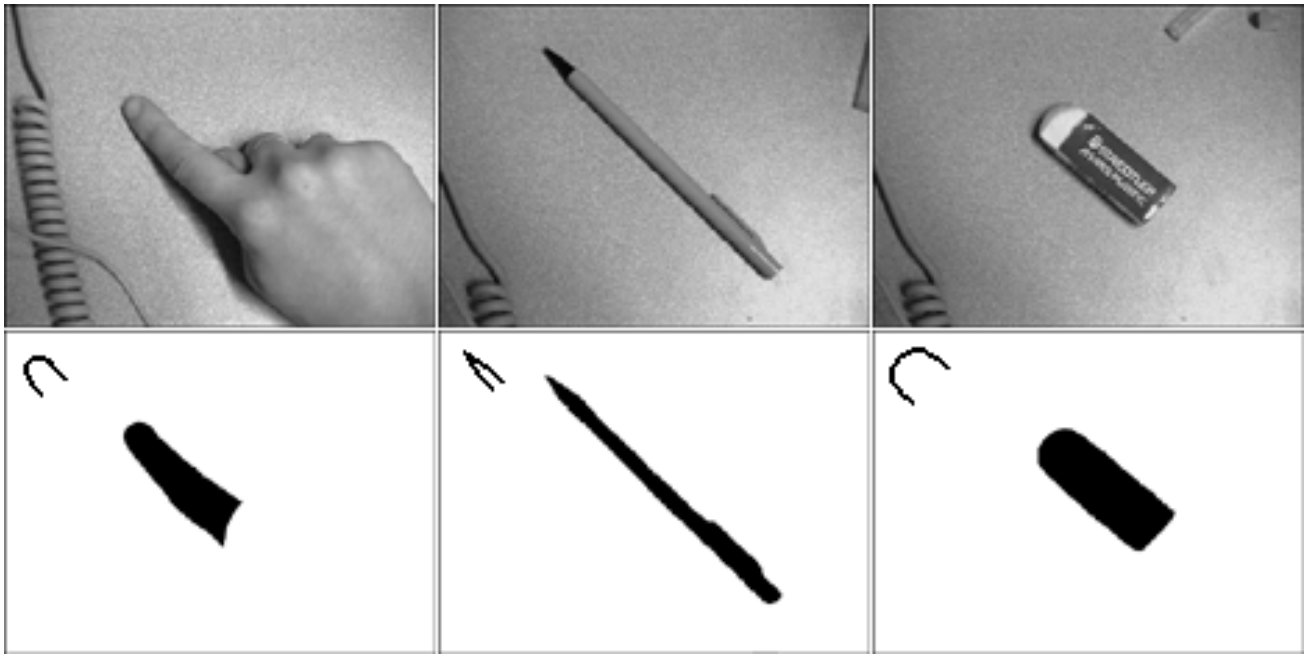


Figure 3.9 : Les trois pointeurs principaux du bureau numérique (photo, forme et contour d'extrémité)

Cependant il faut prévoir l'ouverture de l'algorithme de suivi au cas de nouveaux objets (qui peuvent avoir un rôle différent du pointeur). Par exemple, on voudra connaître la position d'une feuille de papier pour pouvoir y projeter un contenu virtuel quand l'utilisateur la déplace. Il convient donc de considérer la capacité d'adaptation d'une technique de suivi à une variété de formes.

3.5.3. Traitement des échecs

Aucun des algorithmes de suivi actuels ne peut garantir un suivi parfaitement fiable. On sera donc amené à comparer leurs capacités à détecter la perte de la cible. S’il y a possibilité de détection précoce, on examinera l’aptitude à entrer dans une procédure d’exception qui tente de rétablir une situation fiable avant la perte.

Si le suivi n’a pas de moyen de prévenir une perte imminente, il est tout de même nécessaire de détecter l’échec afin de réagir soit en entrant de nouveau dans une phase d’initialisation si celle-ci est automatisée (voir le paragraphe 3.4), soit en avertissant l’utilisateur (par exemple disparition du curseur, ou réapparition de la boîte d’initialisation du suivi) afin que celui-ci réinitialise lui-même le suivi.

La fin de ce chapitre conclut le “cahier des charges” des pré-requis de la technique de suivi destinée à la réalisation du bureau numérique. Nous présentons maintenant l’application que nous avons développée nous permettant de disposer d’un banc de tests comparatifs.

Chapitre 4

NOTRE BANC D'ESSAIS : FINGERPAINT

Au cours des trois premiers chapitres de ce rapport, notre étude est restée entièrement théorique. Dans la suite du document, nous présentons un ensemble de mises en oeuvre expérimentales réalisées afin d'illustrer et de valider cette étude. La programmation s'est déroulée au sein d'une application qui répond à un double objectif :

- fournir un banc d'essais permettant de tester et comparer les deux techniques de suivi que nous avons choisi d'expérimenter,
- conforter l'intérêt que nous portons au bureau numérique en créant un système permettant de tester sa caractéristique principale : la manipulation d'objets virtuels à la main.

Un logiciel de dessin au doigt est le thème de cette application et lui a donné son nom : FingerPaint.

Dans ce chapitre, nous présentons l'environnement matériel qui a servi de support à nos expériences ; puis nous donnons une description des fonctions de FingerPaint et des manipulations qu'elle nous a permis de mettre à l'épreuve.

1. Environnement de développement

Le noeud central du système est un Macintosh 840av équipé d'un processeur 68040 à 40 MHz. Si la puissance du processeur est modeste, la machine offre en revanche plusieurs particularités qui font d'elle un excellent support pour le système du bureau numérique. Elle intègre par défaut :

- une carte d'acquisition vidéo qui fournit également une sortie composite autorisant un affichage par vidéoprojection,

- un processeur de traitement du signal, qui permettra d'accélérer considérablement les algorithmes de vision,
- un système de reconnaissance de la parole (PlainTalk). Cet outil permettra de tester la coopération entre la voie et le geste.

L'affichage se fait sur écran au mur par l'intermédiaire d'un rétroprojecteur et d'un datashow (écran translucide permettant la rétroprojection d'une image vidéo). L'image est capturée par une caméra posée sur trépied dirigée vers le mur. Cette installation est schématisée sur la figure 4.1.

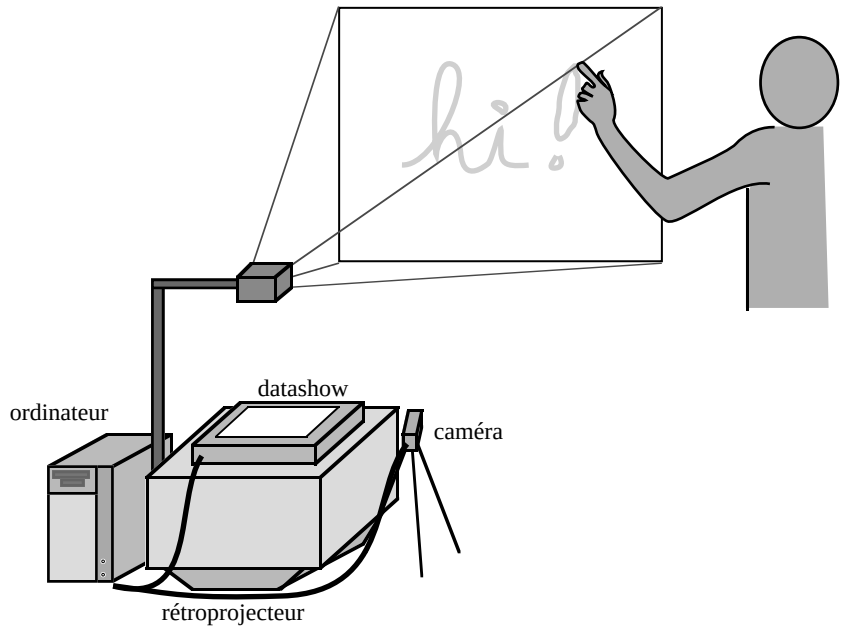


Figure 4.1 : L'installation de FingerPaint.

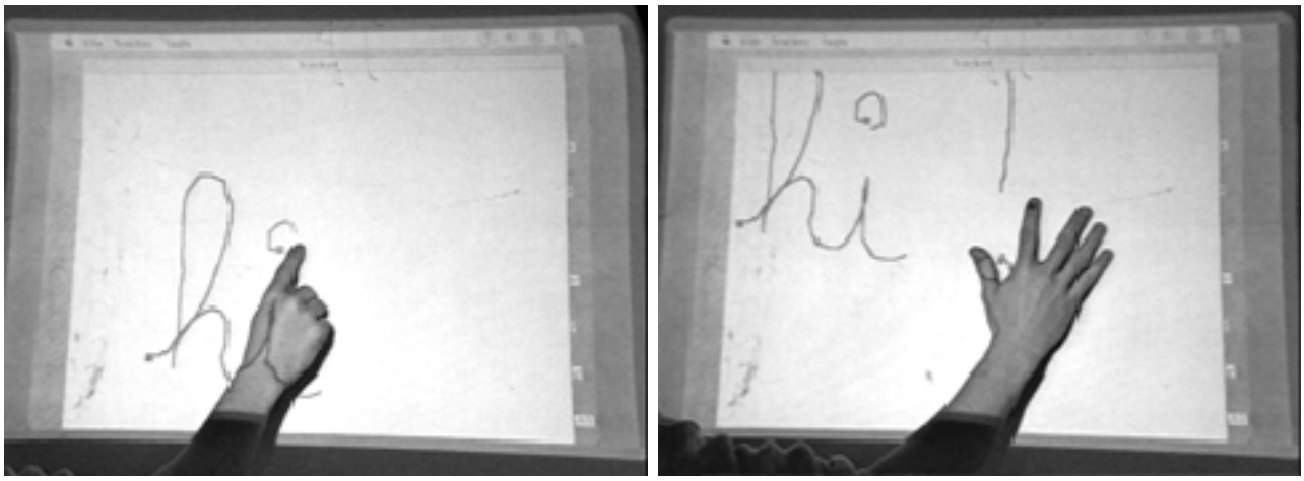
Cette installation propose un plan de travail vertical alors que celui du bureau numérique est horizontal, cependant le mode d'interaction est similaire dans les deux cas. Elle constitue un prototype nous permettant de tester les fonctions du bureau numérique.

2. Utilisation de FingerPaint

FingerPaint est une application qui permet à l'utilisateur de dessiner avec une encre virtuelle sur un mur. L'utilisateur désigne le pointeur qu'il veut utiliser (son doigt, un stylo, ou n'importe quel objet) durant la phase d'initialisation. Ensuite le programme se trouve en permanence dans l'un des trois modes suivants : dessin, effacement, ou déplacement.

La figure 4.2 contient deux photos d'une scène typique d'utilisation de FingerPaint. Sur la figure 4.2a, on voit l'utilisateur se servir de son doigt pour écrire à l'encre virtuelle. Désirant faire de la place sur le tableau, il "pousse" ensuite son dessin vers le haut (figure 4.2b). Cette manipulation,

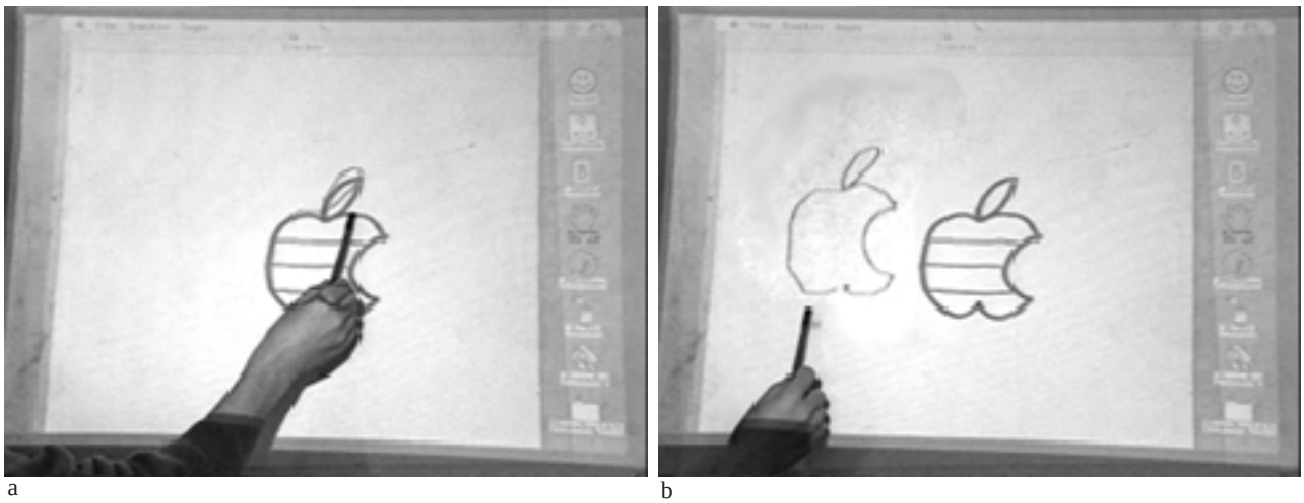
impossible à réaliser si le dessin avait été dessiné à l'encre réelle, devient très vite naturelle pour les utilisateurs de FingerPaint.



a b

Figure 4.2 : Dessin au doigt (a) et déplacement du dessin (b).

La manipulation représentée sur la figure 4.3 est une forme de décalcomanie : une pomme a été dessinée sur transparent. Celui-ci est déposé sur le datashow afin qu'une image de la pomme soit projetée sur le tableau. Un stylo est utilisé comme outil de pointage pour passer sur le contour de la pomme et en réaliser une copie à l'encre virtuelle (figure 4.3b). Finalement, le mode déplacement est utilisé pour une appréciation du résultat (figure 4.3b).



a b

Figure 4.3 : Décalque d'un motif projeté (a) et déplacement du dessin obtenu (b).

Les fonctionnalités à ajouter à FingerPaint sont nombreuses, il n'empêche que cette application a d'ores et déjà remplie son rôle de démonstration de certaines possibilités du bureau numérique. Elle

a également permit de tester et de mettre en application les techniques de suivi expérimentées au cours de ce projet, dont la présentation commence au chapitre suivant.

Chapitre 5

SUIVI PAR CORRÉLATION

Le suivi par corrélation est la solution la plus simple au problème de la recherche d'un motif dans une image. Schématiquement, on effectue une "mesure de similarité" d'un morceau de l'image et du motif dans toutes les positions qu'il peut occuper. La mesure de similarité est calculée en comparant chaque pixel du motif à son correspondant dans l'image.

