Laboratoire DOMUS
Recherche en domotique et
en informatique mobile

UNIVERSITÉ DE
SHERBROOKE

Rapport de stage de fin d'études

Éffectué dans le laboratoire DOMUS
à l'Université de Sherbrooke

Pour l'obtention du
Diplôme National d'Ingénieur

Présenté et soutenu par
Wathek Bellah LOUED

Analyseur d'émotions à partir d'expressions faciales

Encadré par :

Madame Hélène Pigot	**Responsable DOMUS**
Madame Syrine Tlili	**Enseignante ENICarthage**

Année Universitaire 2013-2014

Remerciements

Je tiens naturellement à adresser mes premiers remerciements à Madame Hélène Pigot, qui m'a non seulement offert cette enrichissante opportunité, mais aussi qui m'a encadré pour ce travail de fin d'études. Je la remercie aussi pour la confiance qu'elle m'a accordé et je lui suis reconnaissant pour m'avoir fourni un cadre idéal pour la réalisation de mon stage.

Je tiens aussi à remercier Madame Syrine Tlili pour m'avoir guidé et conseillé tout au long de mon travail et surtout de m'avoir encouragé à faire mon travail de fin d'études à l'Université de Sherbrooke.

Je remercie aussi Monsieur Taoufik Zayani pour m'avoir, tout d'abord, aidé à m'intégrer, mais aussi pour ses précieux conseils et surtout pour m'avoir aidé à réaliser ce travail dans d'excellentes conditions. Je lui suis aussi très reconnaissant pour m'avoir hébergé pendant quelques jours et pour m'avoir soutenu pendant toute cette période.

Je remercie les membres du Laboratoire DOMUS qui m'ont accueilli et m'ont permis d'apprécier le travail en équipe dans un environnement de partage de connaissances.

Je tiens à remercier, toute l'équipe pédagogique de l'École Nationale d'Ingénieurs de Carthage (anciennement ESTI) et les intervenants professionnels responsables de la formation "Génie Informatique Appliquée".

J'adresse enfin mes remerciements à Jeremy Deramchi pour m'avoir aidé financièrement et pour m'avoir soutenu durant mon stage.

Enfin je ne pourrais terminer ce chapitre sans citer mes parents, mon frère et Ghofrane pour leur inconditionnel support.

Présentation du laboratoire DOMUS

Le Laboratoire DOMUS est un laboratoire de domotique et d'informatique mobile à l'Université de Sherbrooke. Depuis 2002, le laboratoire DOMUS étudie l'assistance cognitive, le suivi médical et la télévigilance auprès des personnes avec troubles cognitifs. Les résultats de ses recherches sont applicables aux personnes atteintes de déficits cognitifs imputables en particulier à des traumatismes crâniens, à la schizophrénie, à la maladie d'Alzheimer et à la déficience intellectuelle.

La variété des recherches interdisciplinaires qui y sont menées montre combien l'assistance cognitive, la télévigilance et le suivi médical posent des défis énormes pour le maintien à domicile.

Le laboratoire DOMUS entretient des partenariats régionaux, québécois, canadien et internationaux. Ces partenariats proviennent, du monde universitaire et de la recherche, des intervenants du milieu de la santé et de l'entreprise privé.

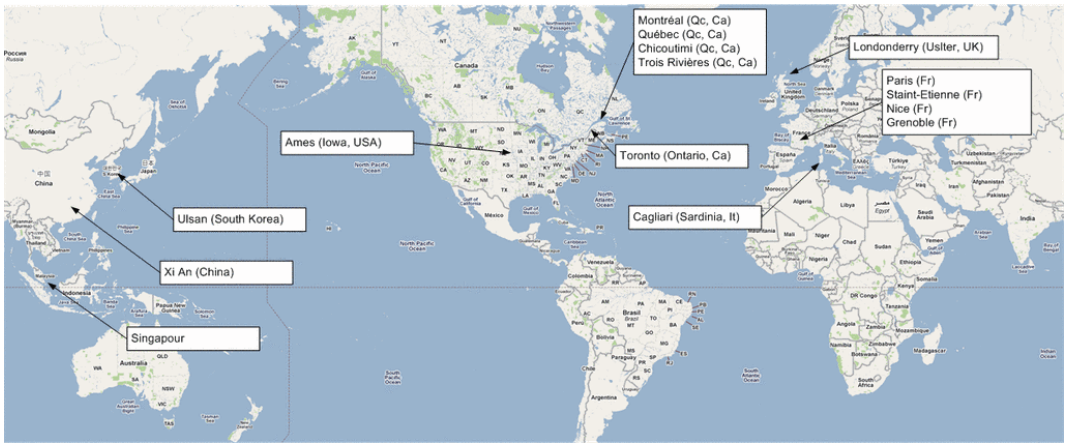

Fig. 1 Partenaires du DOMUS

L'objectif des recherches poursuivies au laboratoire, est de transformer en véritable orthèse cognitive l'habitat pris dans son sens large : résidences, lieux semi-publics, lieux publics, et ce tant à l'intérieur qu'à l'extérieur des édifices.

D'une part, le domicile devient alors un support cognitif personnalisé, contextuel et adaptatif capable de pallier les déficits cognitifs. D'autre part, à l'extérieur, des systèmes mobiles l'aident à se déplacer et à gérer son temps, lui donnent des consignes en fonction du lieu où il se trouve, recueillent des données écologiques indispensables pour un meilleur suivi médical.

Le laboratoire DOMUS est considéré non seulement comme étant un des meilleurs laboratoires dans le domaine de la domotique mais aussi comme étant un des mieux équipés.

Introduction générale

Le monde vit un changement démographique important. La population mondiale a été estimée à 7 milliards en 2011, et selon les Nations Unions, elle va atteindre les 7.2 milliards en 2014. Selon l'Organisation Mondiale de la Santé, le monde comptera bientôt plus de personnes âgées que d'enfants. Ceci est dû essentiellement à l'amélioration du secteur de la santé, qui a augmentée l'espérance de vie chez les personnes âgées.

Au Québec, et d'après la direction de la recherche de l'évaluation et de la statistique du ministère de la Famille et des Aînés [20], en 2011 sur près de 8 millions de Québecois on compte près de 3 millions de personnes âgées de 50 ans ou plus et 1.3 million de personnes âgées de 65 ans ou plus.

D'un autre côté, selon la fondation Martin-Matte [1], chaque année il y a 13 000 nouvelles victimes d'un traumatisme crânien, dont 3 600 qui ne retrouveront jamais leur autonomie. Près de 45% de ces traumatismes crâniens sont causées par des accidents de la route.

Pour cela l'assistance des personnes âgées et des personnes avec traumatisme crânien est très importante et demande des ressources importantes, et ce pour essayer de permettre aux personnes de garder, ou de récupérer, leur autonomie. Une des activités importante dans la vie quotidienne est la préparation de repas. Cependant cette tâche peut avoir des niveaux de difficultés différents, c'est à dire, la personne doit décider de la recette à préparer, par la suite préparer les ingrédients, trouver les ustensiles, faire la préparation, etc. La personne peut aussi être amenée à utiliser des appareils à risque et qui peuvent être dangereux (par exemple la cuisinière).

Le problème majeur avec les personnes âgées et les personnes avec traumatisme crânien, c'est que plusieurs réflexes naturels sont atténués, en plus elles peuvent avoir certaines pertes sensorielles ou perte de mémoire, etc. Tout ceci peut être contraignant lors d'une préparation d'un repas. Pour cela elles ont besoin d'un assistant culinaire intelligent qui les guide durant tout le processus et ce en indiquant étape par étape ce qu'il faut faire pour arriver à préparer un repas choisi par la personne.

Cet assistant exploite plusieurs paramètres de l'environnement par exemple la position de la personne dans la cuisine, ou bien la position des ustensiles, l'état des éléments (cuisinière, robinet d'eau, tiroirs, etc) et bien d'autres informations renvoyées par les capteurs. Cet assistant peut même aider la personne à trouver un ustensile en indiquant sa position grâce aux effecteurs tels que les témoins lumineux, les hauts parleurs, les films opaques, etc.

Une autre information que l'assistant exploite, pour pouvoir éviter les accidents, est l'état de la personne. C'est à dire qu'il y a une analyse de la personne pour savoir dans quel état émotionnel elle se trouve. Par exemple si la personne est en colère, ceci peut avoir des conséquences et peut donc la pousser à commettre des accidents. Les émotions peuvent être

1. http ://www.fondationmartinmatte.com/statisitiques/

considérées comme étant un élément important pour savoir comment guider et surtout éviter les accidents, ou bien inciter la personne à se reposer et reprendre l'activité ultérieurement.

Le travail qui suit consiste à créer l'analyseur des émotions qui va estimer les émotions d'une personne et renvoyer cette information à l'assistant culinaire. Pour cela il y a cinq volets qui peuvent être étudiés :

— Les expressions faciales.
— Le mouvement du corps.
— La voix.
— Les signes vitaux.
— Les capteurs dans la cuisines (Fig. 2).

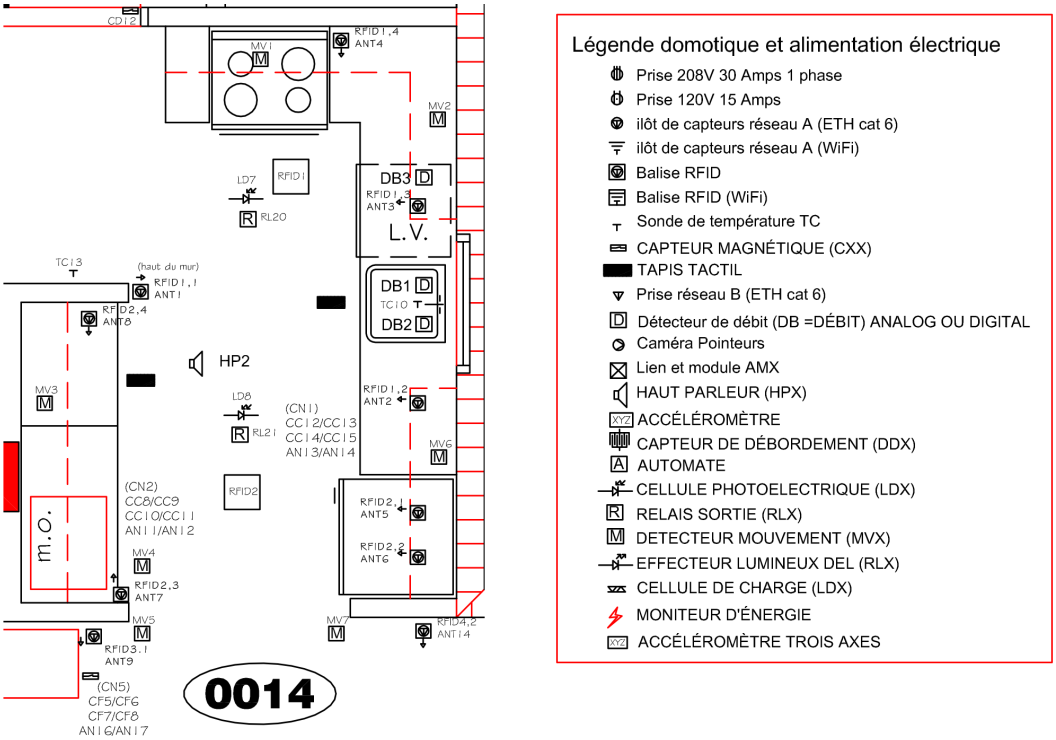

Fig. 2 Plan de la cuisine et disposition des capteurs

Pour commencer, nous nous sommes intéressés plus aux expressions faciales qu'aux autres parties, et ce parce que le visage de l'être humain est plein d'informations qu'il est possible d'utiliser, et surtout que les émotions sont le plus souvent exprimées, d'une façon involontaire et innée, par le visage en premier lieu.

Ce rapport est organisé en cinq chapitres. Au cours du premier chapitre, une étude théorique des émotions et du comportement de l'être humain a été faite. Cette étude permet de mieux comprendre les émotions et comment elles sont exprimées.

Dans le deuxième chapitre, nous présentons l'objectif et les méthodologies existantes, autrement dit les algorithmes disponibles pour pouvoir détecter le visage, trouver les points caractéristiques dans un visage et finalement l'estimation de l'émotion.

Dans le troisième chapitre, nous évoquons les spécifications du système, et ce en énumérant les besoins fonctionnels et les besoins non fonctionnels de l'analyseur des émotions. Dans ce même chapitre nous présentons aussi les technologies logiciels utilisées pour arriver à réaliser ce projet.

Le quatrième chapitre décrit l'implémentation de l'analyseur des émotions. Ce chapitre est divisé en trois parties importantes :

1. La détection du visage pour savoir si dans l'image il y a bien un visage d'un être humain ou pas.
2. Le placement des points caractéristiques (ou "Landmarks") qui seront enregistrés et suivis pour pouvoir estimer l'émotion.
3. La détection de l'émotion qui se fait en comparant les données captées par rapport aux données modèles pour pouvoir faire la classification et donner une estimation de l'émotion.

Ces trois parties sont exécutées en cascade, c'est à dire que si la première étape renvoie un résultat négatif (absence d'un visage) les deux autres étapes ne seront pas exécutées. La détection des émotions, dans ce travail, se base sur une séquence d'images (une vidéo) du visage de la personne. Habituellement la détection des émotions est réalisée sur une seule image[1][7]. Dans ce même chapitre nous expliquons tous les apprentissages qui ont été faits, ainsi que le processus de normalisation des données et la création de modèles de références des émotions qui seront utilisés pour la classification.

Et finalement dans le cinquième chapitre nous présentons les tests qui ont été faits en utilisant la base de données des émotions [15] fournis par l'Université de Pittsburgh. Nous présentons aussi les résultats obtenus et les éventuels évolutions qu'il sera possible d'apporter pour améliorer l'analyseur des émotions.

Table des matières

Remerciements **ii**

Présentation du laboratoire DOMUS **iii**

Introduction générale **iv**

Table des matières **1**

Liste des tableaux **3**

Table des figures **4**

1 Étude comportementale et émotionnelle **6**
1.1 Affect, Humeur et Émotion 7
1.2 Expressions faciales 9
1.2.1 "*Facial Action Coding System*", "*Action Unit*" et "*Action Descriptor*" 10
1.2.2 Groupes des "*Action Units*" 11
1.2.3 Combinaison de plusieurs AUs 12
1.3 Autres moyens pour détecter les émotions 12

2 Objectif et méthodologies **14**
2.1 Objectif 14
2.2 Méthodologies 15
2.2.1 Détection du visage 15
2.2.2 Détection du visage dans un environnement libre 16
2.2.3 Détermination des points caractéristiques (ou "Landmarks") 16
2.2.4 Détection des émotions 17

3 Spécification du système de détection d'émotions **19**
3.1 Besoins fonctionnels et non fonctionnels 19
3.1.1 Besoins fonctionnels 19
3.1.2 Besoins non fonctionnels 19
3.2 Spécification du système 20
3.2.1 Spécification de la couche matérielle 20
3.2.2 Spécification de la couche logicielle 21

3.3 Choix technologique 22
3.3.1 Langages de programmation 22
3.3.2 Librairies 22
OpenCV 22
STASM 23
Threading Building Blocks 23

4 Implémentation du détecteur d'émotions 24
4.1 Détection du visage dans une image 24
4.1.1 Haar Cascades 24
Caractéristiques Pseudo-Haar 24
Image Intégrale 26
Classificateur fort 28
Classificateur en cascades 29
Implémentation 30
4.2 Suivi du visage dans un flux vidéo 31
Implémentation 31
4.3 Détermination des points caractéristiques (ou "Landmarks") 32
4.3.1 Description du fonctionnement de l'algorithme ASM 32
Préparation des données d'apprentissage 33
Normalisation des données d'apprentissage 34
Calcul du modèle statistique 36
Application du modèle sur une image 38
4.3.2 Améliorations de l'algorithme ASM 39
Description du "Scale-Invariant Feature Transform" 40
Description du "Histogram Array Transforms" 41
4.3.3 Modèles de points caractéristiques du visage 41
4.3.4 Apprentissage de ASM 42
4.4 Détection des émotions 43
4.4.1 Création de modèles de références des émotions 43
Normalisation des coordonnées des points caractéristiques 44
Extraction des vecteurs de données caractéristiques ("features") 46
Création des modèles de références pour chaque émotion 48
4.4.2 Création du classificateur "Dynamic Time Warping" 49

5 Tests, résultats, perspectives et conclusion 52
5.1 Tests et résultats 52
5.1.1 Détection du visage et placement des points caractéristiques 52
5.1.2 Détection des émotions 53
5.2 Perspectives 54
5.3 Conclusion 55

Bibliographie 56

Liste des tableaux

1.1 Association de deux émotions adjacentes 9
1.2 Association de deux émotions adjacentes à une émotion près 9
1.3 Association de deux émotions adjacentes à deux émotions près 10
1.4 Relation des AUs et des émotions 13

5.1 Résultat de reconnaissance des émotions et taux de confusion avec les autres émotions (données utilisées pour l'apprentissage) 53
5.2 Résultat de reconnaissance des émotions et taux de confusion avec les autres émotions (données non utilisées pour l'apprentissage) 53

Table des figures

1 Partenaires du DOMUS iii
2 Plan de la cuisine et disposition des capteurs v

1.1 Roue des émotions de Plutchik 8
1.2 Vue 3D de la roue des émotions de Plutchik 9
1.3 Anatomie Musculaire 11
1.4 Action Units de la partie supérieure du visage 11

2.1 Active Shape Model 17
2.2 Carte de choix de l'algorithme d'apprentissage 18

3.1 Description des composants principaux de l'analyseur des émotions 20
3.2 Diagramme de cas d'utilisation du détecteur d'émotion 21
3.3 Diagramme du séquence de l'ensemble du système 21
3.4 Diagramme de classes du module de la détection des émotions 22

4.1 Caractéristiques Pseudo-Haar 25
4.2 Caractéristiques Pseudo-Haar dans une fenêtre de recherche 25
4.3 Caractéristiques Pseudo-Haar pour la détection du visage 26
4.4 Image Intégrale 27
4.5 Image intégrale : somme d'une région 27
4.6 Classificateur en cascades 30
4.7 Expression faciale de la joie 31
4.8 Expression faciale de la surprise 31
4.9 Expression faciale de la colère 31
4.10 Exemple de placement de points caractéristiques 33
4.11 Numérotation des points et emplacement sur le visage 34
4.12 Symétrie et chaînage des points 34
4.13 Deux formes géométriques : le carré bleu est le carré de référence et le carré vert est le carré à normaliser. Les étapes de normalisation : (a) Translation du carré vert, (b) Mise à l'échelle du carré vert, (c) Rotation du carré vert. . 36
4.14 Exemple de nuage de points d'un visage 37
4.15 Modèle moyen d'un visage 37
4.16 Application du modèle avec ASM 39
4.17 Différentes étapes du SIFT 40

4.18 Exemple de correspondance entre deux images 41
4.19 Modèle MUCT/XM2VTS 42
4.20 Modèle Multi-PIE 42
4.21 Représentation graphique du visage neutre moyen 45
4.22 Calcul de l'angle θ et de la distance d entre les deux couples de points (p_0, p_1) et (p_1, p_2) 47
4.23 Calcul de l'angle θ et de la distance d entre les deux couples de points (p_0, p_1) et (p_2, p_3) 47
4.24 Représentation de l'intensité maximale des expressions faciales prototypes (frame 30) de gauche vers la droite : colère, joie, dégoût, peur, tristesse et surprise. 49
4.25 Correspondance euclidienne et correspondance DTW 50
4.26 Matrice de distances DTW 50

Chapitre 1

Étude comportementale et émotionnelle de l'être humain

Avant d'apprendre à parler et à communiquer avec des mots l'être humain utilise les gestes pour faire passer l'information. La communication gestuelle est innée ; tous les nouveaux-nés expriment leurs besoins en nourriture ou autre avec des gestes et des expressions faciales que les mères grâce à leurs instincts maternel peuvent comprendre.

Les études [19] du professeur **Albert Mehrabian** montrent qu'un message est généralement composé de 7% de mots, de 38% d'intonation et autres types de sons et de 55% de gestuelle. Selon le Professeur **Ray Birdwhistell** la communication se fait à 35% d'une façon verbale et 65% en non verbale.

Ces calculs sont faits en comptant la durée pendant laquelle une personne parle durant une journée et selon le Professeur **Ray Birdwhistell**, une personne parle en moyenne pendant 10 à 11 minutes par jour, et chaque phrase prend à peu près 2.5 secondes.

Il est nécessaire lors de l'étude du comportement humain de préciser que la majorité des recherches dans ce domaine indiquent que le canal verbal est essentiellement utilisé pour faire passer de l'information tandis que le canal non verbal est utilisé pour la négociation de l'attitude interpersonnelle et dans certains cas comme un substitut pour les messages verbaux.

L'étude qui suit est composée de cinq parties :
— La différence entre l'affect, l'humeur et l'émotion.
— Les expressions faciales.
— Les gestes et les mouvements du corps.
— Les signes vitaux.
— La voix.
Il est important de savoir que les gestes ne peuvent pas être interprétés seuls. Tout comme

les mots il est nécessaire de comprendre le contexte en premier lieu ainsi la compréhension du sens du geste sera plus correcte. C'est pour cela il est impératif de combiner les expressions faciales avec les mouvements du corps et la voix.

1.1 Affect, Humeur et Émotion

Ces trois termes peuvent prêter à confusion et ils sont étroitement liés. Le professeur **Timothy Judge** et le docteur **Stephen Robbins** définissent ces termes comme suit[21] :

- l'affect est un terme plus générique qui englobe des sentiments vécus où il est possible de trouver les notions d'émotions et d'humeur.
- L'émotion est par contre un sentiment intense provoqué par un événement ou une interaction avec quelqu'un, puis dirigé vers l'événement ou la personne responsable de l'apparition de l'émotion. Une émotion est habituellement accompagnée d'expressions du visage et de mouvements du corps. Elle est aussi d'une durée très courte.
- L'humeur est un sentiment généralement moins intense que l'émotion et dénué de tout stimulus contextuel. L'humeur est de nature cognitive et ne s'accompagne pas forcement de signes extérieurs distinctifs. La durée de l'humeur peut être longue et cela peut aller de quelques heures à plusieurs jours.

L'émotion et l'humeur peuvent s'influencer mutuellement c'est à dire qu'une émotion assez intense peut changer l'humeur de la personne et inversement si une personne est de mauvaise humeur elle peut facilement montrer des émotions négatives exagérées.

Une illustration des émotions correspondrait à la colère ressentie quand une autre personne se montre grossière ou vulgaire ou bien quand un événement tant inattendu survient. En revanche, l'humeur peut être une conséquence d'une émotion, par exemple quand quelqu'un critique le travail d'une autre personne, cette dernière va ressentir de la colère qui, une fois dissipée, peut se transformer en un sentiment de découragement non attribuable à un événement particulier. D'un autre côté ce sentiment de découragement va faire de telle sorte que la réaction aux autres événements devient un peu plus excessive. Cet état correspond à l'humeur.

Dans notre cas, nous nous intéressons surtout aux émotions plutôt qu'à l'humeur. Cependant l'humeur est un élément à ne pas négliger non plus. Pour cela il est nécessaire de faire une liste des émotions de base connues puis d'essayer de les classifier selon l'ordre d'intensité.

Plusieurs études ont été faites pour savoir quelles sont les émotions de base d'un être humain et la liste comporte entre 6 et 8 émotions de base qui sont :

- La peur.
- La colère.
- La joie.
- Le dégoût.

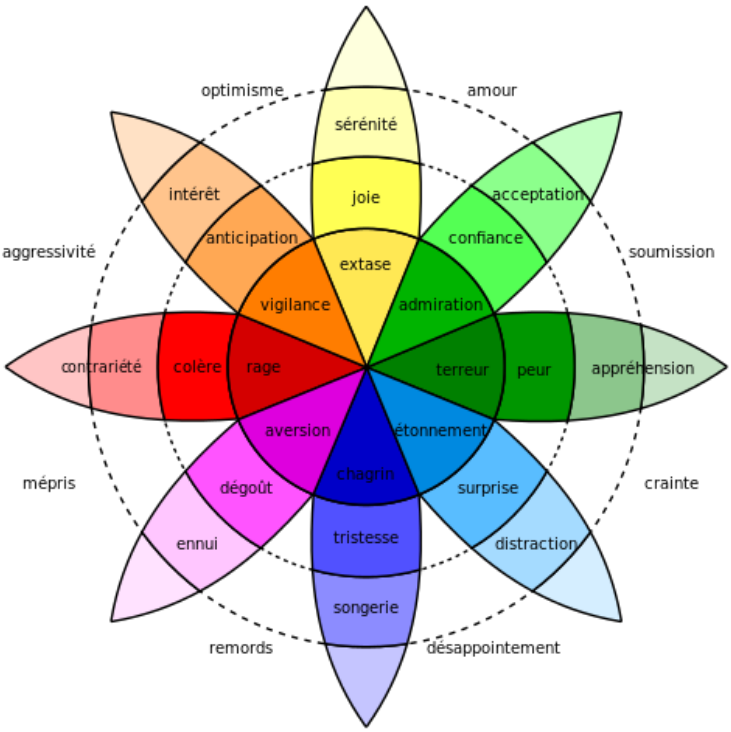

Fig. 1.1 Roue des émotions de Plutchik

— La tristesse.
— La surprise.
— L'anticipation.
— La confiance.

Les deux dernières émotions ont été rajoutées par le docteur **Robert Plutchik** qui a imaginé une roue (Fig 1.1) permettant de résumer les émotions de base et aussi de combiner plusieurs émotions[25].

Le docteur **Robert Plutchik** propose un modèle en 3D (Fig 1.2) qui montre les émotions de bases et les relations entre les émotions. Ce modèle permet même d'associer les couleurs qui sont les plus adaptées (par exemple le rouge est généralement symbole de la colère). D'un autre côté cette roue montre aussi les 8 émotions de base (colère, anticipation, joie, confiance, peur, surprise, tristesse et dégoût) et certains degrés d'une même émotion (plus nous nous rapprochons du centre plus c'est intense et plus nous nous éloignons plus c'est léger). Cette roue montre aussi l'antonyme de chaque émotion et ce en faisant la symétrie par rapport au centre de la roue, par exemple l'inverse de *"la peur"* est *"la colère"*.