Dans ce chapitre, le principe général du suivi par corrélation est tout d'abord présenté. Puis, nous proposons une étude de l'application de cette technique au bureau numérique. Enfin, nous rapportons les résultats de nos expérimentations sur le sujet.

5.1. Principe

Nous présentons dans cette section le principe général du fonctionnement du suivi par corrélation, puis nous entrons dans le détail d'une de ses composantes, la localisation, en nous intéressant à différentes mesures de similarité.

5.1.1. Fonctionnement général

Le principe général du fonctionnement du suivi par corrélation présenté ici est tiré de [Aschwanden88]. La figure 5.1 l'illustre par un schéma.

À l'initialisation du suivi, le motif représentant la cible est mémorisé comme motif de référence. Sa position initiale est passée à l'algorithme de prédiction qui calcule la zone de recherche la plus réduite possible en fonction des paramètres de la cible (position courante, vitesse courante, vitesse maximale, accélération maximale). Cette zone de recherche, qui dépend des principes généraux des techniques de suivi, est étudiée à la section 3.3.

Le processus passe alors à la localisation du motif au sein de la zone de recherche : la mesure de similarité est calculée pour chacune des positions que peut prendre le motif dans cette zone. La position ayant la meilleure valeur de similarité est choisie comme nouvelle position de la cible. Il existe une multitude de formules calculant des mesures de similarité. Nous étudions les principales au paragraphe suivant.

Une fois déterminée la nouvelle position de la cible, il existe deux façons de revenir au début du cycle du processus :

- soit on passe directement à l'image suivante,
- soit on mémorise une nouvelle image de la cible comme motif de référence. Le but de cette mise à jour est de prendre en compte les modifications d'aspect de la cible dues par exemple à ses changements d'orientation ou aux variations d'intensité lumineuse.

Une discussion sur les avantages et inconvénients des deux options, basée sur nos expérimentations, est donnée au paragraphe 5.3.2.

Finalement, le processus entame un nouveau cycle en invoquant de nouveau l'algorithme de prédiction avec les mises à jour des position et vitesse de la cible.

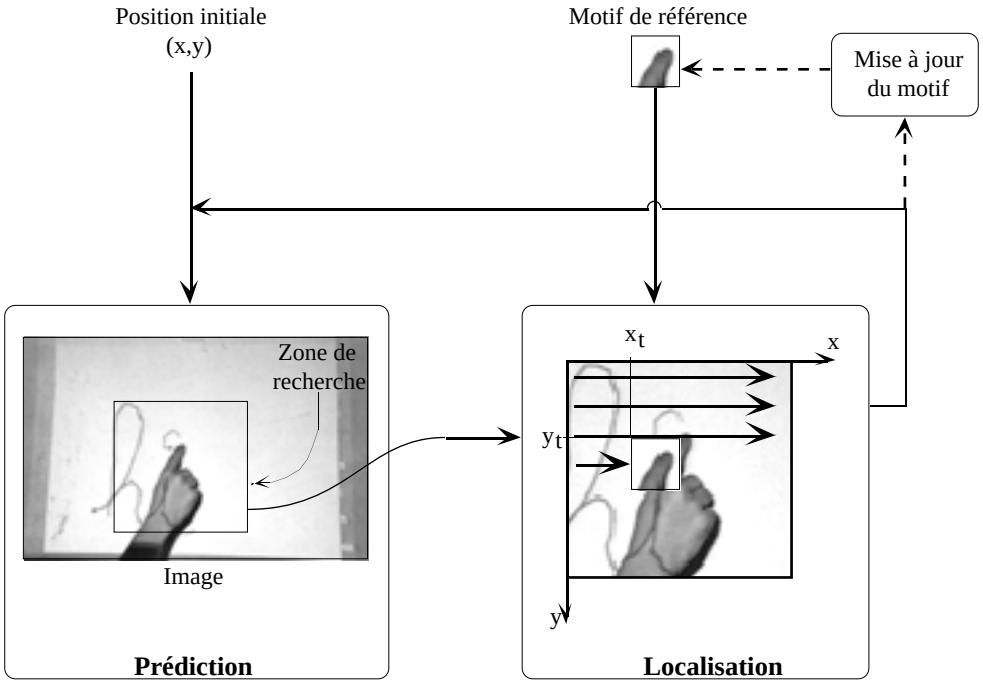


Figure 5.1 : Schéma de fonctionnement général du suivi par corrélation.

On peut deviner que le point clef de cette technique est la fonction qui donne la valeur de similarité entre le motif et un extrait de l'image. C'est pourquoi nous y consacrons le paragraphe suivant.

5.1.2. Mesures de similarité

Le problème peut s'énoncer simplement : il s'agit de comparer deux images de même taille.

Dans le cas de deux nombres, la comparaison est immédiate : on utilise leur écart (valeur absolue de la différence). Dans le cas de deux images, il n'existe pas *une* solution unique pour cette comparaison. C'est pourquoi on trouve une grande variété de suivis par corrélation, chacun appliquant sa mesure de similarité. Nous présentons ici les principales en nous basant sur celles appliquées par [Moravec80] et [Anadan87], ainsi que sur l'étude comparative réalisée par [Aschwanden92]. Dans la suite du paragraphe, les mesures de similarité sont calculées pour un motif M de taille $(m \times n)$ comparé à la zone de même taille dans l'image I à la position (i, j).

La similarité de deux images peut être vue comme l'extension de celle de deux nombres : elle est alors choisie comme étant la somme des écarts des valeurs de pixel. Plus cette somme est proche de zéro et plus les images sont similaires. On la note SAD (Sum of Absolute Differences) et elle se calcule selon la formule :

$$SAD(i,j) = \sum_{u=0}^{m-1}\sum_{v=0}^{n-1} \left| M(u,v) - I(i+u, j+v) \right|$$

En pratique, il est plus intéressant de sommer les carrés des différences de pixels car cela fait intervenir une multiplication moins coûteuse en temps que le calcul d'une valeur absolue. On parle alors de SSD (Sum of Squared-Differences) :

$$SSD(i,j) = \sum_{u=0}^{m-1}\sum_{v=0}^{n-1} \left(M(u,v) - I(i+u, j+v) \right)^2$$

Il est facile de constater que ces deux formules (SAD et SSD) atteignent leur minimum au même point (i, j). [Moravec80] remarque que si la caméra effectue une mesure photométrique exacte, et si les conditions lumineuses restent inchangées entre les différentes images, ces formules sont idéales : les représentations de la cible sur le motif et dans l'image sont parfaitement identiques et les deux fonctions s'annulent lorsque le motif coïncide avec son double dans l'image.

Malheureusement, la réponse de la caméra varie en fonction de la position de la cible dans son champ. Ceci est du aux distorsions de son optique et à certains phénomènes de la numérisation. De même, les conditions lumineuses englobant la cible varient en fonction du temps et de sa position. D'autres mesures de similarité ont été conçues afin de limiter l'effet de ces distorsions.

On se reportera à [Aschwanden92] pour une étude comparative expérimentale de 19 mesures de similarité. Cette étude rapporte le taux de réussite (motif localisé au pixel près) en fonction de la taille du motif et pour 5 différents types de distorsions. Il en ressort que les résultats des différentes mesures de similarité sont généralement proches. Cependant la corrélation normalisée (Normalized Cross-Correlation), qui a donné son nom à cette famille d'algorithmes de suivi, semble présenter une fiabilité supérieure aux autres, particulièrement sur les variations lumineuses. C'est pourquoi elle est utilisée dans nos expériences. Sa forme est la suivante :

$$NCC(i,j) = \frac{\sum_{u=0}^{m-1}\sum_{v=0}^{n-1} M(u,v).I(i+u,j+v)}{\sqrt{\sum_{u=0}^{m-1}\sum_{v=0}^{n-1} M^2(u,v).\sum_{u=0}^{m-1}\sum_{v=0}^{n-1} I^2(i+u,j+v)}}$$

Le principe général du suivi par corrélation étant défini, nous allons l'étudier dans le contexte précis de son utilisation pour le bureau numérique.

5.2. Application au bureau numérique

Nous présentons dans cette section les réponses que le suivi par corrélation peut donner aux problèmes soulevés dans le chapitre 3. Ces réponses sont basées sur l'analyse du principe de cette technique, cependant elles ont été corroborées par une série d'expérimentations dont nous donnons les principaux résultats à la section suivante.

5.2.1. Initialisation

Nous avons vu au paragraphe 5.1.1. que le suivi par corrélation nécessite un motif de référence à l'initialisation. Il semble nécessaire d'avoir recours à l'utilisateur afin que celui-ci positionne explicitement son doigt dans une zone définissant ce motif. La détection semi-automatique de l'initialisation envisagée au paragraphe 3.4.1. s'applique parfaitement à cette technique de suivi : le cadre de détection du mouvement servirait également à définir la taille du motif du suivi.

De même, la corrélation étant un algorithme rapide, il est possible d'envisager que l'ordinateur scrute en permanence la totalité de la surface du bureau en y cherchant un ou plusieurs motifs prédéfinis (extrémité du doigt, du stylo ou de la gomme). Le suivi est initialisé lorsqu'une position de l'image donne une valeur de similarité qui dépasse un certain seuil pour l'un des motifs. On se reportera au paragraphe 3.4.3. pour les détails de cette solution.

5.2.2. Adaptation à différentes formes

Le suivi par corrélation n'impose aucun modèle de la cible puisqu'il se base uniquement sur un motif : une image de taille réduite dont le contenu peut être quelconque. Grâce à cela il est possible de "l'accrocher" à strictement n'importe quel objet.

Dans le cas du bureau numérique, on peut imaginer que le système dispose en mémoire d'un ensemble de motifs "courants" qu'il est capable de localiser immédiatement comme présenté au paragraphe précédent. En plus de cela, le système est capable d'acquérir de nouveaux motifs en vue de l'utilisation ponctuelle d'un nouvel outil de pointage.

5.2.3. Traitement des échecs

Le suivi par corrélation est capable de détecter précocement la perte de la cible : il suffit pour cela d'observer la valeur de similarité dans chaque nouvelle image. Une chute de cette valeur est symptomatique de l'échec du suivi. Il peut y avoir deux raisons à cela :

- soit un changement d'éclairage ou d'orientation de la cible entraîne une modification de son aspect par rapport au motif mémorisé. Ces changements étant en général progressifs, il est possible au suivi de réagir rapidement en mémorisant un nouveau motif prenant en compte le changement d'aspect parce que le capturant sur l'image courante,
- ou bien un déplacement trop rapide de la cible l'a fait sortir de la zone de recherche et dans ce cas la mesure de similarité chute brutalement d'une image à l'autre. La solution ici est d'étendre la zone de recherche. Celle-ci croît tant que le motif n'est pas retrouvé jusqu'à recouvrir la totalité de l'image.

5.2.4. Robustesse

Nous allons voir que la robustesse est le point faible du suivi par corrélation. C'est une conséquence de l'inexistence d'un modèle de la cible dans cette technique. Les erreurs se manifestent en premier lieu lorsque la cible se déplace sur fond bruité.

La taille du motif est un facteur déterminant pour la robustesse du suivi face à un fond bruité : plus celle-ci est réduite, plus il est probable qu'une partie du fond apparaissant dans la zone de recherche ait une valeur de similarité au motif supérieure à celle de la cible. Une telle éventualité est fatale au suivi qui se fixe sur cette zone sans avoir détecté l'échec.

Il semble que la solution soit de prendre un motif de grande taille. Cela n'est possible que dans certaines limites : premièrement la charge de calcul est directement liée à la taille du motif. Ensuite, celle-ci est limitée par la forme de l'objet à suivre.

La figure 5.2 illustre ce problème dans le cas du suivi d'un doigt : la taille du motif ne peut dépasser la largeur du doigt, sans quoi une partie du fond de l'image serait mémorisée dans le motif de référence.

De par son principe, le suivi par corrélation est également fragile face aux variations d'intensité lumineuse : la mesure de similarité s'effectuant sur la valeur d'intensité des pixels, le même objet illuminé différemment fournit deux images dissemblables.

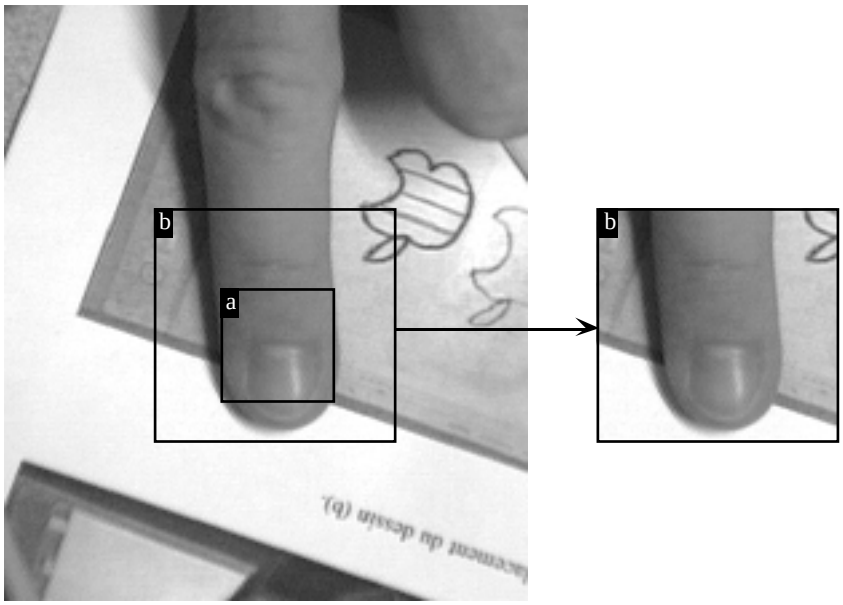


Figure 5.2 : Limite de la taille du motif pour le suivi du doigt :
tout cadre de taille supérieur à (a) contient une partie du fond,
c'est le cas du cadre (b).

De la même manière, deux images d'un même objet présenté sous deux orientations différentes sont dissemblables au sens de la corrélation : les mesures de similarité comparent les couples de pixels qui ont la même position dans le motif et dans l'image (figure 5.3a), alors qu'il faudrait prendre en compte les changements d'orientation pour définir ces couples (figure 5.3b).

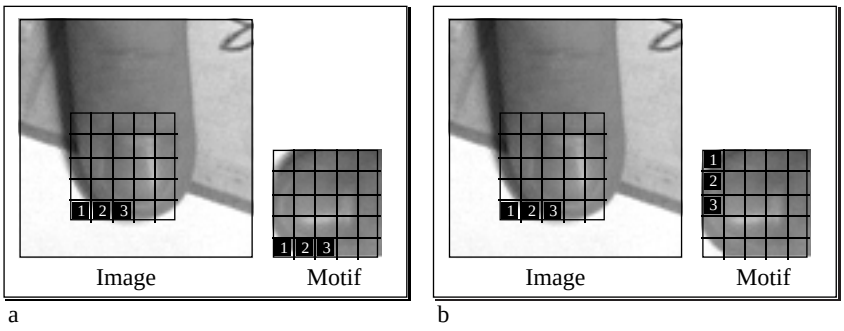


Figure 5.3 : Changement d'orientation du motif.
(a) : 3 couples de pixels comparé par le suivi par corrélation,
(b) : 3 couples de pixels qui correspondent réellement.

En conclusion, si le principe du suivi par corrélation semble adapté à plusieurs aspects du bureau numérique (initialisation, adaptation à différente formes, détection et correction d'erreur) son manque de robustesse est prévisible et sera vérifié dans la section suivante qui présente nos expérimentations à ce sujet.

5.3. Expérimentations

Notre algorithme de suivi par corrélation est une mise en oeuvre du principe décrit au paragraphe 5.1.1. Il est paramétrable afin de pouvoir utiliser de manière indifférente les formules de mesure de similarité SSD et NCC définies au paragraphe 5.1.2. Le point le plus intéressant de sa réalisation est le calcul de la taille optimale de la zone de recherche, qui permet une vitesse de déplacement maximale de la cible.

5.3.1. Taille optimale de zone de recherche

Notre raisonnement est le suivant : soit t la taille de la zone de recherche et e celle du motif. Entre deux images, la cible ne peut se déplacer de plus de $(t-e)/2$ pixels sous peine d'être perdue par le suivi (figure 5.4).

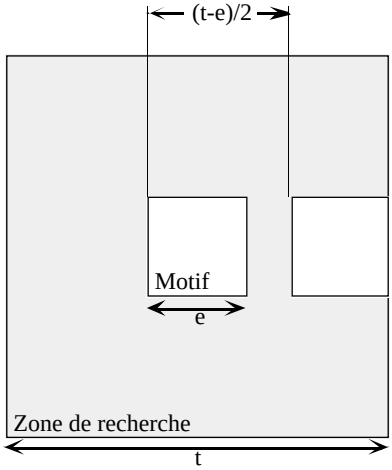


Figure 5.4 : Déplacement maximum de la cible entre deux images.

La plus grande vitesse de déplacement de la cible que peut supporter le suivi est :

$$v_m = \frac{(t-e)}{2}.F(t)$$

où $F(t)$ est la fréquence de fonctionnement de l'algorithme de suivi qui ne dépend que de la taille de la zone d'exploration (t) pour une taille de motif (e) donnée. $F(t)$ est inversement proportionnelle au nombre de corrélations à effectuer entre chaque image, donc à la taille de la zone de recherche, donc au carré de t. Soit

$$F(t) = \frac{k}{t^2}$$

où k est une constante positive, alors

$$v_m = \frac{(t-e)}{2}.\frac{k}{t^2} = \frac{k}{2}\left(\frac{1}{t} - \frac{e}{t^2}\right)$$

La forme de la courbe de la vitesse maximale de la cible en fonction de la taille de la zone de recherche est donnée sur la figure 5.5.

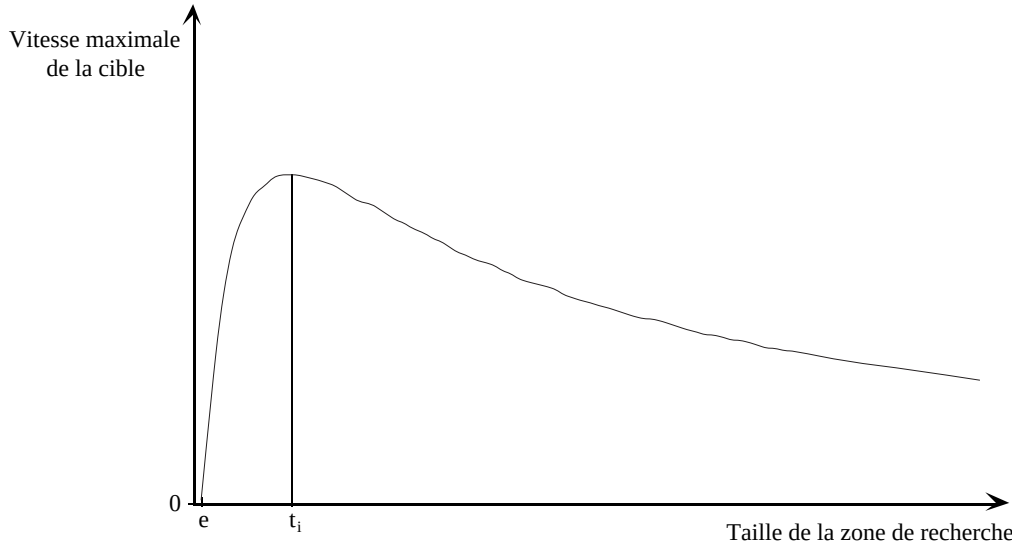


Figure 5.5 : Courbe de la vitesse maximale de la cible en fonction de la taille de la zone de recherche.

Lorsque $t = e$, v_m est nulle puisque la recherche ne se fait que sur la position courante de la cible. Puis, v_m croît rapidement lorsque la taille de la zone de recherche grandit : le suivi est capable de trouver la cible de plus en plus loin de sa position courante. Cependant, il arrive un moment où le surcroît de temps, du à une recherche dans une zone plus grande, compense le gain en vitesse précédent. Cette limite définit la taille idéale t_i de la zone de recherche car ensuite v_m décroît.

Nous avons déterminé cette limite expérimentalement en mesurant les fréquences de fonctionnement du suivi pour une taille de motif de 8 pixels et des tailles de zone de recherche variant de 10 à 46 pixels. Les mesures enregistrées sont récapitulées dans le tableau de la figure 5.6, la courbe correspondante est représentée sur la figure 5.7. Ces mesures nous permettent de connaître la taille de la zone de recherche optimale (t_i) pour notre système : 26 pixels.

5.3.2. Robustesse

La première version de notre algorithme de suivi mémorise une fois pour toute le motif à l'initialisation. On observe alors que les échecs prévus au paragraphe 5.2.4. sont fréquents : le suivi perd le doigt lorsqu'il pénètre dans une zone d'ombre par exemple ou lorsqu'il effectue une rotation dans le plan du bureau. Nous changeons notre algorithme afin qu'il rafraîchisse le motif à chaque image (cette option est présentée au paragraphe 5.1.1.).

t (pixels)	F (Hz)	v_m (pixels/s)
10	32.80	32.80
12	25.60	51.20
14	24.90	74.70
16	24.95	99.80
18	24.80	124.00
20	23.40	140.40
22	20.00	140.00
24	16.70	133.60
26	**16.50**	**148.50**
28	14.00	140.00
30	12.46	137.06
32	11.40	136.80
34	10.00	130.00
36	9.60	124.80
38	9.51	123.63
40	8.84	114.92
42	8.45	109.85
44	8.62	112.06
46	7.70	100.10

Figure 5.6 : Fréquence de fonctionnement (F) et vitesse maximale de la cible (v_m) en fonction de la taille de la zone de recherche (t).

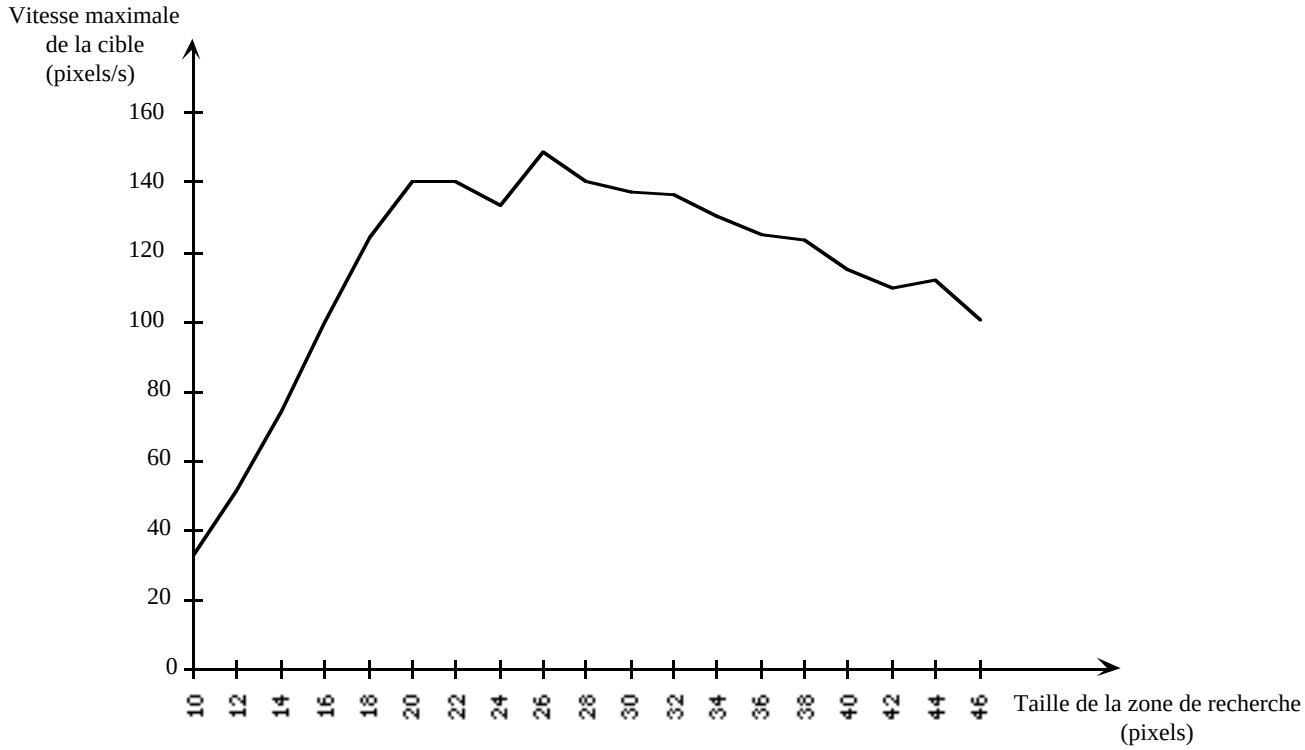


Figure 5.7 : Courbe mesurée des vitesses maximales de la cible en fonction de la taille de la zone de recherche.

Nous espérons ainsi autoriser les changements d'aspect progressifs de la cible et c'est effectivement le cas, mais un nouveau problème apparaît : lors d'une rotation du doigt par exemple, sa localisation est approximative car son aspect change entre chaque image. Il en résulte que le suivi se détache progressivement de sa cible. Ce cas de figure est représenté sur la figure 5.8. Il est la conséquence du fait que le suivi par corrélation n'utilise pas de modèle de la cible. Le plus petit décalage du motif ne peut être détecté ni corrigé.

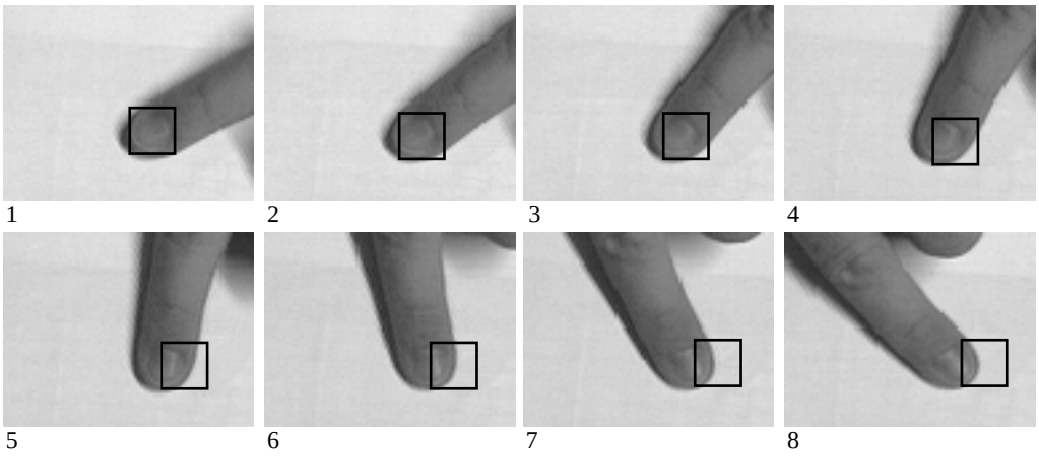


Figure 5.8 : Décalage du suivi lors du changement d'aspect du motif.

Cette situation devient paradoxale lorsque le motif s'est complètement détaché du doigt : c'est le fond de l'image qui est alors "suivi" et le fait d'approcher le doigt fait fuir le cadre qui représente le motif.

En fin de compte, cette expérimentation nous apprend que le suivi par corrélation est rapide aussi bien à mettre en oeuvre qu'à l'exécution. Il apparaît également que cette simplicité est la cause de son manque de robustesse. C'est pourquoi nous nous sommes intéressé à une technique de suivi plus perfectionnée car mettant en oeuvre un modèle complexe de la cible : le suivi par contour actif.