Toutes les émotions de base sont associées à une couleur qui la caractérise (par exemple la colère est généralement représentée par la couleur rouge) et les émotions qui ne sont pas associées à une couleur sont des émotions combinées c'est à dire si nous prenons "*l'amour*" c'est en réalité une fusion entre "*la joie*" et "*la confiance*". De là il a été possible de définir

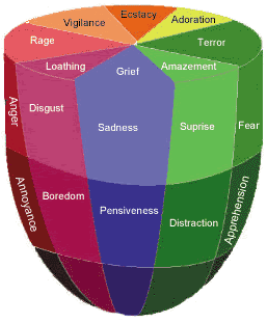

Fig. 1.2 Vue 3D de la roue des émotions de Plutchik

les dyades primaires, secondaires et tertiaires :

- Dyades primaires sont l'association de deux émotions adjacentes (Tab 1.1).
- Dyades secondaires sont l'association de deux émotions voisines à une émotion près (Tab 1.2).
- Dyades tertiaires sont l'association de deux émotions voisines à deux émotions près (Tab 1.3).

Dyades primaires	**Résultats**
Joie + Confiance	Amour
Confiance + Peur	Soumission
Peur + Surprise	Crainte
Surprise + Tristesse	Désappointement
Tristesse + Dégoût	Remords
Dégoût + Colère	Mépris
Colère + Anticipation	Agressivité
Anticipation + Joie	Optimisme

Tab. 1.1 Association de deux émotions adjacentes

Dyades secondaires	**Résultats**
Joie + Peur	Culpabilité
Confiance + Surprise	Curiosité
Peur + Tristesse	Désespoir
Surprise + Dégoût	?
Tristesse + Colère	Envie
Dégoût + Anticipation	Cynisme
Colère + Joie	Fierté
Anticipation + Confiance	Fatalisme

Tab. 1.2 Association de deux émotions adjacentes à une émotion près

1.2 Expressions faciales

Les expressions faciales sont très importantes pour pouvoir connaître l'état de la personne. Grâce à ces expressions il est possible de faire plusieurs déductions et de récupérer plusieurs informations comme :

Dyades tertiaires	Résultats
Joie + Surprise	Ravissement
Confiance + Tristesse	Sentimentalité
Peur + Dégoût	Honte
Surprise + Colère	Indignation
Tristesse + Anticipation	Pessimisme
Dégoût + Joie	Morbidité
Colère + Confiance	Domination
Anticipation + Peur	Anxiété

Tab. 1.3 Association de deux émotions adjacentes à deux émotions près

— L'état affectif que ce soit les émotions (peur, colère, joie, surprise, tristesse, dégoût) ou bien certaines humeurs.
— L'activité cognitive comme la concentration, l'ennui ou la perplexité.
— Le tempérament et la personnalité.
— La véracité et ce à travers les micro-expressions que le visage affiche d'une façon incontrôlée.

L'étude des expressions faciales se fait en se basant sur l'étude des muscles du visage. L'anatomiste **Carl-Herman Hjortsjö** était le premier à travailler sur les expressions faciales et le psychologue **Paul Ekman** s'est inspiré de ses travaux pour faire la liaison entre les émotions et les expressions faciales.

1.2.1 "*Facial Action Coding System*", "*Action Unit*" et "*Action Descriptor*"

"Facial Action Coding System" (ou "FACS") [2][8] est un système qui a été inventé par le psychologue **Paul Ekman** permettant de décrire les mouvements du visage et par la suite de déterminer l'émotion liée à ce mouvement.

"*FACS*" permet de coder n'importe quelle expression faciale possible et ce en la décomposant en une ou plusieurs Unités d'Action ("*Action Unit*" ou "*AU*").

Les "Action Units" (ou "AUs") [8] sont les plus petits mouvements faciaux visuellement discernables. Les "*AUs*" sont associées à un ou plusieurs muscles faciaux.

Les "Action Descriptors (ou "ADs") [8] décrivent les mouvements de plusieurs groupes de muscles.

L'intensité du mouvement [8] permet de rajouter une information importante sur une expression donnée. Il existe 5 niveaux qui peuvent être associés à chaque "AUs". Ces niveaux sont marqués par des lettres de A jusqu'à E qu'on rajoute après le numéro d'un "AU".

— A = Très faible.
— B = Minime.
— C = Moyen.
— D = Sévère.
— E = Maximum.

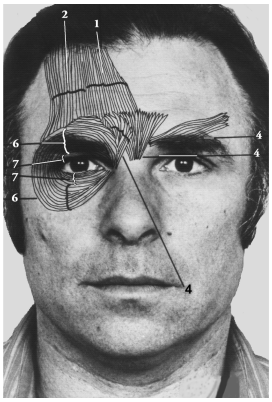

Fig. 1.3 Anatomie Musculaire

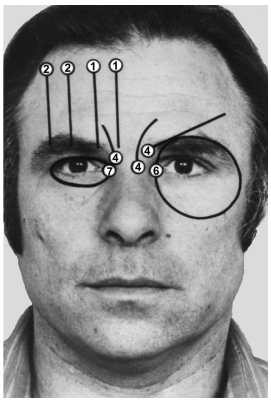

Fig. 1.4 Action Units de la partie supérieure du visage

1.2.2 Groupes des "*Action Units*"

Les "*Action Units*" sont divisées en 7 groupes [8] :

Les AUs de la partie supérieure du visage sont au nombre de 6 et décrivent le mouvement des sourcils, des paupières et des joues (la figure 1.4 illustre les zones liées). Il s'agit donc des :

— AU 1 : Remontée de la partie interne des sourcils.
— AU 2 : Remontée de la partie externe des sourcils.
— AU 4 : Abaissement et rapprochement des sourcils.
— AU 5 : Ouverture entre la paupière supérieure et les sourcils.
— AU 6 : Remontée des joues.
— AU 7 : Tension de la paupière.

Les AUs de la partie inférieure du visage et ayant un mouvement vertical sont au nombre de 8 et décrivent le mouvement vertical du nez, des lèvres, du menton et de la bouche. Il s'agit donc des :

— AU 9 : Plissement de la peau du nez vers le haut.
— AU 10 : Remontée de la partie supérieure de la lèvre.
— AU 15 : Abaissement des coins externes des lèvres.
— AU 16 : Ouverture de la lèvre inférieure.
— AU 17 : Élévation du menton.
— AU 25 : Ouverture de la bouche et séparation légère des lèvres.

— AU 26 : Ouverture de la mâchoire.
— AU 27 : Bâillement.

Les AUs de la partie inférieure du visage et ayant un mouvement horizontal :
— AU 20 : Étirement externe des lèvres.
— AU 14 : Plissement externe des lèvres.

Les AUs de la partie inférieure du visage et ayant un mouvement oblique :
— AU 11 : Ouverture du nasolabial.
— AU 12 : Étirement du coin des lèvres.
— AU 13 : Étirement et rentrée des lèvres.

Les AUs de la partie inférieure du visage et ayant un mouvement orbital :
— AU 18 : Froncement central des lèvres.
— AU 22 : Tension du cou.
— AU 23 : Tension refermante des lèvres.
— AU 24 : Lèvres pressées (pincement des lèvres).
— AU 28 : Succion interne des lèvres.

Les AUs divers : sont des AUs non classées mais qui décrivent plusieurs parties différentes de la tête par exemple la langue ou la nuque.

Les AUs du mouvement de la tête et des yeux : ces AUs décrivent les mouvements possibles de la tête (par exemple la tête tournée vers la droite AU 52), ainsi que les mouvements des yeux (par exemple les yeux vers le hauts AU 63). Ces AUs sont au nombre de quatorze.

1.2.3 Combinaison de plusieurs AUs

Il est possible de combiner plusieurs AUs pour décrire les émotions[8] le tableau 1.4 montre des exemples. À cette information il est possible de rajouter l'intensité d'une AU avec les lettres de A à E.

1.3 Autres moyens pour détecter les émotions

Hormis les expressions faciales, qui sont considérées comme étant l'axe principal pour étudier les émotions d'un être humain, il est possible d'utiliser d'autres données renvoyées par le corps et de les analyser et essayer d'en déduire l'état émotionnelle.

Émotion	Action Units
Joie	6 + 12
Tristesse	1 + 4 + 15
Surprise	1 + 2 + 5 + 26
Peur	1 + 2 + 4 + 5 + 20 + 26
Colère	4 + 5 + 7 + 23
Dégoût	9 + 15 + 16
Mépris	12 + 14

Tab. 1.4 Relation des AUs et des émotions

Il est possible d'étudier les mouvements du corps par exemple les agitations des bras pour savoir si une personne est en colère ou pas. Ou bien d'étudier la voix et l'intonation qui peut être une source supplémentaire d'informations. Et finalement il est possible d'utiliser les signes vitaux qui permettent non seulement de connaître les émotions mais aussi le niveau de fatigue et le niveau du stresse de la personne.

Tous ces moyens peuvent être combinés pour avoir un système complet qui exploite un maximum de données pour avoir une meilleure exactitude.

Ce chapitre a permis la définition dans un cadre théorique des différentes façons avec laquelle un être humain arrive à exprimer ses émotions. Nous nous focalisons plus sur les expressions faciales pour le moment. Ce qui suit est fortement basé sur cette étude théorique, et va permettre l'exploitation de ces données pour la réalisation d'un système de détection d'émotions à partir des expressions faciales.

Chapitre 2

Objectifs et méthodologies

Pour commencer il est important de fixer l'objectif à atteindre ensuite de faire une étude de l'existant d'un point de vue technologique pour arriver à se situer et savoir comment procéder. Cette partie expose les algorithmes existants et les différentes méthodes qu'il sera possible d'adopter pour atteindre l'objectif.

L'analyseur d'émotions à partir des expressions faciale fait partie d'un autre système qui est l'assistant culinaire, et qui permet aux personnes âgées et aux personnes ayant un traumatisme cranio-cérébral de préparer un repas. Pour cela l'utilisateur aura un système proposant différentes recettes de différents niveaux de difficulté. Selon l'état émotionnel de la personne l'assistant culinaire peut changer de comportement pour mieux guider l'utilisateur.

2.1 Objectif

L'objectif principal de ce travail est d'analyser les émotions de l'être humain et donner des estimations pour chaque émotion de base (joie, colère, dégoût, tristesse, peur et surprise). L'analyse des émotions se fait suivant les expressions faciales. Pour cela il est important de fixer les régions d'intérêt du visage et étudier leurs mouvements pour savoir l'état de la personne en temps réel.

D'un autre côté l'analyse des émotions ne se fait pas en se basant sur une seule image mais plutôt sur une vidéo autrement dit une séquence d'images montrant l'évolution d'une expression par rapport au temps.

Grâce au détecteur des émotions il est possible d'éviter les accidents causés par le changement d'humeur (comme la colère).

Cet objectif principal peut être divisé en 3 sous objectifs qui sont :
— Détecter le visage dans un flux vidéo.

— Placer les points caractéristiques (ou “Landmarks”).
— Trouver l’émotion en analysant le mouvement des points caractéristiques.

2.2 Méthodologies

2.2.1 Détection du visage

La détection du visage est une étape primordiale pour la reconnaissance des émotions. Il existe divers moyens pour repérer un visage dans une image ou dans un flux vidéo. Le résultat de chaque méthode peut varier selon le changement des paramètres de l’environnement, c’est à dire certaines méthodes peuvent être plus sensibles que d’autres aux changements de la luminosité, de l’arrière plan, de l’angle du visage, de la distance, etc.

Parmi les méthodes de détection de visages il y a :
— **La détection du visage en contrôlant l’arrière plan :** cette méthode consiste à supprimer l’arrière plan de l’image et de ne garder que le visage. Ceci demande que l’arrière plan soit une image prédéfinie et statique.
— **La détection du visage en se basant sur les couleurs [12][22][16] :** cette méthode exploite la texture et la couleur de la peau pour le repérage du visage. L’inconvénient principal de cette méthode, est qu’elle est sensible aux variation de la lumière.
— **La détection du visage à partir du mouvement :** Le visage d’un être humain est souvent en mouvement. L’idée derrière cette méthode est de trouver l’élément en mouvement et de calculer sa position. Par contre, il est facilement possible de confondre n’importe quel autre objet en mouvement.
— **La détection du visage dans un environnement libre :** ce type de détection est plus général, il ne se base sur aucune des méthodes précédemment décrites, mais plutôt sur de l’apprentissage. Pour cela il est nécessaire de posséder une grande base de données d’images à utiliser comme références. Cette méthode peut donner de très bons résultats selon la technique utilisée.

À noter que la reconnaissance des visages est toute une autre tâche différente de la détection des visages. La reconnaissance des visages est, par contre, une étape qui peut suivre l’étape de la détection du visage mais dans le cas du système de détection d’émotions cette étape n’est pas nécessaire. La reconnaissance de visages permet, en général, d’identifier la personne et de lui associer certains paramètres qui lui sont attribués (une utilisation très fréquente de la reconnaissance des visages est dans les systèmes de sécurité des portes où la porte ne s’ouvre que pour les personnes autorisées).

L’environnement dans lequel le système doit fonctionner est la cuisine par conséquent l’utilisateur aura à se déplacer d’une façon régulière que ce soit pour avoir accès à l’évier, la cuisinière, à l’espace du travail ou bien à d’autres parties de la cuisine. Pour cela la méthode de la détection du visage dans un environnement libre reste la méthode la plus efficace pour

ce genre de tâches.

2.2.2 Détection du visage dans un environnement libre

La détection du visage dans un environnement libre est la méthode la plus utilisée. Il existe plusieurs techniques différentes qui permettent de trouver un visage dans une image ou dans un flux vidéo, comme par exemple un apprentissage se basant sur un réseau de neurones ou bien la détection exploitant un modèle d'un visage [6] généré suite à un apprentissage ou encore l'utilisation de classificateurs faibles[27] (Weak Classifiers Cascades).