Chapitre 6

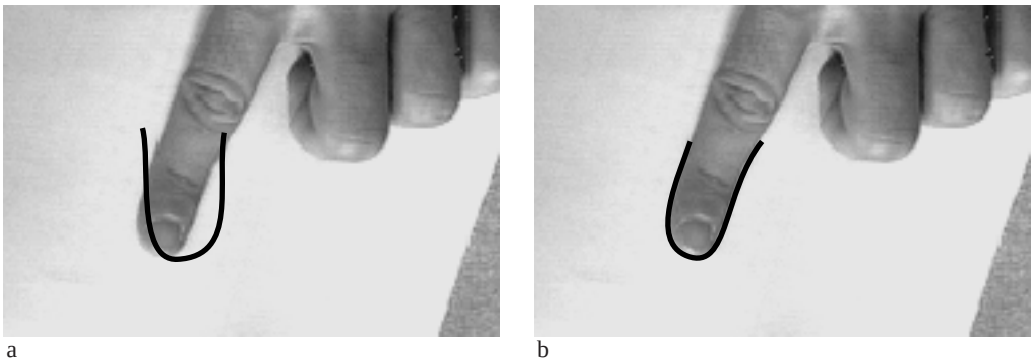
SUIVI PAR CONTOUR ACTIF (SNAKE)

La première apparition des contours actifs date de 1987, lorsque Kass, Witkin et Terzopoulos ([Kass87], [Terzopoulos87]) présentent leur Snakes comme un outil permettant de résoudre plusieurs sujets majeurs de la vision par ordinateur, à savoir la détection de contours et de segments, la mise en correspondance en stéréovision, et surtout le suivi d'objet. La compacité et l'efficacité du concept lui ont assuré un succès rapide et l'on trouve à l'heure actuelle une multitude de publications sur des variations de sa forme originelle ([Cohen90], [Williams90], [Terzopoulos92], [Curwen92], [Ueda92]).

La première section de ce chapitre se base sur la formulation initiale du modèle des snakes ([Kass87]) pour en présenter le concept. Le développement sur leur modèle dynamique s'appuie sur la description qui en est faite dans [Terzopoulos92]. Les problèmes liés à la simulation numérique sont traités dans la seconde section du chapitre. En particulier, nous y exposons les travaux de [Williams90] sur la simulation dynamique du comportement du snake. Ensuite l'intérêt se porte sur l'étude de la mise en oeuvre d'un snake comme outil de suivi pour le bureau numérique. Finalement, nos expérimentations sur le sujet concluent le chapitre.

6.1. Modèle mathématique

Comme le montre la figure 6.1, le principe consiste à placer dans l'image un snake qui "se colle" au contour de la cible. Une énergie lui est associée, fonction de sa déformation et de sa position dans l'image. Le snake cherche en permanence à minimiser cette énergie. Celle-ci est modélisée de telle manière que le comportement résultant est une suite de déplacements et de déformations jusqu'à ce que le snake se stabilise sur un minimum local de son énergie qui correspond alors à la forme de l'objet suivi.



a b

Figure 6.1 : Initialisation (a) et stabilisation (b) d'un snake autour d'un doigt.

6.1.1. Énergie globale

Dans sa forme continue, le snake est modélisé par une courbe paramétrée $v(s) = (x(s), y(s))$ sur le domaine unité ($s \in [0,1]$). Son énergie est fonction de sa configuration géométrique (énergie interne) et de l'image (énergie externe). Elle s'écrit sous la forme :

$$E^*_{Globale}(v) = E^*_{Interne}(v) + E^*_{Externe}(v) \quad (1)$$

Chaque point de la courbe du snake contribue au calcul de cette énergie, ce qui se traduit par :

$$E^*_{Globale}(v) = \int_0^1 E_{Globale}(v,s).ds \quad (2)$$

soit

$$E^*_{Globale}(v) = \int_0^1 \left(E_{Interne}(v,s) + E_{Externe}(v,s)\right)ds \quad (3)$$

Il s'agit maintenant de construire des énergies internes et externes qui procurent au snake le comportement désiré.

6.1.2. Énergie interne

L'énergie interne est conçue de façon à autoriser des déformations longitudinales (élongations ou contractions) et des déformations de courbure. On la modélise par :

$$E_{Interne}(v,s) = \frac{1}{2}\left(\alpha(s).\left|\frac{\partial v}{\partial s}(s)\right|^2 + \beta(s).\left|\frac{\partial^2 v}{\partial s^2}(s)\right|^2\right) \quad (4)$$

Le terme $\frac{\partial v(s)}{\partial s}$ donne la valeur de l'élongation au point s du snake. Il est contrôlé par la fonction $\alpha(s)$ qui règle la tension en fonction de la position sur le snake. Le terme $\frac{\partial^2 v(s)}{\partial s^2}$ quantifie la courbure du snake dont la rigidité est également paramétrable (par $\beta(s)$). En particulier la figure 6.2 montre que l'affectation $\alpha(s_1) = \beta(s_1) = 0$ autorise une discontinuité du snake au point s_1 (ce qui n'a pas grand intérêt), et que spécifier $\beta(s_2) = 0$ uniquement lui permet de développer un angle au point s_2.

Figure 6.2 : Paramétrisation des déformations du snake.

L'énergie interne permet donc de définir la tension et la rigidité du snake. Par contre elle ne lui impose pas de forme particulière : si on laisse un snake se stabiliser uniquement sur le minimum de son énergie interne, c'est-à dire lorsque $\frac{\partial v(s)}{\partial s} = \frac{\partial^2 v(s)}{\partial s^2} = 0$, alors il se détend en une ligne droite dont les "points" sont régulièrement espacés. Il faut noter en particulier que le snake n'a pas de longueur privilégiée. Ses déformations et déplacements résultant de son interaction avec l'image sont modélisés par son énergie externe.

6.1.3. Énergie externe

De manière générale, si $v(s) = (x(s), y(s))$, l'énergie externe est égale à :

$$E^*_{Externe}(v) = \int_0^1 E_{Image}(x(s), y(s)).ds \tag{5}$$

où E_{Image} définit la force externe en chaque point de l'image. Les minimums de ce champ de force sont des attracteurs du snake puisque celui-ci cherche à minimiser son énergie. La modularité du concept des snakes est ici mise en évidence puisque le comportement du snake vis-à-vis de l'image dépend uniquement de E_{Image}. Plusieurs formes différentes de E_{Image} sont proposées dans [Kass87]. Soit $I(x,y)$ la fonction représentant l'intensité de l'image. La proposition la plus simple de la fonction d'énergie externe s'écrit :

$$E_{Ligne}(x,y) = I(x,y) \tag{6}$$

Le comportement du snake dérivant de cette énergie est une attirance vers les points sombres de l'image car leur signal est de valeur faible (en pratique, ceux-ci sont souvent regroupés en ligne d'où le nom de E_{Ligne}). L'attraction aux points lumineux est réalisée par une simple inversion du signe :

$$E_{Ligne}(x,y) = -I(x,y) \tag{7}$$

La deuxième forme de l'énergie externe proposée par [Kass87] a pour objectif de créer une attraction du snake par les contours de l'image. Elle est réalisée en utilisant le gradient de l'image $(\nabla I(x,y))$ qui représente les variations d'intensité du signal. Un exemple d'image et son gradient est représenté sur les figure 6.3a et 6.3b. (L'image du gradient a été inversée afin que les pixels de fort gradient soient de couleur noire.) L'énergie externe s'écrit alors :

$$E_{Contour}(x,y) = -|\nabla I(x,y)|^2 \tag{8}$$

L'élévation au carré élimine le problème du signe dû au sens de variation du gradient et fournit une valeur toujours positive. De plus, elle amplifie les forts gradients. L'inversion du signe rend cette fonction d'autant plus petite (générant l'attirance du snake) que la norme du gradient est grande.

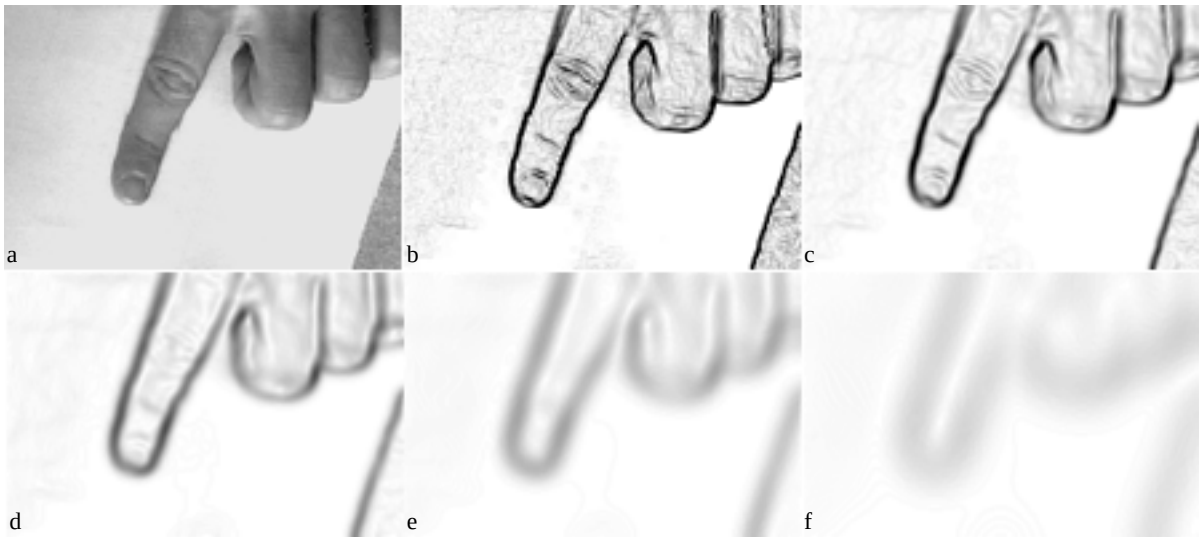


Figure 6.3 : Image d'un doigt (a), son gradient (b),
gradient calculé sur l'image lissée par un filtre Gaussien de rayon 1 (c), 2 (d), 4 (e), et 8 (f) pixels.

Si le snake est assez proche d'une ligne de forte valeur de E_{Image}, alors il s'y stabilise. Dans certains cas, et en particulier pour E_{Ligne} et $E_{Contour}$, la distance d'attraction peut être augmentée en lissant au préalable l'image par un filtre Gaussien. Un exemple en est donné sur la figure 6.3 où l'on voit les gradients d'une image non lissée, puis lissée par un filtre Gaussien de rayon 1, 2, 4 et 8 pixels. Une première stabilisation du snake est effectuée sur l'image 6.3f ce qui donne une approximation de la position du doigt. On peut alors progressivement améliorer la résolution en stabilisant le snake sur les images 6.3e, d, c jusqu'à atteindre la précision maximale sur l'image 6.3b.

6.1.4. Aspect dynamique

Le modèle du snake tel qu'il est décrit jusqu'ici peut être utilisé aussi bien pour localiser précisément un contour dans une image fixe que pour suivre un mouvement non rigide du contour. Pour trouver la position de la cible du suivi dans une nouvelle image, le snake est initialisé sur sa position dans l'image précédente et le processus de minimisation est relancé. L'idée d'utiliser le snake comme technique de suivi était déjà présente dans [Kass87] : le suivi du mouvement des lèvres d'une personne en train de parler y est présenté (voir figure 6.4 extraite de [Kass87]).

Cependant, si entre les deux images le déplacement de la cible est trop grand, le snake la perd et se stabilise sur un minimum local de l'image la plus proche. C'est pourquoi le modèle de base des snakes a été adapté à l'objectif particulier du suivi d'objet.

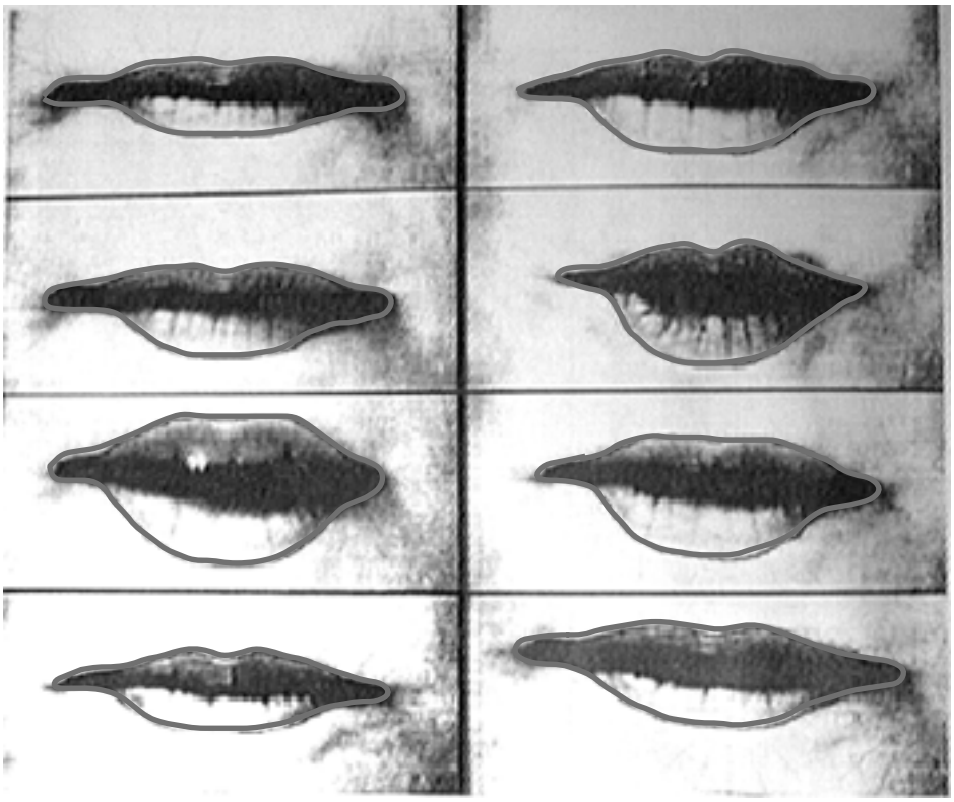


Figure 6.4 : Utilisation d'un snake pour suivre le mouvement des lèvres [Kass87].