Certaines de ces techniques peuvent être plus performantes[23][18] que d'autres, surtout au niveau de la détection des différentes poses d'un visage, c'est à dire détecter un visage vue de profil, ou bien un visage avec une partie cachée par exemple avec une main, etc.

Outre le taux d'exactitude de détection de visage il y a un autre paramètre très important à prendre en considération qui est le temps d'exécution. Certaines de ces techniques peuvent être plus lente que d'autres, et ce à cause des calculs à faire et des données à traiter.

La technique utilisée pour le système de détection des émotions se base sur les classificateurs faibles en cascades[27][26] et plus précisément sur le **Haar Cascades**. Cette technique inventée par **Paul Viola** et **Michael Jones** permet d'avoir des résultats impressionnants pour la localisation des visages.

2.2.3 Détermination des points caractéristiques (ou "Landmarks")

Une fois qu'un visage est détecté dans l'image captée par la caméra il faut déterminer les zones d'intérêt qu'on appelle aussi les points caractéristiques (ou "Landmarks"). Ces points permettent d'encadrer les régions telles que les yeux, la bouche, le nez, les sourcils, etc.

Pour pouvoir déterminer les "landmarks", qui sont les points caractéristiques du visage, il existe plusieurs algorithmes tel que :

- *Active Shape Model (ASM)* [5][4] qui est un modèle statistique permettant de placer les points caractéristiques sur un objet.
- *Active Appearance Model (AAM)* [3] est une version similaire à ASM mais qui utilise d'autres informations dans l'image.
- *Elastic Bunch Graph* [28] est un algorithme qui se base sur un graphe pour la reconnaissance des objets.

Ces "landmarks" sont liés aux "Actions Units" définis par **Paul Ekman**, autrement dit ces points permettent de définir l'état de chaque muscle du visage et par la suite il est possible de suivre leurs mouvements et de déduire l'émotion exprimée à partir des expressions faciales.

Dans le cadre du système de détection des émotions l'algorithme utilisé est une variante [17] de l'algorithme ASM développée par **Stephen Milborrow** et **Fred Nicolls**. Cette

variante permet d'avoir de meilleurs résultats par rapport au AAM ou même par rapport à la version originale de ASM.

L'algorithme ASM permet de générer un modèle de la forme du visage (dans le cas de détection du visage), et ce modèle est alors adapté automatiquement d'une façon itérative grâce à d'autres algorithmes d'adaptation pour mieux correspondre au visage sur une image.

Il est possible d'utiliser ASM pour la détection de n'importe quels types d'objets. Pour cela il est nécessaire d'avoir un grand ensemble d'images de l'objet à détecter, de fixer un nombre de points et de positionner manuellement ces points sur les images à utiliser pour l'apprentissage (Fig. 2.1).

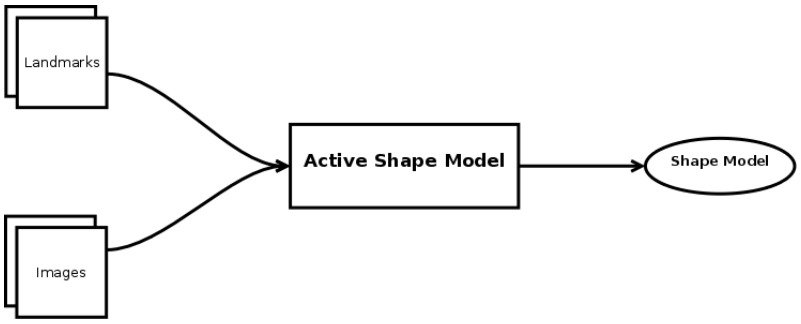

Fig. 2.1 Active Shape Model

2.2.4 Détection des émotions

Cette étape est la dernière étape de l'analyseur des émotions et elle est divisée en deux sous étapes :

1. Normalisation des données et création de modèles de références et ce en extrayant les caractéristiques et en lançant un apprentissage.
2. Création du classificateur permettant de classifier les nouvelles données en comparant par rapport aux données de référence.

La première sous étape est une étape qui se base sur les coordonnées des points caractéristiques définies précédemment. Ces coordonnées représentent le point de départ qu'il faut exploiter pour créer un modèle de référence par émotion de base.

Par la suite la classification peut être faite de plusieurs façons [24] par exemple il est possible d'utiliser un réseau de neurones à plusieurs couches qu'il faut entraîner au début. Mais il est possible d'utiliser d'autres techniques moins compliquées que le réseau de neurones, telle que SVM (Support Vector Machine) ou bien K-Neighbors, etc. Ce choix dépend essentiellement de la quantité des données dont on dispose et de leur nature aussi. La figure [1] 2.2 explique comment ce choix peut être fait.

1. $http://scikit-learn.org/stable/tutorial/machine_learning_map$

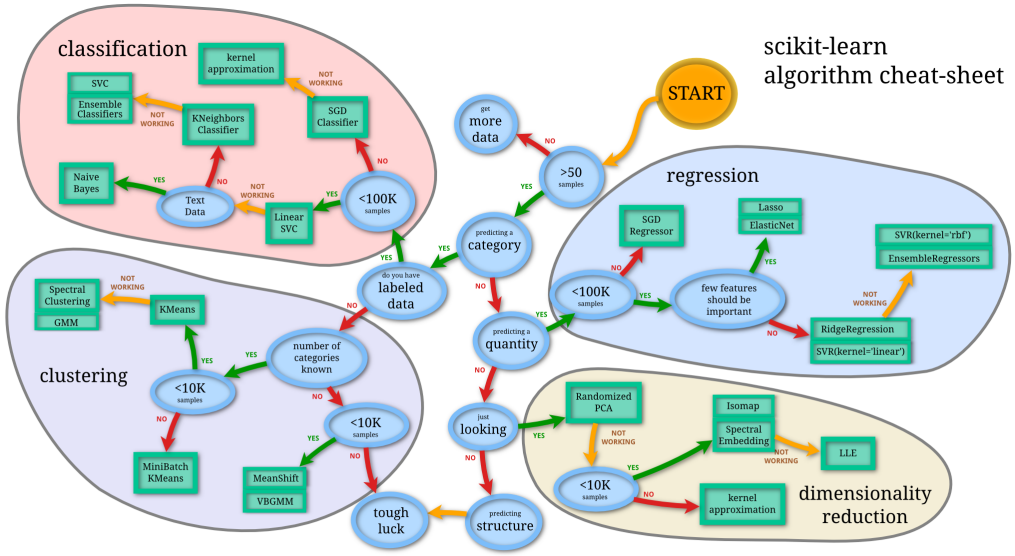

Fig. 2.2 Carte de choix de l'algorithme d'apprentissage

L'algorithme utilisé pour la classification dans le travail qui suit est le Dynamic Time Warping (DTW). Cet algorithme permet de calculer la distance entre deux séquences d'images de tailles différentes.

Plusieurs méthodes existent pour arriver à détecter les émotions à partir des expressions faciales. Comme expliqué précédemment, ceci se fait en trois étapes (détection du visage, placement des points caractéristiques et détection de l'émotion). La méthode proposée se base sur une version récente et modifiée de l'algorithme ASM, ainsi qu'une implémentation personnalisée de l'algorithme DTW.

Chapitre 3

Spécification du système de détection d'émotions

Avant de commencer l'étape de l'implémentation, nous avons besoin d'énumérer les besoins fonctionnels et les besoins non fonctionnels du système. Par la suite il est important de modéliser l'ensemble des parties du système tout en les décrivant. Tout ceci permet de mieux faire le choix technologique qui permettra la réalisation concrète de chacune des parties.

3.1 Besoins fonctionnels et non fonctionnels

3.1.1 Besoins fonctionnels

Le système de détection d'émotions doit assurer les besoins fonctionnels suivants :

- Exploiter toutes les caméras présentes dans la cuisine et ce pour pouvoir couvrir toute la région.
- Détecter le visage de la personne sans tenir compte de la position dans la cuisine.
- Arriver à placer les points caractéristiques sur le visage et suivre en temps réel le mouvement de ces points.
- Détecter les émotions à partir du mouvement des points caractéristiques.
- Donner une estimation de l'émotion exprimée.
- Formater les données relatives à l'émotion détectée pour pouvoir les renvoyer à l'assistant culinaire.

3.1.2 Besoins non fonctionnels

Les besoins non fonctionnels sont :

— Garantir un fonctionnement en temps réel.
— Permettre l'évolutivité du système en adaptant une architecture modulaire (chacune des trois parties de l'analyseur des émotions peut être modifiée et améliorée d'une façon indépendante des autres).
— Fournir un système multiplate-formes.
— Permettre l'utilisation de n'importe quel type de caméra vidéo (par exemple : webcam, kinect, camera de smartphone, etc).

3.2 Spécification du système

La figure 3.1 décrit d'une façon générale les composants principaux de l'analyseur qui se décompose en 3 couches (la couche physique, la couche du traitement et la couche de la sortie).

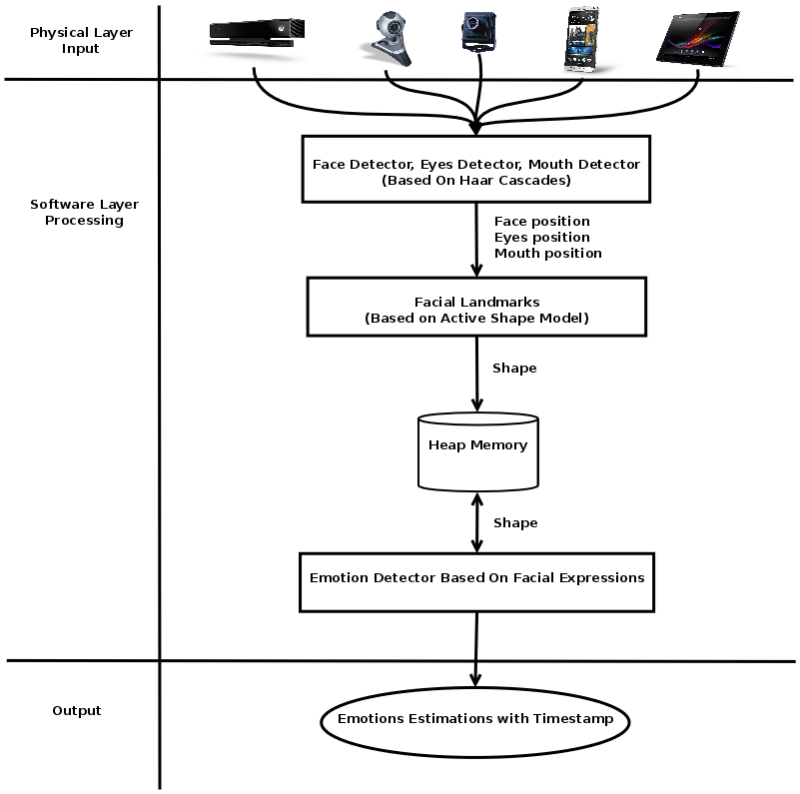

Fig. 3.1 Description des composants principaux de l'analyseur des émotions

3.2.1 Spécification de la couche matérielle

La détection des émotions dans ce travail se fait en se basant sur les expressions faciales, pour cela il est nécessaire d'utiliser des caméras pour avoir des images de l'environnement

et de la personne qui est présente dans la cuisine.

Les caractéristiques techniques de la caméra sont importantes pour garantir un nombre d'images suffisant permettant de discerner les moindre mouvements des muscles faciaux. La qualité de l'image est aussi importante, c'est à dire la résolution qui est exprimée en pixels, plus la résolution est grande plus la qualité sera meilleure et donc ceci permettra de pouvoir mieux détecter le visage et par la suite mieux placer les points caractéristiques. Un autre paramètre intéressant et qu'il ne faut pas négliger, qui est le champ visuel, ceci permet de designer la zone que la caméra est capable de couvrir et grâce à ce paramètre il sera possible d'optimiser le nombre de caméras dans la cuisine.

3.2.2 Spécification de la couche logicielle

Le diagramme ci-dessous (fig. 3.2) décrit les fonctionnalités du système de détection des émotions :

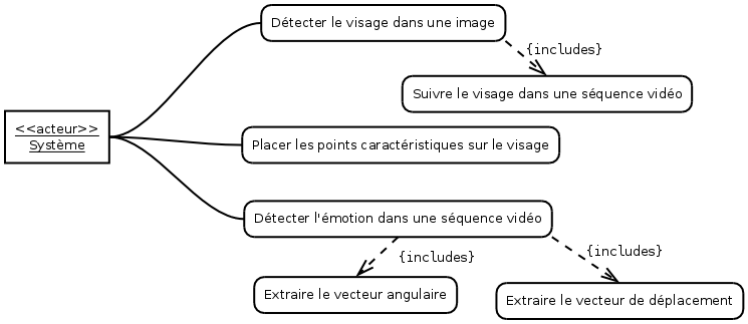

Fig. 3.2 Diagramme de cas d'utilisation du détecteur d'émotion

Le diagramme de séquence (fig. 3.3) ci-dessous décrit le déroulement du processus :

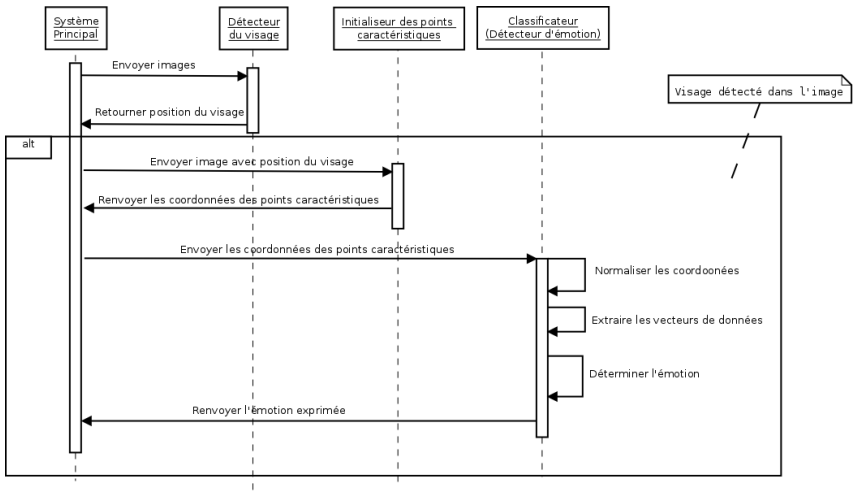

Fig. 3.3 Diagramme du séquence de l'ensemble du système

Le diagramme de classes (fig. 3.4) suivant décrit le module de la détection des émotions :

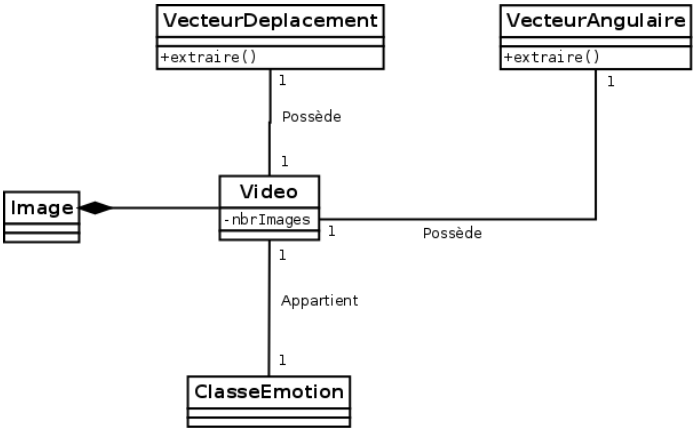

Fig. 3.4 Diagramme de classes du module de la détection des émotions

3.3 Choix technologique

3.3.1 Langages de programmation

Le choix du langage de programmation est important pour pouvoir réaliser le projet. Le langage de programmation choisi pour implémenter le système est le langage **C++**. Ce langage permet de garantir une bonne vitesse d'exécution et par conséquent arriver à avoir un système qui s'exécute en temps réel. La détection du visage ainsi que le placement des points caractéristiques ont été implémentés en **C++**.

Outre le **C++** le langage de programmation **Python** a été choisi pour permettre de faire certaines tâches telles que la normalisation des données, la création des émotions modèles ou même pour faire des tests en statiques pour la classification. L'avantage du langage de programmation **Python** est qu'il intègre plusieurs librairies permettant l'écriture de code de traitement rapidement.

3.3.2 Librairies

OpenCV

OpenCV [1] est une bibliothèque graphique libre qui offrent plusieurs outils pour le traitement d'images en temps réel. Cette bibliothèque intègre plusieurs classes permettant :

— L'accès à la caméra pour la capture des images (ou même une vidéo).

1. Lien vers la librairie OpenCV : http ://opencv.org/

— Le traitement des images pour effectuer une égalisation d'histogramme, lissage, filtrage, etc.
— La détection des objets et aussi les visages grâce à la technique Haar Cascades.
— L'apprentissage avec AdaBoost, réseau de neurones, K-means, etc.

Cette librairie a été utilisée pour permettre de réaliser les deux premières parties du projet (détection du visage et placement des points caractéristiques).

STASM

Cette bibliothèque [2] est une implémentation améliorée de l'algorithme **ASM**. Elle offre la possibilité de détecter les caractéristiques sur un visage, autrement dit de placer les points caractéristiques sur un visage.

Threading Building Blocks

Threading Building Blocks [3] (ou **TBB**) est une librairie permettant de bénéficier du multithreading de Intel. Grâce à cette librairie il sera possible d'avoir un traitement plus rapide et d'assurer un système fonctionnant en temps réel.

Ainsi nous avons pu fixer l'ensemble des technologies qui permettront de faciliter la réalisation du détecteur des émotions à partir des expressions faciales. Étant donné que ce système est décomposé en trois parties, il est donc possible d'implémenter chaque partie indépendamment des autres. Cette architecture modulaire, facilite aussi de tester différentes solutions pour chaque partie.

2. Lien vers la librairie STASM http ://www.milbo.users.sonic.net/stasm/
3. Lien vers la librairie TBB : https ://www.threadingbuildingblocks.org/

Chapitre 4

Implémentation du détecteur d'émotions

Dans ce chapitre nous expliquons l'implémentation de chacune des trois parties du système, et ce en présentant les algorithmes implémentés ainsi que les données d'entrée et les données de sortie de chaque partie. À chaque partie du projet, il y a eu un apprentissage automatique à faire. Cet apprentissage permet de générer des modèles à utiliser pour la réalisation du détecteur des émotions.

4.1 Détection du visage dans une image

La première étape, à la réception d'une image, est de trouver le visage. Nous utilisons pour cela l'algorithme **Haar Cascades** (implémenté dans **OpenCV**) pour renvoyer la position du visage dans l'image.

4.1.1 Haar Cascades

La méthode de **Viola** & **Jones** est une technique différente de ce qui a été répertorié dans la littérature. Elle ne se base pas sur des paramètres tels que la couleur ou la texture mais plutôt sur les caractéristiques pseudo-Haar qui doivent leur nom aux ondelettes de Haar étant donné la ressemblance.

Caractéristiques Pseudo-Haar

Ces caractéristiques pseudo-Haar sont en réalité des images (Fig 4.1) utilisées pour la reconnaissance des objets, et plus précisément pour la reconnaissance de régions dans une

image. La Figure 4.1 montre quelques unes des ces caractéristiques qui permettent de détecter par exemple un bord ou une ligne.

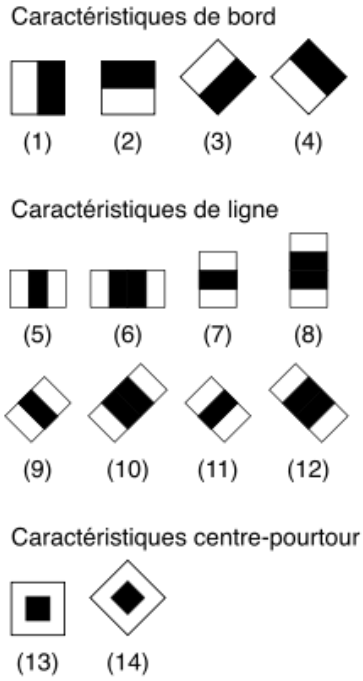

Fig. 4.1 Caractéristiques Pseudo-Haar

Le Haar Cascades est en réalité une technique automatique d'apprentissage qui se base sur un système de fonctions cascades. Pour cela il est nécessaire d'avoir un grand nombre d'images positives et un grand nombre d'images négatives pour pouvoir faire l'apprentissage (dans la première phase du détecteur des émotions les images positives sont des images avec des visages de personnes et les images négatives des images qui ne contiennent pas de visages).

Pour pouvoir lancer l'apprentissage il faut tout d'abord définir la taille de la zone de recherche, qui est aussi appelé "fenêtre". La taille de départ par défaut de cette fenêtre (comme défini par **Viola** & **Jones** dans [27][26]) est de 24x24 pixels. La figure 4.2 montre quatre fenêtres de recherche avec chacune une caractéristique différente.

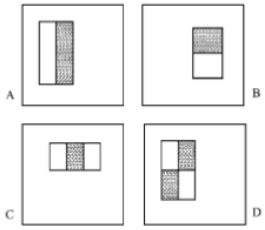

Fig. 4.2 Caractéristiques Pseudo-Haar dans une fenêtre de recherche

Une fois les caractéristiques Pseudo-Haar et la taille de la fenêtre de recherche fixées,

l'étape qui suit est de parcourir toute l'image par bloc et d'appliquer les caractéristiques Pseudo-Haar. Ceci consiste à faire la sommes des pixels dans la région sombre, ensuite la somme des pixels dans la région claire et finalement d'enregistrer la différence entre les deux régions.

Pour mieux expliquer cela, le visage de l'être humain contient des zones plus sombres que d'autres, par exemple la région des yeux est généralement plus foncée que le nez ou les joues. La figure 4.3 montre deux caractéristiques intéressantes. La première permet de détecter que les joues et le nez sont plus clairs que les yeux et la deuxième permet de détecter que la racine du nez est plus claire que les yeux.

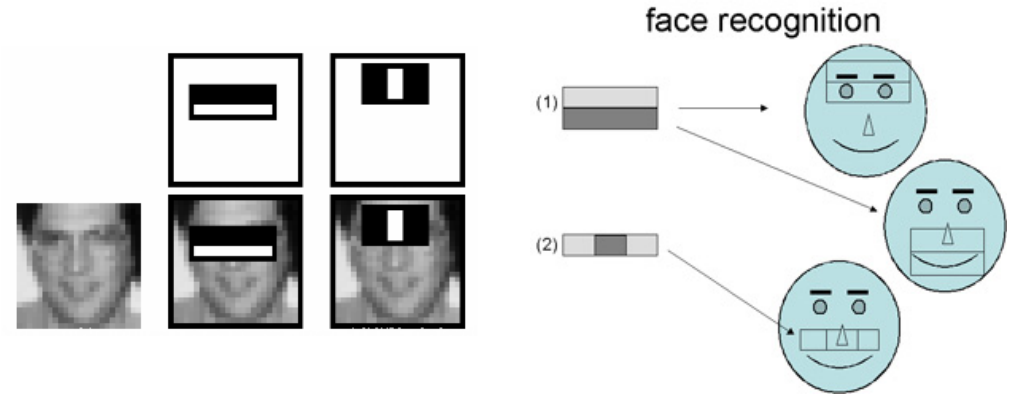

Fig. 4.3 Caractéristiques Pseudo-Haar pour la détection du visage

Le problème rencontré est le nombre de caractéristiques Pseudo-Haar qu'on peut générer, sachant qu'elles peuvent avoir n'importe quelle taille qui ne dépasse pas la taille de la fenêtre de recherche et n'importe quelle orientation. Pour une fenêtre de 24x24 pixels il est possible de générer plus de 180 000 caractéristiques. Par conséquent le calcul peut être lourd et très lent et donc la détection d'un visage devient une tâche coûteuse.

Pour optimiser le calcul, une technique a été mise en place permettant de faire les opérations d'une façon extrêmement rapide. Au lieu d'utiliser l'image d'origine pour appliquer les caractéristiques Pseudo-Haar, une représentation de l'image d'origine sera générée. Cette représentation est appelée **"Image Intégrale"**.

Image Intégrale

Cette image est définie tel que le pixel à la position (x, y) de l'image intégrale est égal à la somme des pixels au dessus et ceux à gauche :

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{4.1}$$

Avec i est l'image d'origine et ii est l'image intégrale. Il est possible de générer l'image

intégrale en une seule itération et ce en utilisant les formules suivantes :

$$s(x, y) = s(x, y - 1) + i(x, y) \tag{4.2}$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \tag{4.3}$$

Où *s(x, y)* est la somme cumulative des lignes et *s(x, -1) = 0*, et *ii(-1, y) = 0*. De là il est obligatoire de rajouter une colonne et une ligne qui seront égales à 0 (Fig 4.4).

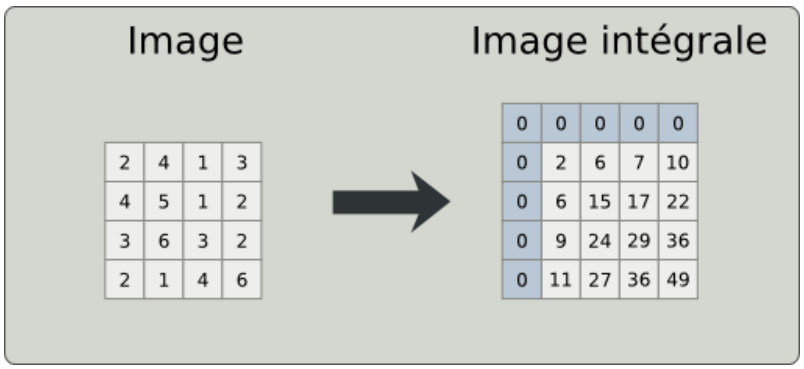

Fig. 4.4 Image Intégrale

Par la suite pour pouvoir avoir la somme des pixels dans une région il suffit juste de faire une seule opération. La figure 4.5 montre un exemple pour avoir la somme d'une région.

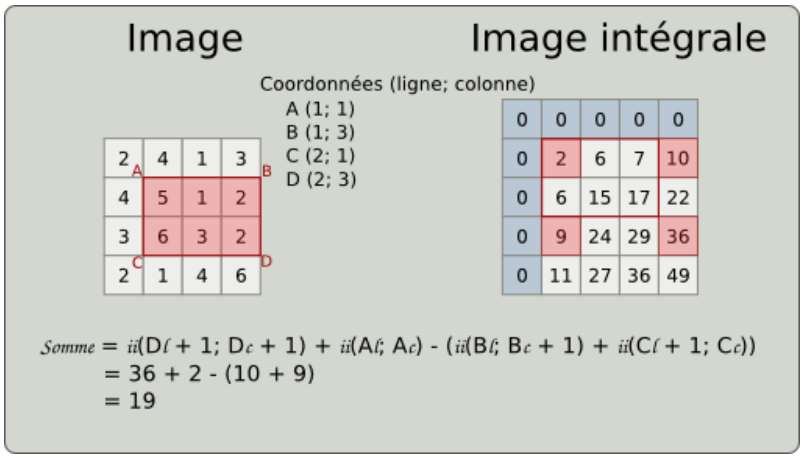

Fig. 4.5 Image intégrale : somme d'une région

Grâce à l'image intégrale il est possible d'accélérer le calcul et de faire moins d'opérations et de cette façon il est possible d'appliquer le plus de caractéristiques. La question qui se pose : est-ce utile d'utiliser toutes les caractéristiques Pseudo-Haar disponibles ?