L'idée, présentée notamment dans [Terzopoulos92], est d'associer une masse au snake afin que son énergie interne intègre une énergie dynamique minimale lorsque le snake se déplace en ligne droite à vitesse constante. Mathématiquement, le temps est introduit comme paramètre de la configuration géométrique du snake qui est alors représenté par $v(s,t)$. La masse le long du snake est modélisée par la fonction $\mu(s)$. L'énergie dynamique peut être définie par :

$$E_{Dynamique}(v,t,s) = \frac{1}{2}\mu(s).\left|\frac{\partial^2 v}{\partial t^2}(t,s)\right|^2 \tag{9}$$

L'énergie dynamique est bien minimale (égale à zéro) lorsque la variation de vitesse (l'accélération) $\frac{\partial^2 v}{\partial t^2}(t,s)$ est nulle. Prendre en compte ce terme permet de suivre des contours plus rapidement à condition que ceux-ci ne soient pas sujets à des accélérations trop brusques, ou que la fréquence d'échantillonnage de l'image soit assez élevée. L'introduction de ce terme a cependant l'inconvénient de faire osciller le snake autour de sa position de stabilisation. Une force simulant un frottement visqueux est ajoutée au modèle afin de dissiper l'énergie cinétique lorsque le snake est proche de sa position d'équilibre :

$$E_{Frottement}(v,t,s) = \frac{1}{2}\gamma(s).\left|\frac{\partial v}{\partial t}(t,s)\right|^2 \tag{10}$$

où $\gamma(s)$ représente le facteur de viscosité le long du snake. Son comportement est toujours simulé par minimisation de son énergie globale qui prend la forme suivante :

$$E^*_{Globale}(v,t) = \int_0^1 \left(E_{Dynamique}(v,s,t) + E_{Frottement}(v,s,t) + E_{Interne}(v,s,t) + E_{Externe}(v,s,t) \right) ds \quad (11)$$

La prise en compte de la masse du snake est à considérer en parallèle du calcul de la zone de recherche du suivi présenté à la section 3.3. Tous deux ont pour objectif d'utiliser la vitesse de déplacement de la cible afin de prévoir sa nouvelle position. Ce qui différencie le cas du snake est que la vitesse fait partie intégrante du modèle par l'intermédiaire de sa masse. Ainsi, la prédiction de la nouvelle position est intrinsèque car elle est incluse dans la minimisation globale de l'énergie : l'énergie dynamique est minimale lorsque le snake est exactement sur la position prédite.

Après avoir présenté le modèle mathématique du snake, nous poursuivons au paragraphe suivant par sa mise en œuvre informatique.

6.2. Mise en œuvre

L'adaptation du modèle mathématique à la simulation informatique passe en premier lieu par une discrétisation du modèle continu. Ensuite, il est nécessaire de trouver un algorithme qui simule le comportement du snake par minimisation de son énergie. Étant donné un snake de longueur l, il existe, dans une image fixe, une et une seule configuration géométrique du snake qui correspond à sa plus faible énergie. La minimisation de l'énergie du snake ne cherche pas à trouver exactement cette solution unique, d'abord parce que c'est un problème trop complexe, ensuite parce que l'intérêt porte sur un comportement dynamique du snake lié aux minimums locaux du système.

En effet, que ce soit dans le cas d'une recherche de contours sur image fixe ou dans celui du suivi dans une séquence animée, la cible du snake a peu de chance de correspondre à l'unique solution minimale. La recherche est donc ramenée à une zone réduite, autour de la position précédente du snake, et le recours à un algorithme de plus haut niveau (ou même à l'utilisateur) est nécessaire pour initialiser la position du snake "près" du centre d'intérêt (contour à déterminer, ou cible à suivre). La minimisation tente alors de réduire l'énergie globale du contour par déplacements successifs de ses points.

Dans ce qui suit, nous définissons un modèle discret du snake, puis nous présentons deux techniques différentes de minimisation.

6.2.1. Discrétisation

Le principe de la discrétisation que nous présentons ici est tiré de [Kass87], mais il a été repris par l'ensemble des publications sur le sujet. Les seules variations se situent au niveau de la représentation de la courbure du snake. Rappelons que le snake est modélisé par une courbe paramétrée $v(s) = (x(s), y(s))$ où $s \in [0,1]$ est un paramètre continu. Le modèle discret du snake s'obtient naturellement en construisant un vecteur u regroupant n nœuds du snake. Ces nœuds sont les points $v(ih) = (x(ih), y(ih))$ avec $i = 0..n-1$ et $h = 1/(n-1)$. Dans la suite de ce paragraphe,

$u_i = (x_i, y_i)$ représente la position du i^{ème} nœud du snake. L'énergie interne d'un nœud du snake, représentée dans sa version continue par l'équation (4) se présente en discret sous la forme :

$$E_{Interne}(i) = \frac{1}{2}\left(\alpha(i).\left|\frac{u_i - u_{i-1}}{h}\right|^2 + \beta(i).\left|\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}\right|^2 \right) \tag{12}$$

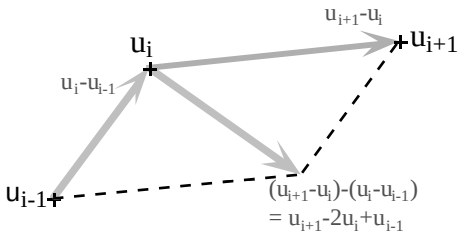


Figure 6.5 : Vecteurs distance (u_i-u_{i-1} et u_{i+1}-u_i) et variation (u_{i-1}-2u_i+u_{i+1})

$u_i - u_{i-1}$ représente le vecteur entre les nœuds $i-1$ et i. On supprime un calcul de racine dans l'évaluation de l'énergie d'élongation en faisant intervenir le carré de la norme de ce vecteur qui est calculé ainsi :

$$\left|u_i - u_{i-1}\right|^2 = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \tag{13}$$

La courbure du snake est représentée par le vecteur $u_{i-1} - 2u_i + u_{i+1}$: c'est la différence (variation) entre les vecteurs $\overrightarrow{u_{i-1}u_i}$ et $\overrightarrow{u_i u_{i+1}}$ (voir figure 6.5). Même si ce n'est pas la courbure réelle du snake, elle a l'avantage de fournir une formule économe en temps de calcul (elle ne fait pas intervenir de fonctions trigonométriques) :

$$\left|u_{i-1} - 2u_i + u_{i+1}\right|^2 = (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2 \tag{14}$$

En ce qui concerne l'énergie externe, la discrétisation est immédiate puisque le potentiel est calculé à partir de l'image qui est dans la plupart des cas disponible sous forme discrète ($I(i, j)$) et non pas sous forme continue ($I(x, y)$). On utilise par exemple :

$$E_{Externe}(i) = E_{Contour}(x_i, y_i) \tag{15}$$

L'énergie globale du système peut finalement être présentée sous la forme :

$$E^*_{Globale} = \sum_{i=0}^{n-1} E_{Interne}(i) + E_{Externe}(i) \tag{16}$$

Le même principe est utilisé pour la discrétisation des énergies dynamiques (9) et (10). On se reportera à [Terzopoulos92] pour le détail des formules. La représentation discrète du snake et de son modèle d'énergie étant définie, on s'attache maintenant à touver une méthode pour minimiser cette énergie.

6.2.2. Minimisation de l'énergie par calcul variationnel

La première solution pour minimiser l'énergie globale du snake, proposée dans l'article originel ([Kass87]), se fonde sur le calcul variationnel : la fonction d'énergie du snake atteint un minimum local lorsque sa dérivée est nulle. Supposons que les paramètres d'élongation et de courbure soient constants ($\alpha(s)=\alpha$, $\beta(s)=\beta$). L'application de ce principe à l'équation (1) produit les deux équations suivantes :

$$\alpha.\frac{\partial^2 x}{\partial s^2}+\beta.\frac{\partial^4 x}{\partial s^4}+\frac{\partial E_{Externe}}{\partial x}=0 \tag{17}$$

$$\alpha.\frac{\partial^2 y}{\partial s^2}+\beta.\frac{\partial^4 y}{\partial s^4}+\frac{\partial E_{Externe}}{\partial y}=0 \tag{18}$$

Ou bien, dans leur forme discrète :

$$\begin{aligned}&\alpha_i(u_i-u_{i-1})-\alpha_{i+1}(u_{i+1}-u_i)\\&+\beta_{i-1}(u_{i-2}-2u_{i-1}+u_i)-2\beta_i(u_{i-1}-2u_i+u_{i+1})+\beta_{i+1}(u_i-2u_{i+1}+u_{i+2})\\&+\left(f_x(i),f_y(i)\right)=0\end{aligned} \tag{19}$$

où $f_x(i)=\frac{\partial E_{Externe}}{\partial x}(x_i,y_i)$ et $f_y(i)=\frac{\partial E_{Externe}}{\partial y}(x_i,y_i)$ sont les versions discrètes des dérivées partielles de la force externe. Les deux équations associées dans la formule (19) peuvent s'écrire sous forme matricielle :

$$Ax+f_x(x,y)=0 \tag{20}$$

$$Ay+f_y(x,y)=0 \tag{21}$$

Ce système est résolu itérativement par la méthode d'Euler en affectant aux côtés droits des équations (20) et (21) le produit d'une taille de pas γ et l'opposée de la variation entre les itérations :

$$Ax_t+f_x(x_{t-1},y_{t-1})=-\gamma(x_t-x_{t-1}) \tag{22}$$

$$Ay_t+f_y(x_{t-1},y_{t-1})=-\gamma(y_t-y_{t-1}) \tag{23}$$

A l'équilibre, la variation entre deux itérations disparaît ($x_t\approx x_{t-1}$, et $y_t\approx y_{t-1}$) et une solution des équations (20) et (21) est obtenue. Les équations (22) et (23) peuvent être résolues par inversion de matrice :

$$x_t=(A+\gamma I)^{-1}(x_{t-1}-f_x(x_{t-1},y_{t-1})) \tag{24}$$

$$y_t=(A+\gamma I)^{-1}(y_{t-1}-f_y(x_{t-1},y_{t-1})) \tag{25}$$

La matrice $A+\gamma I$ est pentadiagonale (l'énergie d'un nœud dépend de sa position, et de celle des ses deux voisins précédents et de ses deux voisins suivants). Son inverse peut être calculée par décomposition triangulaire haute et basse avec une complexité de l'ordre de O(n) (n est le nombre de nœuds du snake).

La méthode que nous venons de présenter tente de minimiser en parallèle l'énergie de chacun des nœuds en fonction de leurs positions précédentes. [Williams90] propose une autre approche dont la principale différence tient au fait que l'énergie est minimisée séquentiellement, nœud après nœud.

6.2.3. Minimisation de l'énergie par algorithme dynamique

Dans cet algorithme, la minimisation se fait tour à tour sur chaque nœud du snake, indépendamment de celle des autres. Soit u_i le nœud considéré. Son énergie est évaluée de la même manière que par l'algorithme précédent, soit comme la somme des formules (12) et (15). Cette opération est effectuée pour toutes les positions du voisinage de u_i. [Williams90] utilise un voisinage de neuf points (les huit voisins et la position initiale). La position correspondant à l'énergie minimale est retenue comme nouvelle position de u_i, et on passe au traitement du nœud suivant. La minimisation globale se termine lorsqu'après un cycle de traitement sur tous les nœuds, le nombre de nœuds ayant bougé est inférieur à un certain seuil.

Ce fonctionnement est résumé par le pseudo-code suivant :

```
# n           = nombre de nœuds
# déplacés    = nombre de points déplacés au cours d'un cyle
# m           = nombre de voisins
# V[0..m-1]   = voisins

# Boucle de minimisation globale
Répéter
    déplacés = 0;

    # Cycle de minimisation (traitement de tous les nœuds
    #                        un par un)
    Pour (i = 0 à n)  # Le nœud 0 est traité en premier et dernier
        E_min = MAXIMUM
        V = les voisins de u_i

        # Boucle sur le voisinage
        Pour (j = 0 à n-1)
            E_j = E_Interne(V[j]) + E_Externe (V[j])

            Si (E_j < E_min)
                E_min = E_j
                j_min = j

        Si (V[j_min] différent de la position courante)
            Déplacer nœud courant vers V[j_min]
            déplacés = déplacés + 1

Jusqu'à (déplacés < seuil)
```

La minimisation globale du système est obtenue au fur et à mesure des minimisations en chacun des nœuds du snake : par exemple un nœud peut se déplacer pour atteindre un pic du gradient sans prendre en compte l'augmentation d'énergie que cela provoque pour son voisin du fait de l'élongation du snake (figure 6.6b). Au cycle suivant, le voisin se rapproche afin de ramener son énergie interne au niveau minimal (figure 6.6c). Le bilan des deux cycles est une diminution de l'énergie globale du snake car un nœud a minimisé son énergie externe.

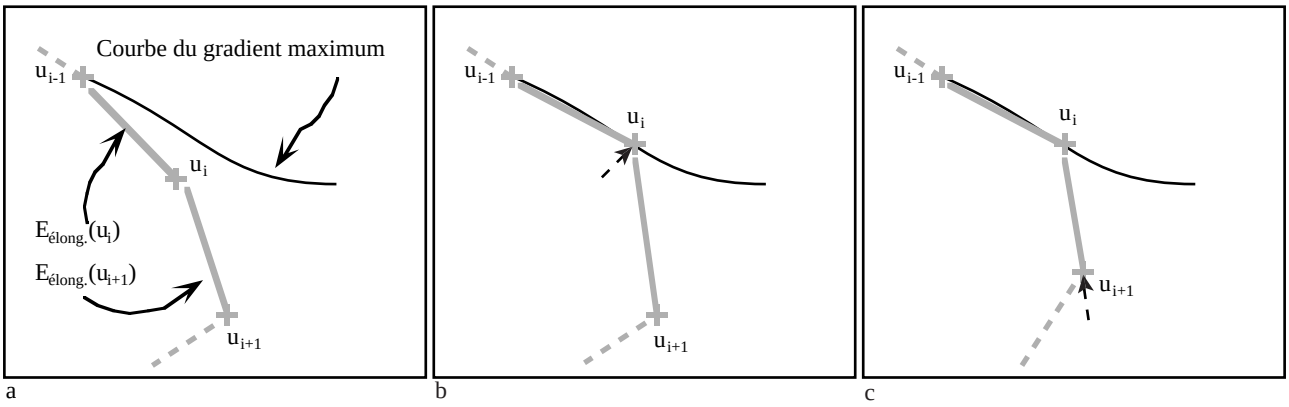


Figure 6.6 : Minimisation par algorithme dynamique.
(a) état initial. (b) le nœud i diminue son énergie externe, l'énergie interne du nœud i+1 augmente.
(c) le nœud i+1 réduit son énergie interne.

Nous avons choisi d'effectuer une réalisation des snakes dont la minimisation d'énergie est basée sur l'algorithme dynamique de [Williams90]. Le détail des choix de conception et des résultats est donné au paragraphe 6.4 traitant de cette expérimentation.

Maintenant que la définition du snake a été détaillée, nous allons nous intéresser à sa mise en oeuvre en tant qu'outil de la vision par ordinateur. Par exemple, ce modèle a été appliqué tel quel avec succès dans des applications de suivi d'automobile ([Koller93]) et d'analyse d'images médicales ([Cohen90]). Nous verrons au paragraphe suivant que de légères adaptations en font un instrument prometteur dans le cadre précis du bureau numérique.