Classificateur fort

Pour trouver les caractéristiques Pseudo-Haar les plus significatives pour la détection d'un visage, il faut utiliser une technique de classification, et dans ce cadre la technique de "Boosting" a été utilisée.

Le "Boosting" est en réalité un apprentissage automatique supervisé. Ceci permet d'avoir un apprentissage puissant en combinant les performances de plusieurs classificateurs forts, et ce pour avoir un système de classification performant et rapide.

Un classificateur fort est un ensemble de classificateurs faibles. Un classificateur faible peut être considéré comme étant une fonction h qui est appliquée sur un vecteur de données $\vec{x}$ et qui renvoie une valeur ± 1 (dans l'intervalle [-1, 1]). Par exemple, un classificateur faible peut être une fonction qui calcule l'entropie.

Le classificateur fort est en réalité la somme de plusieurs classificateurs faibles en leurs associant un poids. La formule suivante décrit le fonctionnement d'un classificateur fort :

$$y(\vec{x}) = sign(\sum_{j} w_j h_j(\vec{x})) \tag{4.4}$$

Où y est appelé classificateur fort, w_j est un réel qui représente le poids associé au classificateur faible h_j et $\vec{x}$ est le vecteur des données.

L'apprentissage lancé permet de déterminer la meilleure valeur de chaque poids à appliquer à chaque fonction h pour avoir le meilleur taux de reconnaissance qui permet de couvrir la globalité des données positives. Bien entendu l'apprentissage nécessite une préparation des données et une division de ces données en deux groupes manuellement (données positives contenant l'information à détecter et données négatives ne contenant pas l'information à détecter).

Dans le cas du détecteur d'émotions, l'algorithme de classification utilisée est **AdaBoost** qui est une variante très utilisée du "Boosting". Avec **AdaBoost** il est possible de sélectionner les caractéristiques Pseudo-Haar les mieux adaptées et qui permettent d'avoir un meilleur résultat et il est aussi possible de lancer un apprentissage pour les classificateurs. L'idée derrière cette technique est que chaque classificateur faible trouve la caractéristique (parmi les 180 000 caractéristiques Pseudo-Haar) qui permet d'avoir un meilleur taux de reconnaissance.

L'algorithme 1 (comme décrit dans [26]) explique comment il est possible d'extraire la caractéristique la mieux adaptée.

Cet algorithme permet d'avoir de très bons résultats et avec **AdaBoost** il est possible de réduire la liste des caractéristiques Pseudo-Haar de plus de 180 000 éléments à quelques centaines d'éléments, ce qui permet de réduire le temps d'exécution.

Algorithm 1: Algorithme AdaBoost pour la sélection d'une caractéristique des 180 000 caractéristiques Pseudo-Haar

Soit un ensemble d'images $(x_1, y_1), ..., (x_n, y_n)$, avec x_i l'image et $y_i = 0$ si c'est une image négative (ne contenant pas de visage) et 1 si c'est une image positive (contenant un visage)

Initialisation des variables poids $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ pour $y_i = 0, 1$ respectivement, où m est le nombre total d'images négatives et l le nombre total d'images positives.

for *t = 1, ..., T* **do**

1. Normalisation des poids :
$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}} \tag{4.5}$$
ainsi w_t est une loi de probabilité.
2. Pour chaque caractéristique j, lancer un apprentissage d'un classificateur h_j. Le taux d'erreur est calculé par rapport à w_t, $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
3. Choisir le classificateur, h_t, qui a le taux d'erreur ϵ_t le plus faible.
4. Mettre à jour les poids :
$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i} \tag{4.6}$$
Où $e_i = 0$ si l'image x_i est classifiée correctement sinon $e_i = 1$ et $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$

Le classificateur fort est construit en faisant la somme des classificateurs faibles de la façon suivante :

$$h(x) = \begin{cases} 1 & Si \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & Sinon \end{cases} \tag{4.7}$$

Où $\alpha_t = \log \frac{1}{\beta_t}$

Cependant il est possible d'améliorer encore plus le système en utilisant la technique de **classificateur en cascade**.

Classificateur en cascades

Le système en cascades permet de réduire considérablement le temps de traitement et ce en mettant en place plusieurs étages de classificateurs forts. Étant donné qu'une fenêtre de traitement a été préalablement fixée l'idée est d'ignorer les fenêtres qui ne contiennent pas la donnée recherchée et de se focaliser sur les régions qui peuvent potentiellement contenir cette donnée.

Pour cela chaque région de l'image passe par plusieurs étages et si jamais la détection échoue à un des étages, alors cette région n'est plus traitée et elle est ignorée. La figure 4.6 illustre le système de cascades.

Les étages dans le classificateur en cascades sont créer par un apprentissage en utilisant **AdaBoost**, ensuite il faut fixer un seuil par étage pour minimiser les faux négatifs (c'est à

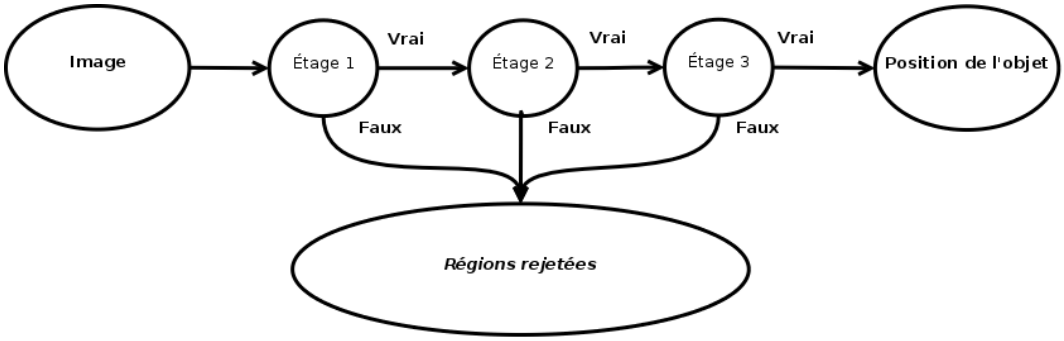

Fig. 4.6 Classificateur en cascades

dire les images qui contiennent la donnée recherchée mais que le système ne détecte pas).

L'algorithme de **Viola & Jones** [26] décrit un classificateur en cascades qui possède 38 étages et avec en tout 6000 caractéristiques Pseudo-Haar.

Implémentation

L'implémentation de l'algorithme **Haar-Cascades** a été faite avec la librairie **OpenCV** et avec le langage de programmation **C++**. La librairie **OpenCV** fournit non seulement une classe permettant de mettre en place un système de **Haar-Cascades** mais aussi des outils permettant de lancer un apprentissage du classificateur, et par la suite générer un fichier XML contenant le résultat de l'apprentissage. Ce fichier XML est utilisé comme base de référence pour la détection des visages.

La structure du fichier XML décrit l'ensemble des étages du classificateur, les caractéristiques utilisées et le seuil de chaque étage.

Le système de détection des émotions utilise le fichier XML préparé par la communauté de **OpenCV** qui permet de détecter les visages des personnes (vue frontale). L'algorithme suivant explique la détection du visage à partir d'une image :

Algorithm 2: Algorithme de détection de visage à partir d'une image

1. Charger le fichier XML "$haarcascade\_frontalface\_default.xml$" qui contient l'apprentissage fait au préalable.
2. Lire l'image.
3. Convertir l'image au niveau de gris.
4. Égalisation de l'histogramme de l'image.
5. Appeler la méthode "$CascadeClassifier :: detectMultiScale$" pour retourner la position du visage dans l'image au niveau de gris.
6. Vérifier le retour de la méthode appelée pour savoir si le classificateur en cascade a trouvé le visage ou pas.

4.2 Suivi du visage dans un flux vidéo

Un flux vidéo étant constitué d'un ensemble d'images donc il suffit d'analyser chaque image de la vidéo. Ceci peut poser un problème étant donné le nombre des images disponibles par secondes, c'est à dire pour avoir une vidéo fluide et non saccadée il faut avoir au minimum 30 images par seconde. Par conséquent le traitement de 30 images par seconde peut être coûteux en terme de ressources. Pour cela il est important d'optimiser le code et d'utiliser des techniques permettant de faire du parallélisme.

D'un autre côté en analysant les images [15] fournies par l'Université de Pittsburgh, pour les études des émotions à partir des expressions faciales, il est clair que la durée pour exprimer une émotion peut varier d'une personne à une autre. La durée minimale pour discerner une expression facile est de 0.16 seconde c'est à dire 5 images. Et la durée maximale est de 2.4 secondes, ce qui fait 72 images.

Fig. 4.7 Expression faciale de la joie

Fig. 4.8 Expression faciale de la surprise

Fig. 4.9 Expression faciale de la colère

Pour cela il est important de lancer la détection du visage sur toutes les images reçues par la caméra et par la suite la position du visage sera renvoyée à l'algorithme de positionnement des points caractéristiques qui va déterminer les coordonnées de chaque point.

Implémentation

L'algorithme du suivi du visage est quasiment le même que l'algorithme 2 sauf qu'au lieu de lire l'image d'un périphérique de stockage, la source des images devient la caméra. L'algorithme suivant explique les changements apportés pour pouvoir faire le suivi du visage :

Algorithm 3: Algorithme de suivi de visage à partir d'une image

1. Charger le fichier XML "$haarcascade\_frontalface\_default.xml''$ qui contient l'apprentissage fait au préalable.
2. Vérifier que la caméra n'est pas occupée par un autre système.
3. **repeat**
   - Lire l'image de la caméra.
   - Convertir l'image au niveau de gris.
   - Égalisation de l'histogramme de l'image.
   - Appeler la méthode "$CascadeClassifier :: detectMultiScale''$ pour retourner la position du visage dans l'image au niveau de gris.
   - Vérifier le retour de la méthode appelée pour savoir si le classificateur en cascade a trouvé le visage ou pas.

   until *jusqu'à l'arrêt du système*;

4.3 Détermination des points caractéristiques (ou "Landmarks")

Si un visage a été détecté dans l'image reçue, nous passons à la deuxième étape qui consiste à placer les points caractéristiques. Ces points permettent de suivre le mouvement de chacune des parties du visage (yeux, bouche, sourcilles et nez). L'algorithme **Active Shape Model** est utilisé pour faire le placement des points.

4.3.1 Description du fonctionnement de l'algorithme ASM

L'algorithme ASM (Active Shape Model) est un algorithme statistique qui permet de générer un modèle de points à partir d'un ensemble de données d'apprentissage et ce à fin de pouvoir placer ces points caractéristiques (appelés aussi "Landmarks") sur un objet. Dans le cas de détection d'émotions, les points sont très importants et doivent être placés d'une façon permettant de calculer les estimations de chaque émotion.

Ces points indiquent la position de chaque élément du visage par exemple les sourcils, les lèvres de la bouche, les yeux, le nez et le contour du visage. La figure 4.10 montre le placement de ces points sur un visage.

Il est important de bien placer ces points parce que le mouvement de ces points doit être suivi et par la suite une analyse sera faite pour en déduire l'émotion à partir d'une expression faciale. Pour cela il est nécessaire de préparer un ensemble assez grand d'images, puis de placer dessus les points caractéristiques et finalement ces données seront utilisées pour pouvoir lancer un apprentissage et générer un modèle utilisable et adaptable.

L'idée générale de l'algorithme ASM est de calculer un modèle moyen à partir des points caractéristiques des visages utilisés pour l'apprentissage. Et il faut aussi déterminer le comportement des points, c'est à dire les variations des points d'un visage à un autre. Pour cela,

Fig. 4.10 Exemple de placement de points caractéristiques

il est important de préparer les données d'apprentissage avant de commencer.

Préparation des données d'apprentissage

Les données à préparer consistent essentiellement en un ensemble de points. Pour cela il est important de définir le nombre de points à détecter sur un visage et par la suite de définir les relations entre eux. Dans le cas du détecteur des émotions le nombre de points à été fixé à 68 points et ce pour deux raisons :

— Les 68 points permettent de couvrir les régions les plus importantes du visage pour pouvoir détecter les émotions. C'est à dire avec ces points là il est possible de suivre la bouche, les yeux, les sourcils et le nez.

— La base de données des émotions utilisée [15] pour l'apprentissage ne définit que 68 points dans chaque séquence d'émotion.

Ensuite il faut spécifier les relations entre ces points c'est à dire définir le partenaire (ou le point symétrique), son précédent, son suivant et son poids. Pour cela une structure de données est définie comme suit :

```
1          struct LANDMARK_INFO {
2            int partner; // L'id du point symetrique.
3            int prev, next; // L'id du point precedent et l'id du point suivant
4            double weight; // Le poids
5            unsigned bits; // Information supplementaire
6          };
```

Le champ des informations supplémentaires permet de rajouter des données en plus qui peuvent être utilisées, par exemple il est possible de dire que tel point peut renseigner

l'emplacement de la barbe ou de la moustache.

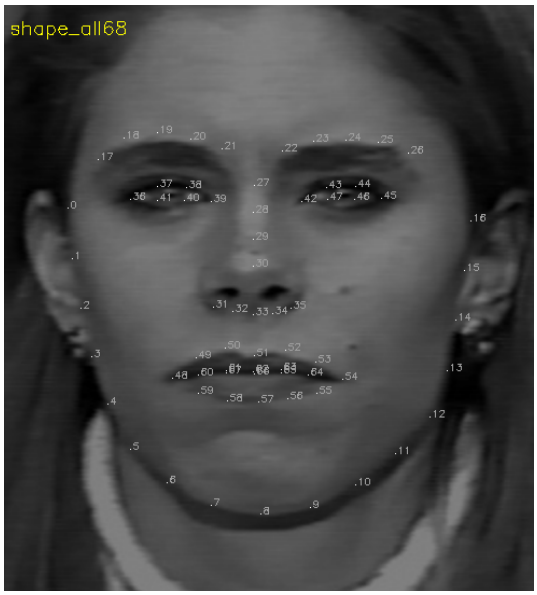

Fig. 4.11 Numérotation des points et emplacement sur le visage

Fig. 4.12 Symétrie et chaînage des points

La figure 4.11 montre l'emplacement de chaque point ainsi que son identifiant et la figure 4.12 montre le point symétrique de chaque point et son précédent et son suivant. Ceci est très important parce que par la suite le modèle moyen généré sera adapté automatiquement et chaque déplacement d'un des points entraînera un déplacement de tous les autres points.

Normalisation des données d'apprentissage

Pour pouvoir calculer le modèle moyen il faut tout d'abord aligner l'ensemble de données d'apprentissage. C'est à dire faire en sorte que les points caractéristiques de chaque visage

soient normalisés sur un repère et ce pour pouvoir comparer chaque point d'un visage avec son homologue d'un autre visage. Cette normalisation est faite en 3 étapes :

1. Une mise à l'échelle.
2. Une rotation.
3. Une translation.

Il existe plusieurs méthodes possibles pour effectuer cette normalisation :

— La méthode décrite dans "$1 Gesture" [29] qui exploite une technique très simple, où la rotation est faite par rapport au premier point de la liste des points et la mise à l'échelle est fixée au départ d'une façon statique.

— La méthode décrite dans "Protractor" [14] qui se base essentiellement sur des fonctions trigonométriques pour trouver le meilleure angle de rotation. Cette méthode, tout comme la méthode précédente, est utilisée pour la reconnaissance des gestes sur des périphériques tactiles.

— L'analyse procustéenne [11], dont l'algorithme ASM s'inspire, permet de déformer un objet pour le rendre le plus semblable possible à un autre objet. Cette technique permet de distinguer les différences entre les deux objets.

Les points de la première image sont considérés comme étant une référence et tous les autres points des autres images seront alignés par rapport à cette image là. Chaque image est représentée de la façon suivante :

$$x_i = (x_{i0}, y_{i0}, x_{i1}, y_{i1}, ..., x_{in-1}, y_{in-1})^T \tag{4.8}$$

Pour aligner deux modèles de points x_1 et x_2, il faut déterminer l'angle de rotation θ, la valeur de la mise à l'échelle s et la translation (t_x, t_y). Et ce en faisant correspondre x_2 à $M(x_2) + t$ de telle sorte que la somme suivante soit minimisée :

$$E = (x_1 - M(s,\theta)[x_2] - t)^T W (x_1 - M(s,\theta)[x_2] - t) \tag{4.9}$$

Où

$$M(s,\theta) = \begin{bmatrix} x_{jk} \\ y_{jk} \end{bmatrix} = \begin{pmatrix} (s\cos\theta)x_{jk} - (s\sin\theta)y_{jk} \\ (s\sin\theta)x_{jk} + (s\cos\theta)y_{jk} \end{pmatrix}, \tag{4.10}$$

$$t_j = (t_{xj}, t_{yj}, ..., t_{xj}, t_{yj})^T \tag{4.11}$$

Et W une matrice diagonale de poids pour chaque point. L'explication en détails de cette méthode peut être trouvée dans [4]. La figure 4.13 montre les étapes de normalisation de chaque ensemble de points. L'algorithme 4 décrit la normalisation et la création d'un modèle moyen.

Le résultat de cette normalisation est un ensemble de points normalisés et aussi un modèle moyen de point (figure 4.15). Ces données là seront utilisées pour la création d'un modèle qui s'adapte automatiquement.

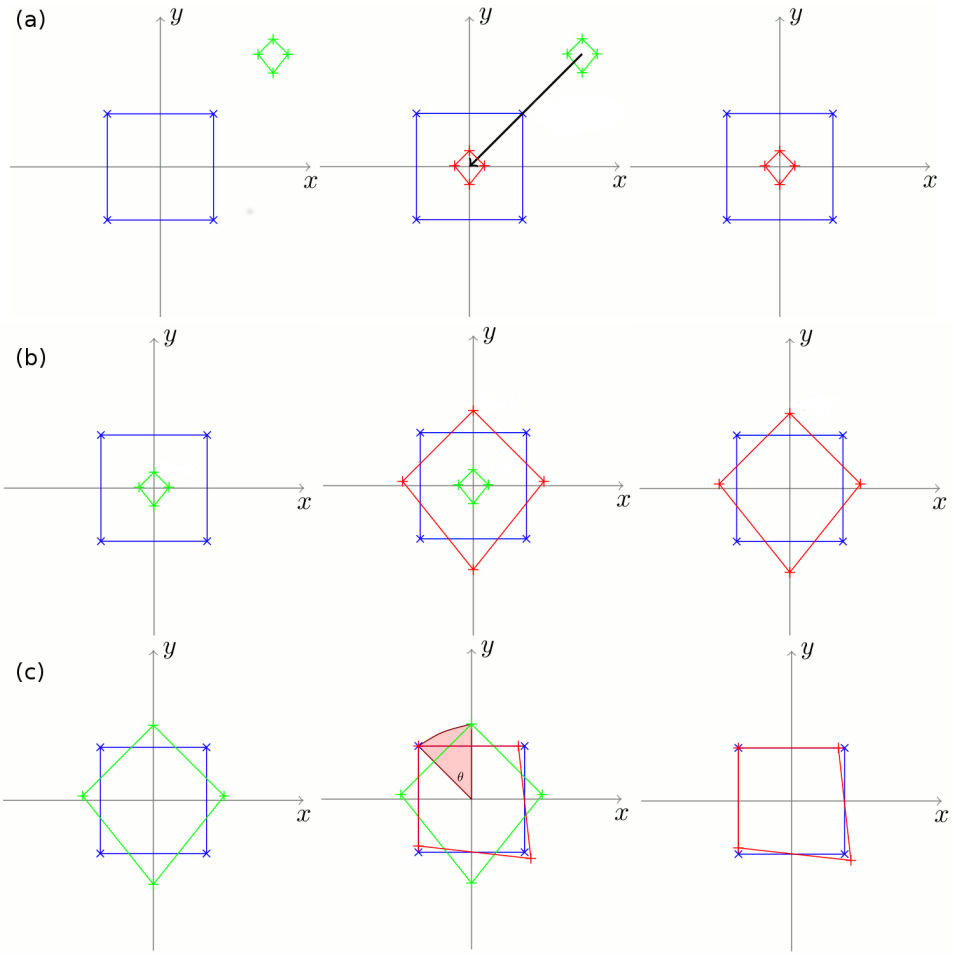

Fig. 4.13 Deux formes géométriques : le carré bleu est le carré de référence et le carré vert est le carré à normaliser. Les étapes de normalisation : (a) Translation du carré vert, (b) Mise à l'échelle du carré vert, (c) Rotation du carré vert.

Calcul du modèle statistique

Cette étape permet de calculer le modèle qui sera utilisé par la suite pour être placé sur l'objet détecté (le visage de l'être humain). Ce modèle est un modèle statistique (suivant l'ensemble des données d'apprentissage) détermine l'espace dans lequel chaque point peut évoluer et son influence sur les positions des autres points.

En superposant les ensembles des points des visages, un nuage de points sera constitué (figure 4.14). Pour évaluer l'espace dans lequel évolue chaque point et surtout l'influence du mouvement d'un point sur les autres points, il est possible d'utiliser une méthode d'analyse de données qui est l'**Analyse en Composantes Principales (ou ACP)**. Cette analyse permet d'expliquer la variance observée dans l'ensemble des points des visages.

Les données d'entrées de l'ACP sont donc les ensembles de points de visages normalisés. Chaque ensemble de points est représenté par un vecteur de *2n-D* (comme décrit dans

Algorithm 4: Algorithme de normalisation des points

— Normaliser chaque ensemble de points par rapport au premier ensemble de point et ce en appliquant la rotation, la mise à l'échelle et la translation.
— **repeat**
  — Calculer le modèle moyen des ensembles des points alignés.
  — Normaliser le modèle moyen.
  — Aligner chaque ensemble de points par rapport au modèle moyen courant.
until *jusqu'à convergence*;

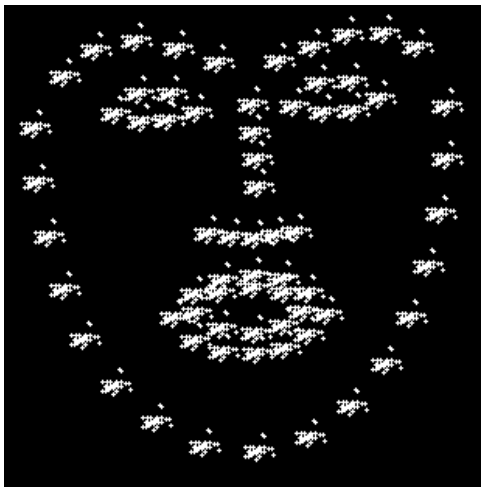

Fig. 4.14 Exemple de nuage de points d'un visage

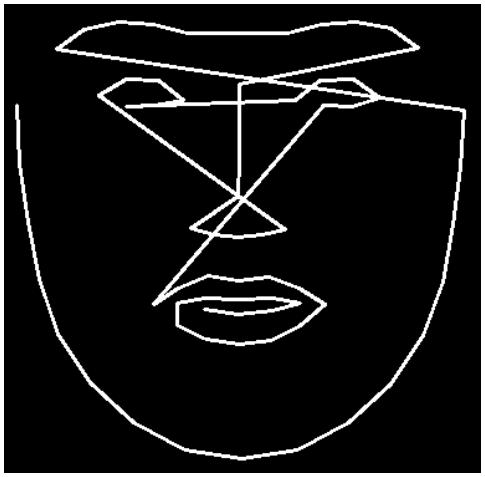

Fig. 4.15 Modèle moyen d'un visage

l'équation 4.8) où n est le nombre de points. Avec l'ACP il sera donc possible de réduire les dimensions des vecteurs et ce en limitant les pertes des données, ensuite il sera possible de calculer une approximation de la position de chaque point par rapport aux axes principaux trouvés à partir du nuage de points. L'algorithme 5 décrit le fonctionnement de l'analyse en composantes principales.

Par la suite il est possible de dire que n'importe quelle donnée d'apprentissage $\mathbf{x}$ utilisée est équivalente au modèle moyen généré additionné à la matrice des vecteurs propres $P = \{p_i\}$ multipliée par un vecteur de poids $\mathbf{b}$ et on écrit :

$$x \approx \bar{x} + Pb \tag{4.16}$$

Où $P = (p_1|p_2|...|p_t)$ une matrice contenant les t vecteurs propres et $\mathbf{b}$ est un vecteur de dimension t définit comme suit :

$$b = P^T(x - \bar{x}) \tag{4.17}$$

Avec l'équation 4.16 il est possible de générer d'autres modèles de visages similaires au modèle moyen calculé en variant les valeurs des composantes du vecteur $\mathbf{b}$. Par contre une limite existe pour les valeurs des composantes du vecteur $\mathbf{b}$ qui dépendent des valeurs propres

Algorithm 5: Algorithme de l'Analyse en Composantes Principales

1. Calculer modèle moyen des points :

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{4.12}$$

2. Calculer la covariance des données :

$$S = \frac{1}{N}\sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T \tag{4.13}$$

3. Calculer les vecteurs propres, p_i, et les valeurs propres correspondantes, λ_i, de $\mathbf{S}$. Trier par ordre décroissant les valeurs propres.

$$Sp_i = \lambda_i p_i \tag{4.14}$$

4. Chaque valeur propre permet de donner la variance des données par rapport au modèle moyen et ce suivant la direction indiquée par le vecteur propre correspondant. Calculer la variance totale :

$$V_T = \sum_i \lambda_i \tag{4.15}$$

λ_i, dans [4] la limite de chaque composante du vecteur $\mathbf{b}$ est donnée comme suit :

$$-3\sqrt{\lambda_i} \leq b_i \leq 3\sqrt{\lambda_i} \tag{4.18}$$

Application du modèle sur une image

Les étapes précédentes ont permis de générer un modèle flexible qu'il sera possible d'utiliser pour que cela s'adapte automatiquement et que cela prenne la forme de l'objet à suivre.