6.3. Application au bureau numérique

La modélisation du snake par une courbe paramétrée $v(s) = (x(s), y(s))$ lui donne une nature 2D. Cette caractéristique essentielle en fait un bon candidat pour le bureau numérique. Nous allons maintenant passer en revue les besoins mis en évidence au chapitre 3 afin d'évaluer son utilisation dans nore contexte applicatif.

6.3.1. Initialisation du suivi

L'initialisation de la position du snake est un problème majeur de ce modèle : le snake n'a aucune connaissance à priori de l'objet auquel il doit s'accrocher. Il est nécessaire de l'initialiser dans une position proche de sa cible, sans quoi il s'accroche sur le premier minimum local d'énergie externe qu'il rencontre. En particulier, les solutions proposées aux paragraphes 3.4.2. et 3.4.3. qui mettent en jeu une recherche du doigt dans la totalité de l'image ne sont pas envisageables ici.

Une solution partielle à ce problème est proposée dans [Cohen90] par l'introduction d'une force tendant à faire enfler le snake comme un ballon. Du moment que celui-ci est initialisé à l'intérieur de sa cible, il enfle jusqu'à être retenu par la barrière que constitue le contour de la cible

sous forme de minimum d'énergie externe. Cependant, cette solution n'est pas applicable à l'initialisation autour d'un doigt car il n'a pas un contour fermé. Même si nous étions capables de placer le snake à l'intérieur du doigt, sa phase d'expansion ne pourrait pas se terminer.

La seule solution que nous voyons impose de faire intervenir l'utilisateur : il s'agit d'afficher un contour sur le bureau dans lequel l'utilisateur vient placer son outil de pointage (doigt, stylo ou gomme). Une détection localisée du mouvement autour du contour affiché (détaillée dans le paragraphe 3.4.1.) permet une activation semi-automatique du suivi.

6.3.2. Simulation du clic souris

Nous avons proposé deux solutions au problème de la simulation d'un clic. Ce problème, on le rappelle, provient de la restriction du suivi à deux dimensions (voir section 3.1.). La première solution imaginée par [Krueger91], est d'associer le clic à une pression du pouce contre l'index. Cette idée s'adapte particulièrement bien à l'utilisation d'un snake pour suivre le contour de la main : le snake modélisé à cet effet a pour cible le contour de la main englobant pouce et index. Une occurence de clic est alors détecté lorsque la partie du contour accrochée à l'extrémité du pouce entre en contact avec une autre partie du contour. Cette situation est représentée dans la figure 6.7.

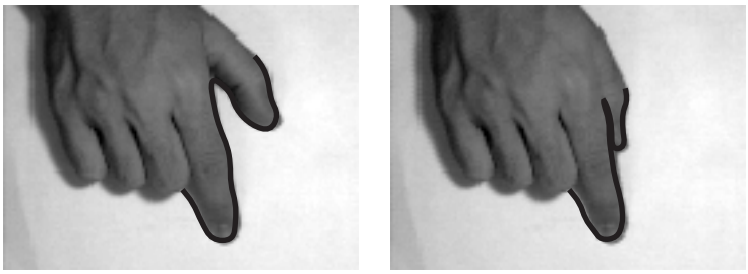
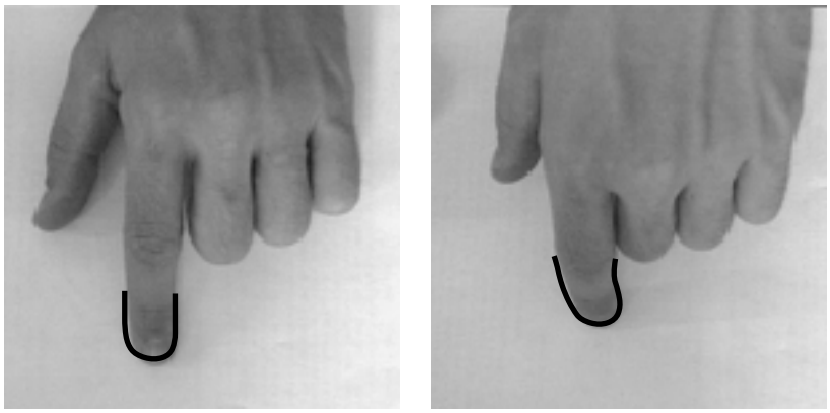


Figure 6.7 : Simulation d'un "click" à l'aide d'un snake.

L'autre solution proposée pour la simulation du clic, nécessite l'utilisation d'une deuxième caméra. Ici, cette approche n'est pas envisageable : nous avons vu au paragraphe précédent que l'initialisation des snakes est problématique car elle impose une intervention de l'utilisateur. S'il est possible de projeter un contour sur le bureau pour que l'utilisateur y introduise son doigt, cette manipulation est impossible dans un plan vertical au bureau, qui est celui de la seconde caméra.

6.3.3. Robustesse

Les snakes ne sont en rien influencés par l'orientation dans le plan de l'image de l'objet suivi. Il sont donc capables de suivre les évolutions sans contraintes du doigt. L'orientation verticale du doigt (définie par l'angle β de la figure 3.8) ne pose pas de problème non plus comme le montre la figure 6.8 : la forme du doigt varie très peu même lorsque la main fait un angle supérieur à 70 degrés.

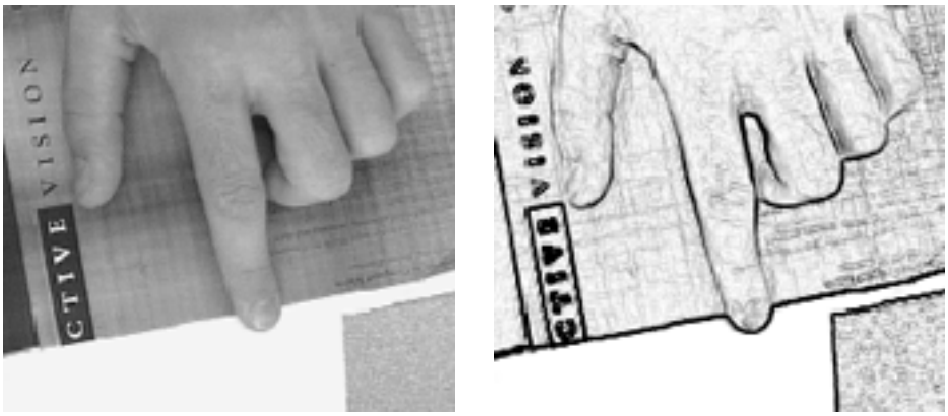


a b

Figure 6.8 : (a) Snake accroché au doigt dans le plan de l'image et (b) formant un angle supérieur à 70 degrés par rapport à ce plan.

Du point de vue des variations d'intensité lumineuse, les snakes ont une grande robustesse puisqu'ils s'accrochent au gradient de l'image. Or celui-ci est toujours maximum sur les contours, même si le contraste est faible ou si la luminosité globale subit de grandes variations.

Par contre, la présence d'un fond bruité perturbe le suivi : si, comme le montre la figure 6.9, le doigt passe au-dessus de deux feuilles de papier, l'une sombre, l'autre claire, le contraste de leur frontière est bien supérieur au contraste du doigt. Le snake reste accroché à cette frontière.



a b

Figure 6.9 : Fond bruité : la frontière entre un fond sombre et clair (a) produit un gradient supérieur à celui du doigt (b).

Nous proposons dans la partie expérimentation une légère modification du modèle du snake lui affectant une "forme préférée" : celle du doigt. Cette spécialisation au suivi du doigt lui apporte une plus grande robustesse par rapport au fond bruité.

6.3.4. Traitement des échecs

La détection de la perte de la cible s'effectue par analyse de la valeur d'énergie du snake : plus celle-ci est grande et plus il y a de chance que la cible soit perdue. Cependant une certaine tolérance au niveau des variations de l'énergie globale doit être prise en compte. En effet, les variations de contraste du doigt entraînent des variations de l'intensité du gradient qui se répercutent dans l'énergie globale.

Le modèle du snake que nous utilisons pour nos expériences a une forme prédéfinie. Il semblerait plus intéressant de considérer uniquement l'énergie interne qui correspond aux déformations du snake par rapport à cette position initiale. Ses variations sont plus réduites que celle de l'énergie globale, elle est donc un témoins plus précis de l'occurence d'un échec.

L'analyse de l'augmentation de l'énergie du snake pourrait également permettre une prévision de la perte de la cible lors du suivi. Cependant il n'existe pas de procédure automatique permettant de revenir à une situation fiable. Il sera donc nécessaire de prévenir l'utilisateur *après* détection de l'échec du suivi afin de le réinitialiser selon le principe exposé au paragraphe 6.3.1.

Pour valider notre étude, nous avons choisi de réaliser un suivi par contours actifs. Nous expliquons dans la section suivante nos motivations, choix de conception et résultats expérimentaux.

6.4. Expérimentation

Le suivi par snakes nous semble un bon candidat à l'intégration dans le système global du bureau numérique. Nous avons donc décidé d'effectuer une série d'expériences sur ce modèle avec FingerPaint, notre banc d'essais présenté au chapitre 4.

Nous commençons par détailler nos choix de conception et les raisons qui les ont motivés. Puis nous donnons les résultats des expériences menées.

6.4.1. Choix de conception

6.4.1.1. Principe général

Nous avons démontré qu'il était inutile de prendre en compte la vitesse courante de la cible dans le calcul de sa position suivante car les accélérations de la main sont très élevées par rapport à sa vitesse de déplacement maximum (ceci est détaillé au paragraphe 3.3.2.). Cette constatation implique deux choix majeurs de conception :

- nous nous limitons au premier modèle des snakes ne faisant pas intervenir de masse ni de frottements (le modèle dynamique est détaillé au paragraphe 6.1.4.),
- la fréquence de fonctionnement doit être privilégiée au détriment de la taille de la zone de recherche afin de rapprocher au maximum les positions successives de la cible. L'algorithme de

minimisation de l'énergie s'appuie sur celui de Williams et Shah ([Williams90]) déjà présenté en détail au paragraphe 6.2.3. Le nombre de nœuds du snake est réduit de 5 à 7.

6.4.1.2. Forme prédéfinie

Le modèle du snake lui donne un comportement tel qu'il se détend en ligne droite et réduit sa taille lorsqu'il n'est soumis à aucune force. C'est la conséquence de la minimisation de son énergie interne (élongation et courbure). Cependant dans notre domaine d'application, nous désirons lui associer une "forme préférée" : celle du bout d'un doigt (l'adaption au bout d'un stylo ou d'une gomme est aisée). Cela nous a amené à deux modifications du modèle initial du snake.

En premier lieu, pour modéliser cette forme préférée avec précision, nous utilisons une énergie de courbure plus rigoureuse que celle définie par l'équation (14). Cette dernière est modélisée comme la norme au carré du vecteur variation (représenté sur la figure 6.5) ce qui ne reflète pas la courbure réelle comme le montre la figure 6.10.

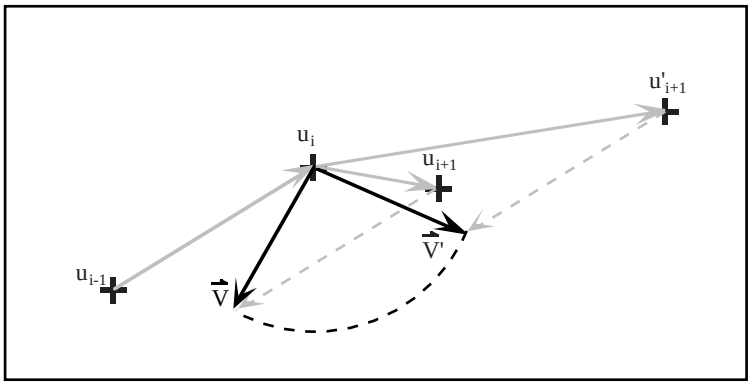


Figure 6.10 : Courbure selon la norme du vecteur variation :
La courbure de [u_{i-1},u_i,u_{i+1}] (norme de $\vec{V}$) est égale à celle de [u_{i-1},u_i,u'_{i+1}] (norme de $\vec{V}'$)

Nous préférons utiliser la courbure réelle qui est représentée par l'angle entre les vecteurs $u_{i-1}u_i$ et u_iu_{i+1}. Notre fonction d'énergie de courbure est donc la forme :

$$E_{courb.}(i) = \operatorname{atan2}(y_{i+1} - y_i, x_{i+1} - x_i) - \operatorname{atan2}(y_i - y_{i-1}, x_i - x_{i-1}) \quad (26)$$

Puis, pour que le snake ait naturellement tendance à adopter sa forme préférée, nous modifions l'équation de son énergie interne afin qu'elle soit minimale (en fait, nulle) lorsque le snake a exactement la forme préférée. Ceci est réalisé en deux étapes :

1) les énergies d'élongation préférée $\left(E^{pref}_{élong.}(i)\right)$ et de courbure préférée $\left(E^{pref}_{courb.}(i)\right)$ sont calculées et mémorisées pour chacun des points lorsque le snake est dans sa forme préférée. La forme de l'énergie d'élongation est identique à celle du modèle initial (basée sur l'équation (13)) et l'équation (26) est utilisée pour la courbure.

2) au cours du suivi, l'énergie interne de chaque point est calculée comme l'écart de ses élongation et courbure par rapport aux valeurs mémorisées en 1) :

$$E_{Interne}(i) = \left| E^{pref}_{élong.}(i) - \left((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \right) \right|$$
$$+ \left| E^{pref}_{courb.}(i) - \left(\text{atan}\,2(y_{i+1} - y_i, x_{i+1} - x_i) - \text{atan}\,2(y_i - y_{i-1}, x_i - x_{i-1}) \right) \right| \quad (27)$$

La première forme de snake que nous avons utilisée est représentée sur la figure 6.11 où les nœuds sont schématisés par des cadres.

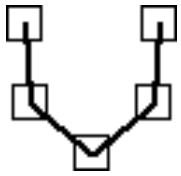


Figure 6.11 : Forme préférée du snake destiné à suivre le bout du doigt.

Les quatre points latéraux ont pour but de suivre les côtés du doigt alors que le point central est destiné à river le snake au bout du doigt (et l'empêcher ainsi de "glisser" le long du doigt).

La formule de l'énergie interne d'un nœud faisant intervenir ses voisins suivants et précédents, il est nécessaire de prévoir des cas particuliers pour les deux extrêmités. Par exemple, il est possible de ne pas donner d'énergie de courbure à ces points. Dans ces conditions, ils n'auraient plus de contrainte et ne seraient plus enclins à respecter la forme préférée. Les nœuds étant numérotés de 0 à NbNœud-1, nous avons utilisé les conventions suivantes :

$$E_{élong.}(0) = E_{élong.}(1)$$
$$E_{courb.}(0) = E_{courb.}(1)$$
$$E_{courb.}(NbNoeud - 1) = E_{courb.}(NbNoeud - 2)$$

6.4.1.3. Force externe

La force externe est le gradient de l'image calculé par le détecteur de Sobel : soient

$$m_1 = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \text{ et } m_2 = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (28)$$

et soit l'image représentée par $I(i, j)$. La force externe du snake au nœud i est définie par :

$$E_{Externe}(i) = \sqrt{G_1(x_i, y_i)^2 + G_2(x_i, y_i)^2} \text{ avec} \quad (29)$$
$$G_1(i, j) = m_1 \otimes I(i, j) \text{ et} \quad (30)$$
$$G_2(i, j) = m_2 \otimes I(i, j) \quad (31)$$

en prenant le symbole $\otimes$ comme opérateur de convolution.

L'avantage des filtres du détecteur de Sobel est qu'ils intègrent un lissage, ce qui étend l'influence du gradient dans l'image (comme expliqué au paragraphe 6.1.3.) :

$$m_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \tag{32}$$

6.4.2. Résultats

La première mesure qui a pu être faite est la fréquence de fonctionnement de l'algorithme (incluant l'affichage vidéo et le dessin du snake). Pour le snake défini par cinq nœuds (représenté par la figure 6.11) et une zone de recherche de 121 points, la fréquence de fonctionnement est 13 Hz.

Ensuite nous avons constaté un échec de l'algorithme à suivre le doigt. Nous expliquons ici les symptômes constatés et le raisonnement qui a permis d'expliquer cet échec.

A l'initialisation, le snake vient se coller correctement sur le contour du doigt. Il y reste tant que le mouvement du doigt est latéral. Le problème apparaît lors d'un mouvement longitudinal : le snake se détache du doigt et pivote autour du premier nœud (numéro 0). Une photo d'écran (commentée) du programme dans cette situation est donné sur la figure 6.12a.

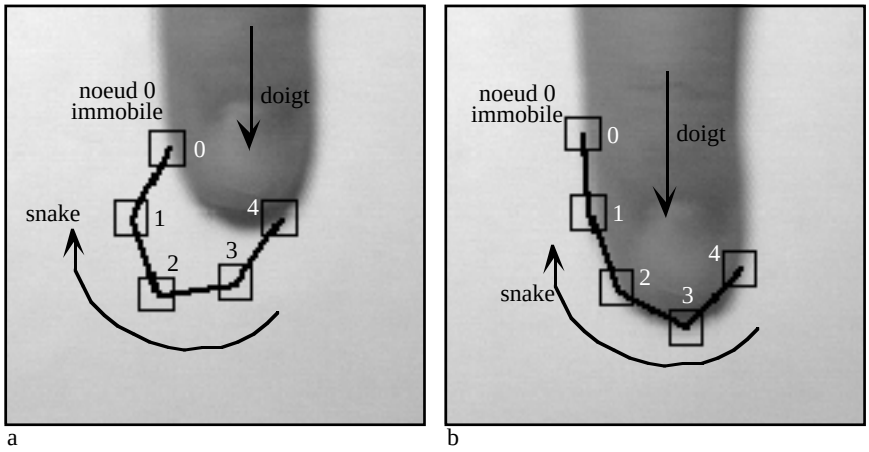


Figure 6.12 : Immobilisation du snake par son premier nœud :
(a) : modèle complet; (b) : contraintes de courbure relachées.

Afin de tester si le problème est lié à la forme du snake, nous relâchons complètement les contraintes d'angle (l'énergie de courbure est annulée). Le problème se reproduit sous une forme légèrement différente : au fur et à mesure que le doigt se déplace, les nœuds du snake glissent sur son côté gauche car les nœuds 0 et 1 semblent immobilisés dans le sens vertical (voire figure 6.12b).

Ceci nous permet d'identifier la raison de l'échec qui se trouve lié à la forme de l'énergie d'élongation. Nous avons utilisé la forme proposée par [Kass87] et [Williams90] qui est la distance au carré entre le nœud et son voisin précédent. Un nœud n'a donc aucune raison d'être attiré par son voisin suivant. Ceci est totalement indépendant de l'ordre d'évaluation de l'énergie des nœuds.

La situaton de la figure 6.12b s'explique ainsi : la minimisation de l'énergie du nœud 2 le laisse sur le contour du doigt (énergie externe minimum) mais sa distance au voisin précédent (nœud 1) reste constante (énergie d'élongation minimum). Le résultat est un glissement du nœud 2 vers le nœud 1. Ce glissement est répercuté sur le nœud 3 qui cherche à garder une distance constante avec son voisin précédent, et ainsi de suite.

Nous modifions alors le modèle pour intégrer dans l'énergie d'élongation à la fois la distance au voisin précédent et suivant. C'est encore un échec lors d'un nouveau mouvement longitudinal du doigt : le nœud central (numéro 2) reste un moment accroché au bout du doigt alors que tous les autres sont presque immobiles. Puis les contraintes de distance l'emportent sur le gradient et le nœud 2 se détache du doigt (figure 6.13).

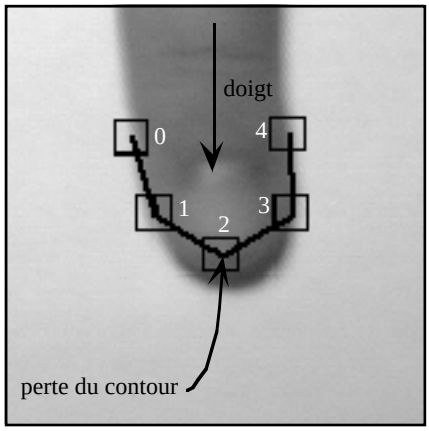


Figure 6.13 : Décrochage du nœud central.

L'interprétation est la suivante : en minimisant à la fois la distance par rapport aux nœuds précédents et suivants, l'attraction entre nœuds est freinée. Prenons le cas des trois nœuds 0, 1 et 2 (figure 6.14a) : lorsque le nœud 2 s'éloigne du nœud 1 pour rester sur une énergie de gradient minimum, il serait souhaitable qu'à son tour le nœud 1 se rapproche afin de rétablir la distance initiale (figure 6.14b). Or un déplacement vers le nœud 2 implique un éloignement d'une distance supérieure du nœud 0, donc une augmentation de l'énergie d'élongation. Le nœud 1 se stabilise à égale distance entre les nœuds 0 et 2, ce qui n'est pas suffisant pour compenser l'éloignement au nœud 2 (figure 6.14c).

On vérifie expérimentalement qu'en-dessous d'une certaine vitesse de déplacement, au fur et à mesure des itérations, les nœuds latéraux ont le temps de se rapprocher du nœud central. Le suivi du doigt est donc également réalisé dans le sens longitudinal.

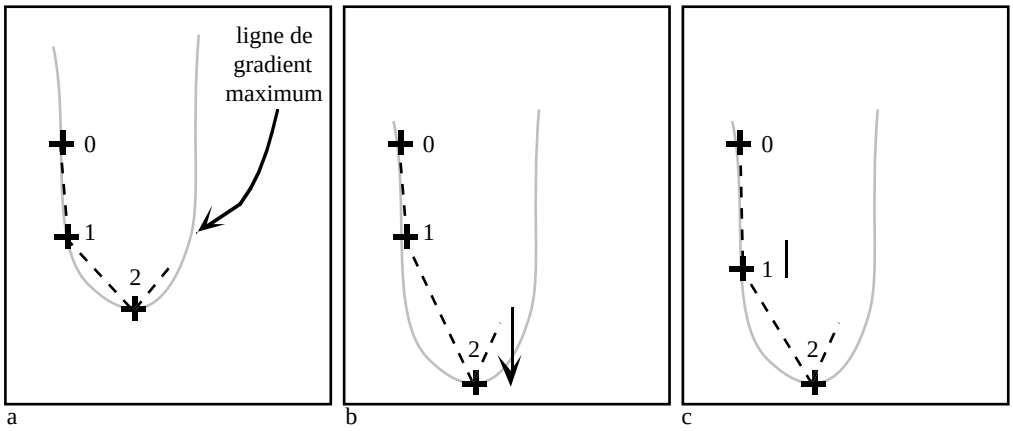


Figure 6.14 : Augmentation de l'énergie interne du snake.
(a) : état initial, (b) : augmentation de l'énergie d'élongation entre 1 et 2,
(c) : réduction partielle de l'énergie d'élongation globale.

Cependant ce modèle du snake a une grande inertie du fait du manque de liberté des nœuds pour se déplacer. En particulier une rotation du doigt dans le plan du bureau déforme le snake et le fait rapidement perdre sa cible.

Cette série d'expériences met en évidence que le modèle et l'algorithme dynamique des snakes de Williams et Shah [Williams90] ne peuvent être utilisés tels quels dans notre domaine d'application. Le succès de la minimisation par l'algorithme dynamique semble fondé sur la liberté de mouvement des nœuds et leur possibilité de se déplacer le long du contour suivi. Nous avons vérifié cette hypothèse en réalisant un snake respectant fidèlement les choix de [Williams90]. En particulier nous avons de nouveau utilisé une énergie d'élongation basée sur la distance au voisin précédent. Mais surtout nous utilisons un contour fermé.

La figure 6.15 montre une photo d'écran d'un snake de ce type accroché à un disque dessiné sur une feuille de papier. Le glissement des points le long du contour observé lors de nos premières expérimentations se traduit ici par une rotation du snake autour du disque.

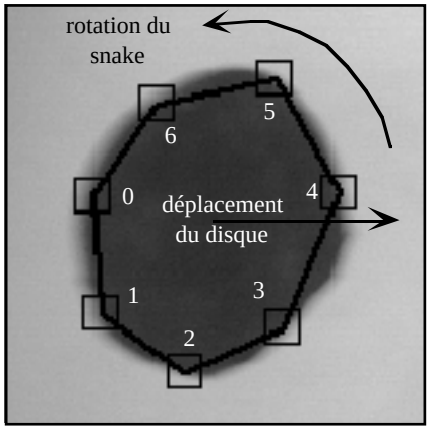


Figure 6.15 : Suivi d'un contour fermé.

Pour conclure, si l’objectif d’un suivi fiable du doigt n’a pas encore été atteint, les résultats obtenus, notamment sur contour fermé, montrent que la voie des contours actifs est digne d’intérêt. Il est important de poursuivre l’étude, notamment en s’éloigant légèrement du modèle général des snakes pour réaliser un algorithme de minimisation d’énergie spécialisé au contour du type particulier du doigt.

CONCLUSION

Nous aimerions conclure ce rapport en résumant les points contributifs de cette étude puis en présentant les prolongements possibles.

Contribution

Ce travail contribue au domaine des interfaces homme-machine sous deux formes complémentaires : la définition d'un espace de classification des nouvelles interfaces et des recommandations pour la vision par ordinateur.

Espace de classification

Notre espace de classification, bien que simple, a le mérite de distinguer les mondes réels des mondes virtuels, les réalités virtuelles des réalités augmentées. Dans un univers où les progrès techniques transforment radicalement la morphologie des stations de travail et de là, les conditions de travail, nous assistons à une prolifération de termes aux contours mal cernés. Notre espace à deux dimensions (objet et action) permet de donner corps à ces termes :

- nous dirons qu'un monde est virtuel si ses objets et les actions qu'il permet sont réalisés électroniquement et uniquement sous cette forme. Tel est le cas des systèmes actuels de bureautique ;
- la réalité est virtuelle lorsque les actions du monde physique sont appliquées à des objets électroniques : typiquement, l'utilisateur, auteur des actions physiques, est immergé dans un univers simulé ;
- nous parlerons de réalité augmentée si les objets physiques sont amplifiés de fonctions électroniques. Par exemple, un texte sur support papier est enrichi d'une fonction de recherche électronique.

Dans cet espace d'interfaces avancées, nous avons retenu un exemple prometteur de réalité augmentée : le bureau numérique. Ce concept vise à intégrer de manière harmonieuse les activités du monde réel et les capacités fonctionnelles de l'électronique. Le succès de cette entreprise repose en partie sur une solution technique : l'usage de la vision par ordinateur.

La vision par ordinateur devrait, dans les années à venir, avoir une place de choix dans la communication homme-machine. Elle a l'avantage, nous l'avons vu, de ne pas faire intrusion dans l'activité de l'utilisateur. En dépit de ses qualités "écologiques", elle n'a été que peu étudiée dans les IHM.

Recommandations pour la vision par ordinateur

La seconde contribution de ce travail est l'étude de l'usage de la vision par ordinateur dans le cadre précis du bureau numérique. Dans ce contexte, nous avons choisi d'étudier en détail un problème incontournable : le suivi d'objets comme le doigt, le crayon, la gomme. Nous avons retenu le suivi dans un plan 2D (celui de la table de travail) sachant que le posé et le relevé peut être détecté par d'autres artifices : utilisation d'une seconde caméra ou d'un microphone intégré au bureau. Nous avons testé cette deuxième solution et indiqué qu'elle était trop sensible aux bruits de l'environnement (pose brutale d'objets sur la table, par exemple). Nous recommandons l'usage d'une seconde caméra dédiée à la détection de lever-poser des objets pertinents.

Le suivi 2D ayant été retenu, nous avons recruté deux familles de suivi qui font référence dans ce domaine : le suivi par corrélation et le suivi par contour actif (snake). Nous avons évalué leur adéquation aux besoins du bureau numérique en fonction de quatre critères que nous avons estimés essentiels : robustesse, facilité d'adaptation à différentes formes d'objet, traitement des échecs et simplicité d'intégration. Chaque méthode a été appliquée à la mise en œuvre d'un prototype sous-ensemble des services du bureau numérique : FingerPaint. Ce système permet, pour l'essentiel, de dessiner une figure libre avec un doigt.

Nos expérimentations ont mis en évidence les points suivants :

- le suivi par corrélation est rapide aussi bien dans sa mise en œuvre que dans sa fréquence de fonctionnement. Il n'exige pas une allocation régulière de la ressource processeur. Il manque, hélas, de robustesse. Ce déficit ne semble pas pouvoir être comblé : cette technique n'a pas de faculté d’adaptation au suivi spécifique d'un doigt, condition indispensable au bureau numérique.
- le suivi par snake est plus lourd à mettre en place et nécessite plus de ressources de calcul que le suivi par corrélation mais il est plus robuste. Le modèle général du snake n’est pas capable de fournir le suivi d’une forme telle que le bout du doigt mais nous avons su l'adapter. La spécialisation d’un snake sur une forme donnée telle que le doigt est réalisable.

En synthèse, notre étude technique indique que le suivi par snake est mieux adapté à notre problème que la corrélation. De manière générale, la vision par ordinateur a un rôle à jouer dans la réalité augmentée et notamment comme système de suivi d'objets physiques dans un plan de travail. Le problème est d'envergure mais reste dans l'épure du réalisable. Il convient alors de s'interroger sur les perspectives.

Perspectives

Ce travail voit son prolongement selon deux axes d'étude : la composante IHM et la composante vision.

Composante IHM

Dans le chapitre introductif, nous avons évoqué le principe directeur auquel notre travail doit obéir obstinément : l'utilisabilité. Maintenant que nous disposons d'un prototype, il conviendrait d'en tester l'utilisabilité pour une tâche donnée de tracé de dessin. Nous pourrions mesurer les performances d'un ensemble représentatif d'utilisateurs (nombre d'erreurs, temps d'exécution, appréciations subjectives, etc.) et les comparer avec celles obtenues au moyen d'un logiciel classique de dessin tel MacDraw ou dans le monde réel avec papier-crayon.

Il conviendrait aussi de tester l'impact des tâches parasites qu'introduisent nos solutions techniques ou nos hypothèses de solution. Par exemple, le placement du doigt dans une zone réservée pour déclencher le système de suivi ou bien le maintien du doigt jusqu'à sa détection par le système.

Composante vision par ordinateur

A court terme, un premier travail consisterait à améliorer la fiabilité de l'algorithme des snakes. Notre approche reviendra à spécialiser le modèle en exploitant les spécificités du doigt.

Adoptant une approche incrémentale, nous avons restreint l'analyse au suivi d'un seul objet. Il faudra étendre nos résultats au suivi de plusieurs objets en parallèle : cas des deux mains travaillant simultanément à la manipulation d'objets réels ou virtuels.

A plus long terme, il serait intéressant d'exploiter les propriétés de la vision active pour la capture contrôlée de l'image. Nous avons proposé une piste au paragraphe 2.3.3 qu'il nous faudrait explorer.

En synthèse, cette étude prospective ouvre de nouvelles voies dans la communication homme-machine et pose des problèmes techniques spécifiques auxquels la vision par ordinateur doit pouvoir répondre.

BIBLIOGRAPHIE

[Anadan87] P. Anadan. *"Measuring Visual Motion From Image Sequence"*. Phd dissertation and COINS Technical Report 87-21, University of Massachussetts, Amherst, 1987.

[Apple88] Apple Computer Inc.*"The Macintosh User Interface Guidelines"*. Dans Inside Macintosh Vol. I-VI, Addison-Wesley, 1988.

[Apple93] Apple Computer Inc. *"Inside Macintosh - QuickTime"*. Addison-Wesley, 1993.

[Aschwanden92] P. Aschwanden, W. Guggenbühl. *"Experimental Results from a Comparative Study on Correlation-Type Registration Algorithms"*. in "Robust Computer Vision" pp. 268-289, Förstner and Ruwiedel, Wichmann Publisher, 1992.

[Aschwanden88] P. Aschwanden. *"Real-time Tracker with Signal Processor"*. Signal Processing IV : Theories and Applications. Elsevier Science Publishers, 1988.

[Austin62] J. L. Austin. *"How to do thing with words"*. Oxford : Clarendon Press, 1962.

[Barnard93] P. Barnard, J. May. *"Real time blending of data streams: a key problem for the cognitive modelling of user behaviour with multimodal systems"*. UM/WP10, User Modelling, Working Paper 10, The Amodeus Project, Esprit Basic research Action 7040, (9 Juin 1993).

[Balbo93] S. Balbo, J. Coutaz, D. Salber. *"Towards Automatic Evaluation of Multimodal User Interfaces"*. International Workshop on Intelligent User Interfaces, Orlando, USA, Jan., 1993.

[Bier93] E. Bier, M.C. Stone, K.Pier, W. Buxton, T. D. DeRose. Toolglass and Magic Lenses: The See-Through Interface. Proceedings of the Siggraph'93 (Anaheim, August), Computer Graphics Annual Conference Series, ACM, 1993, pp. 73-80.

[Bier94] E. Bier, M.C. Stone, K. Fishkin, W. Buxton, T. Baudel. A taxonomy of See-Through Tools. Proceedings of CHI'94, ACM, 1994, pp. 358-364.

[Burdea93] G. Burdea, P. Coiffet. *"La Réalité Virtuelle"*. Hermes, 1993.

[Cadoz94] C. Cadoz. *"Le geste canal de communication homme/machine. La communication instrumentale"*. Technique et science informatiques, Vol. 13, No 1, 1994.

[Card83] S. K. Card, T. P. Moran, A. Newell. *"The Psychology of Human-Computer Interraction"*. Lawrence Erlbaum Associates, 1983.

[Cohen90] L. D. Cohen, I. Cohen. *"A finite element method applied to new active contour models and 3D reconstruction from cross sections"*. Proc. 3rd International Conf. on Computer Vision, pp. 587-591, 1990.

[Coutaz90] J. Coutaz. "Interfaces hommes-ordinateur. Conception et réalisation.". Dunod Informatique, 1994.

[Coutaz93] J. Coutaz, D. Salber, S. Balbo. *"Towards Automatic Evaluation of Multimodal User Interfaces"*. Journal on Knowledge-Based Systems, special issue on intelligent user interfaces, (1993).

[Coutaz94] J. Coutaz. *"Interface Homme-Machine"*. Support de cours du DEA, INPG, 1994.

[Crowley94] J.L. Crowley, H. Christensen. "Vision as Process: Integration and Control of Real Time Active Vision System" dans *Experimental Environments for Computer Vision and Image Processing*, Series in Machine Perception and Artificial Intelligence, Vol. 11, World Scientific Pub., H.I Christensen, J.L. Crowley Eds., pp 127-155, 1994.

[Curwen92] R. Curwen, A. Blake. *"Dynamic Contours : Real-time Active Splines"*. in "Active Vision", The MIT Press, 1992.

[Dix93] A. Dix, J. Finlay, G. Abowd, R. Beale. *"Human-Computer Interaction"*. Prentice Hall, 1993.

[Fahlén93] L. E. Fahlén, C. G. Brown, O. Ståhl, C. Carlsson. "A Space Based Model for User Interaction in Shared Synthetic Environments". Proceedings InterCHI'93, pp 43-48.

[Feiner93] S. Feiner, B. Macintyre, D. Seligmann. *"Knowledge-Based Augmented Reality"*. Communications of the ACM, Vol.36 No.7, July 1993.

[Fitzmaurice93] G. W. Fitzmaurice. *"Situated information sources and spatially aware palmtop computers"*. Communications of the ACM, Vol.36 No.7, July 1993.

[Frohlich91] D. M. Frohlich. *"The Design Space of Interfaces, Multimedia Systems, Interaction and Applications"*. Proceedings of 1st Eurographics Workshop, Stockholm, Sweden (Avril 18/19,1991), Springer Verlag, pp. 53-69.

[Harris92] C. Harris. *"Tracking with Rigid Models"*. in "Active Vision", The MIT Press, 1992.

[Hinckley94] K. Hinckley, R. Pausch, J. C. Goble, N. F. Kassel. "Passive Real-World Interface Props for Nerosurgical Visualisation". Proceedings CHI'94, pp 452-458.

[Hutchins86] E. L. Hutchins, J. D. Hollan, D. A. Norman, *"Direct Manipulation Interfaces"*. Dans "User Centered System Design, New Perspectives on Computer Interaction" édité par D. A. Norman, S.W. Draper, Hillsdale, New Jersey : Lawrence Erlbaum Associates, (1986), pp. 87-124.

[Johnson93] W. Johnson, H. Jellinek, L. Klotz, Jr. R. Rao, S. Card. *"Bridging the Paper and Electronic Worlds : The Paper User Interface"*. Proceedings InterCHI'93, pp 507-512.

[Kass87] M. Kass, A. Witkin, D. Terzopoulos. *"Snakes : Active Contour Models"*. Proc. 1st International Conf. on Computer Vision, pp. 259-268, 1987.

[Koller92] D. Koller, K. Daniilidis, T. Thórhallson, H. H. Nagel. *"Model-Based Object Tracking in Traffic Scenes"*. Proc. 2nd European Conf. on Computer Vision, pp. 437-452, 1992.

[Koller93] D. Koller, J. Weber, J. Malik. *"Robust Multiple Car Tracking with Occlusion Reasoning"*. Technical report UCB/CSD-93-780, University of California at Berkeley, Octobre 1993.

[Krueger91] M. W. Krueger. *"Artificial Reality II"*. Addison Wesley, 1991.

[Laurel86] B. K. Laurel. *"Interface as Mimesis"*. Dans "User Centered System Design, New Perspectives on Computer Interaction" édité par D. A. Norman, S.W. Draper, Hillsdale, New Jersey : Lawrence Erlbaum Associates, (1986), pp. 67-85.

[Maes94] P. Maes, T. Darrel, B. Blumberg, S. Pentland. *"The ALIVE system : Full-Body Interaction with Animated Autonomous Agent"*. M.I.T. Media Laboratory Perceptual Computing Technical Report No. 257, Jan. 1994.

[Maury94] S. Maury. *"Suivi du Regard"*. DEA Informatique de l'INPG et l'UJF, Grenoble, juin 1994.

[Mignot93] C. Mignot, C. Valot, N. Carbonell. "An Experimental Study of Future 'Natural' Multimodal Human-Computer Interaction". InterCHI'93 Adjunct Proceedings, pp. 67-68.

[Moravec80] H. P. Moravec. *"Obstacle Avoidance and Navigation in the Real World by a Seing Robot Rover"*. Phd Thesis, Stanford University, 1980.

[Meyer92] F. Meyer, P.Bouthemy. *"Region-Based Tracking in an Image Sequence"*. Proc. 2nd European Conf. on Computer Vision, pp 476-484, 1992.

[Newman92] W. Newman, P. Wellner. *"A Desk Supporting Computer-based Interaction with Paper Documents"*. Proceedings CHI'92, pp. 587-592.

[Nigay94] L. Nigay. *"Conception et Modélisation logicielles des systèmes interactifs"*. Thèse préparée à l'université Joseph Fourier, Grenoble, 1994.

[Norman86] D. A. Norman. *"Cognitive Engineering"*. Dans "User Centered System Design, New Perspectives on Computer Interaction", édité par D. A. Norman, S.W. Draper, Hillsdale, New Jersey : Lawrence Erlbaum Associates, (1986), pp. 31-61.

[Rehg93] J. M. Rehg, T, Kanade. *"DigitEyes : Vision-Based Human Hand Tracking"*. Carnegie Mellon University Technical Report CMU-CS-93-220, December 1993.

[Salber93] D. Salber, J. Coutaz. *"Applying the Wizard of Oz Technique to the Study of Multimodal Systems"*. In Human Computer Interaction, 3rd International Conference EWHCI'93, East/West Human Computer Interaction, Moscow. L. Bass, J. Gornostaaev, C. Unger Eds. Springer Verlag Publ., Lecture notes in Computer Science, Vol. 753, 1993, pp.219-230.

[Shaw92] C. Shaw, J. Liang, M. Green, Y. Sun. *"The Decoupled Simulation Model for Virtual reality Systems"*. Proceedings CHI'92, pp 321-328.

[Terzopoulos87] D. Terzopoulos, A. Witkin, M. Kass. *"Energy constraints on deformable models : Recovering shape and non-rigide motion"*. Proc. International Conf. on Computer Vision, pp266-275, 1992.

[Terzopoulos92] D. Terzopoulos, R. Szeliski. *"Tracking with Kalman Snakes"*. in "Active Vision", The MIT Press, 1992.

[Ueda92] N. Ueda, K. Mase. *"Tracking Moving Contours Using Energy-Minimizing Elastic Contour Models"*. Proc. 2nd European Conf. on Computer Vision, pp 453-457, 1992.

[Waterworth94] J. A. Waterworth, L. Serra. "VR Management Tools : Beyond Spatial Presence". CHI'94 Conference Proceedings, pp 319-320.

[Wellner93a] P. Wellner, Wendy Mackay, Rich Gold. *"Computer-augmented environments : back to the real world"*. Communications of the ACM, Vol.36 No.7, July 1993.

[Wellner93b] P. Wellner. *"Interacting with paper on the DigitalDesk"*. Communications of the ACM, Vol.36 No.7, July 1993.

[Williams90] D. J. Williams, M. Shah. *"A Fast Algorithm for Active Contours"*. Proc. 1st European Conf. on Computer Vision, pp 592-595, 1990.

Résumé

Ce travail a pour thème l'utilisation de la vision par ordinateur dans le contexte du bureau numérique (digital desktop). Le concept de bureau numérique, imaginé à EuroPARC, vise à créer une symbiose entre le monde électronique et les objets réels manipulés en bureautique. Par exemple, avec une gomme réelle on peut effacer une figure MacDraw projetée sur la table de travail. Cette synergie entre le réel et le virtuel définit une "réalité augmentée" (par opposition aux réalités virtuelles et artificielles qui utilisent le tout-digital pour reproduire ou ré-inventer le monde réel).

Dans le cadre de cette étude, on explore des techniques de la vision par ordinateur pour détecter et interpréter les mouvements des mains lorsqu'un utilisateur manipule des objets électroniques ou des objets réels d'un bureau numérique. Plus précisément , il s'agit de localiser et de suivre en temps réel le mouvement des doigts dans un espace de travail planaire tel le bureau numérique. Une étude comparative des techniques de suivi en temps réel à base de corrélation ou des "snakes" est présentée.

Mots-clefs : IHM, Bureau Numérique, Vision par Ordinateur, Suivi d'objets.