L'algorithme ASM utilise une méthode itérative pour trouver la meilleure disposition pour appliquer un modèle sur un objet. L'algorithme 6 décrit la fonction itérative permettant d'appliquer un modèle de points à un visage. Cet algorithme utilise les données renvoyées par **Haar Cascades** pour avoir la position de points de références, par exemple la position du visage en premier puis les positions des yeux et finalement la position de la bouche. Grâce à ces informations il est possible de générer un premier modèle à adapter d'une façon itérative jusqu'à la convergence. La figure 4.16 montre un exemple d'application du modèle après quelques itérations.

Algorithm 6: Algorithme de la fonction itérative pour adapter le modèle de points à un visage

repeat

1. Récupérer la position du visage, des yeux et de la bouche.
2. Examiner la région aux alentours de chaque point X_i du modèle, et ce pour trouver le bord sur le quel doit être positionner le point.
3. Mettre à jour les paramètres (X_t, Y_t, s, θ, b)
4. Vérifier que $|b_i| < 3\sqrt{\lambda_i}$

until *convergence*;

Fig. 4.16 Application du modèle avec ASM

4.3.2 Améliorations de l'algorithme ASM

Dans [17], **Millborrow** et **Nicolls** proposent une amélioration de l'algorithme **ASM**, en utilisant les descripteurs HAT (Histogram Array Transforms) qui est une version modifiée des descripteurs SIFT (Scale-Invariant Feature Transform). Ceci permet de prendre en considération plusieurs paramètres dans l'image, tel que le changement de l'intensité de la lumière.

Description du "Scale-Invariant Feature Transform"

Les descripteurs SIFT permettent d'extraire d'une image les régions d'intérêt qui peuvent aider pour le placement des points, par exemple les coins des yeux sont des régions d'intérêt. Il est aussi possible d'utiliser cet algorithme dans la modélisation 3D ou bien pour chercher les correspondances entre deux images ou même pour faire de l'assemblage d'images panoramique. Les descripteurs HAT ont l'avantage d'être plus rapide en calcul.

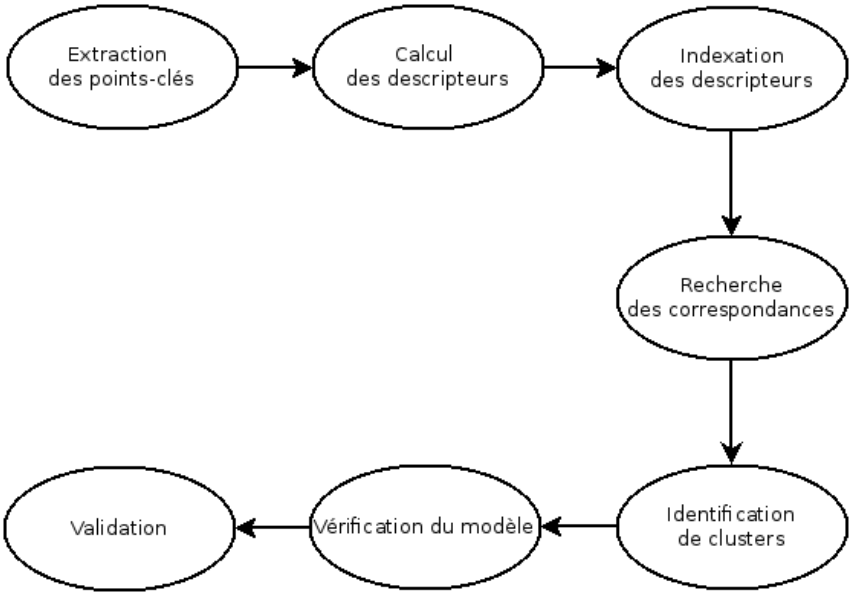

Fig. 4.17 Différentes étapes du SIFT

La figure 4.17 décrit les différentes étapes du SIFT qui utilise plusieurs images de références et une image cible et ce pour pouvoir trouver les correspondances. La première étape consiste à extraire les points clefs de l'ensemble des images. Pour cela il faut d'utiliser plusieurs techniques telles que la différence gaussienne et la pyramide de gradients. Ensuite il faut calculer les descripteurs de chaque point clef dans l'image. Un descripteur SIFT est en réalité un tableau d'histogrammes d'orientation qui indiquent la direction des gradients autour d'un point clef. Par la suite ces descripteurs sont indexés et ce pour permettre une recherche plus rapide. Une fois que l'indexation est faite il est possible de lancer une recherche entre deux images pour trouver les correspondances et d'identifier les clusters qui sont un groupe de correspondances, grâce à ceci il est possible de ne pas tenir compte des différentes poses qu'un objet peut prendre. Après cette identification, une vérification est faite sur l'image cible et l'image de référence et ce en utilisant les clusters. Et finalement une décision finale permet de dire à la suite des étapes précédentes si les objets dans les deux images correspondent ou pas. La figure 4.18 montre les correspondances relevées entre deux images ayant des poses différentes (les traits blancs indiquent la correspondance de chaque région).

Fig. 4.18 Exemple de correspondance entre deux images

Description du "Histogram Array Transforms"

Les descripteurs HAT sont quasiment les mêmes que les descripteurs SIFT sauf qu'il est possible de les calculer plus rapidement. Pour commencer les points clefs avec HAT sont déjà prédéterminés (ce sont les points caractéristiques définit avant). Le calcul des descripteurs HAT ne prend pas en considération plusieurs paramètre comme par exemple la mise à l'échelle ou la rotation des images de références.

Comme le montre la figure 2.1, le système ASM prend en paramètres d'entrées les coordonnées des points caractéristiques pour chaque image, ainsi que l'image en question. Cela permet donc, de calculer les descripteurs HAT en utilisant l'image.

Une fois les descripteurs sont calculés, ceci va permettre d'améliorer le positionnement des points et ce en se basant sur la comparaison des descripteurs dont dispose le système et les descripteurs d'une nouvelle image (ou visage).

4.3.3 Modèles de points caractéristiques du visage

Il existe plusieurs modèles de placement des points caractéristiques sur le visage et cela dépendant essentiellement des zones visées et du but final. Dans le cas du système de détection des émotions les zones visées sont les yeux, la bouche, les sourcils et le nez. Deux modèles existent pour cela :

— Le modèle Multi-PIE (figure 4.20).
— Le modèle MUCT/XM2VTS (figure 4.19).

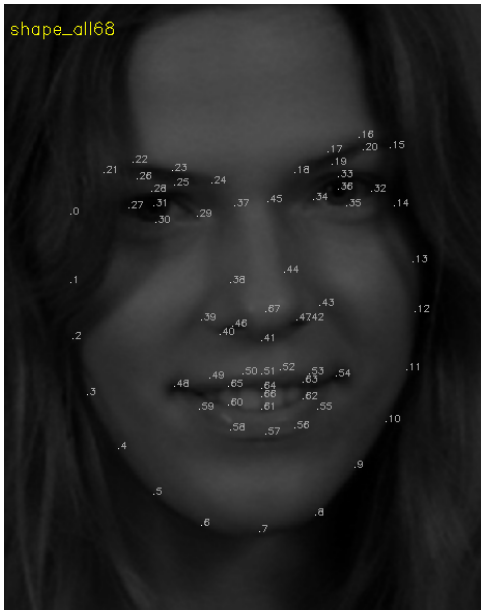

Fig. 4.19 Modèle MUCT/XM2VTS

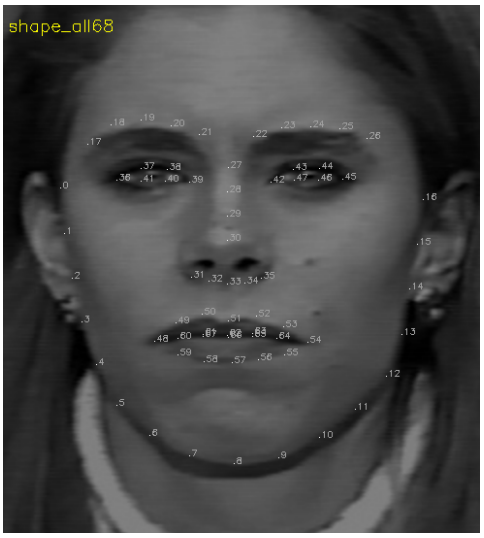

Fig. 4.20 Modèle Multi-PIE

Le système développé se base sur le modèle Multi-PIE et ce parce que la base de données des images utilisées pour l'apprentissage définit les points suivant ce modèle, ceci nous permet d'éviter l'étape où il faut manuellement placer les points pour avoir des données de références.

4.3.4 Apprentissage de ASM

Cette étape consiste à utiliser les données mises à disposition pour lancer un apprentissage ASM qui permet par la suite de placer les points caractéristiques sur d'autres visages. Pour cela nous avons utilisées les images et les coordonnées des points caractéristiques prédéfinis.

Cet apprentissage nécessite deux données :

— Les images.

— Le fichier contenant les modèles des visages (les coordonnées des points).

À la fin de l'apprentissage cette librairie génère des fichiers C++ à utiliser pour recompiler la librairie "STASM". Ces fichier C++ contiennent le modèle du visage déduit des données d'entrées.

Le fichier contenant les modèles des visages est un fichier particulier qui a la structure suivante :

```
1  Directories /chemin/vers/les/images
2
3  0000 fichier1 {
4    68 2
5    253.38504 225.7277
6    255.7222 248.9568
7    259.73156 271.88716
```

```
8           264.70656 294.58636
9           ...
10        }
11
12        0000 fichier2 {
13          68 2
14          182.82938 192.01613
15          184.18566 222.41096
16          188.9035 252.22931
17          196.54442 281.35039
18          ...
19        }
```

Pour chaque fichier il y a 4 bits (ligne 3 et ligne 12) à définir qui permettent de donner plus de détails sur le contenu de l'image, c'est à dire indiquer si la personne porte des lunettes ou pas, si la bouche est ouverte, ou si la personne possède une moustache, etc. Ensuite il faut déclarer les coordonnées de chaque point mais avant cela il faut dire combien de points seront déclarées et ce en combien de colonnes (ligne 4 et ligne 13), dans notre cas 68 est le nombre de points et 2 veut dire que chaque point possède une coordonnée x et une coordonnée y.

Dans l'apprentissage lancé nous avons ignoré les bits supplémentaires qui permettent de données plus de détails sur le visage et ce parce que ces informations ne sont pas très importantes pour la détection des émotions.

4.4 Détection des émotions

La dernière étape consiste à analyser le mouvement des points caractéristiques et en déduire l'émotion exprimée. La détection des émotions se fait en deux temps[10][30] :

— Création de modèles de références pour chacune des émotions de base (joie, tristesse, colère, dégoût, surprise, peur). Ces modèles de références seront utilisés par la suite pour permettre de faire la classification et donc savoir quelle est l'émotion exprimée.

— Création du classificateur qui sera utilisé pour pouvoir comparer entre l'émotion exprimée et les modèles des émotions de référence.

4.4.1 Création de modèles de références des émotions

La création de modèles de références des émotions est faite en utilisant la base de données des émotions [15]. Le problème avec les émotions dans cette base de données est que la durée d'une même émotion peut varier d'un sujet à un autre, par exemple une l'émotion de la joie peut être exprimée entre 0.33 seconde (10 frames) et 0.93 seconde (28 frames).

Pour cela nous avons préalablement normalisé le nombre d'images par émotion. Pour ce faire nous avons fixé la durée d'une émotion à la moyenne des durées c'est à dire à *une*

seconde (30 images). La normalisation a été réalisée en dupliquant les images en partant de la dernière image, qui montre l'émotion à son intensité maximale, et en revenant jusqu'à la première image, qui montre un visage neutre.

Normalisation des coordonnées des points caractéristiques

La première étape à faire est de centrer toutes les coordonnées des points caractéristiques, c'est à dire de ramener tous ces points au même point d'origine sur un repère.

Pour chaque séquence d'une émotion nous pouvons écrire :

$$S^k = \{p_0^k, p_1^k, p_2^k, ..., p_{67}^k\} \tag{4.19}$$

Où k est le numéro de la séquence et p_i^k définit les coordonnées du point i de la séquence k dans chaque image (frame). Donc on écrit :

$$p_i^k = \{(x_0, y_0)_i^k, (x_1, y_1)_i^k, (x_2, y_2)_i^k, ..., (x_{29}, y_{29})_i^k\} \tag{4.20}$$

Pour chaque coordonnées $(x_l, y_l)_i^k$ nous avons :
— l définit le numéro du frame.
— i définit le numéro du point.
— k définit le numéro de la séquence (vidéo).
Par conséquent nous aurons pour chaque expression d'une émotion la matrice de pré-normalisation suivante de taille 30 (frames) x 68 (points) :

$$\begin{array}{c} \\ 68Points \end{array} \begin{array}{c} \text{30 Frames} \\ \begin{bmatrix} (x_0, y_0)_0^k & (x_1, y_1)_0^k & \cdots & (x_{29}, y_{29})_0^k \\ (x_0, y_0)_1^k & (x_1, y_1)_1^k & \cdots & (x_{29}, y_{29})_1^k \\ \vdots & \vdots & \ddots & \vdots \\ (x_0, y_0)_{67}^k & (x_1, y_1)_{67}^k & \cdots & (x_{29}, y_{29})_{67}^k \end{bmatrix} \end{array}$$

Pour commencer il faut calculer un visage neutre moyen (fig 4.21) en utilisant la première image de chaque séquence d'émotion, nous obtenons donc le vecteur de points suivant :

$$68Points \begin{bmatrix} (\mu_x, \mu_y)_0 \\ (\mu_x, \mu_y)_1 \\ \vdots \\ (\mu_x, \mu_y)_{67} \end{bmatrix}$$

Où $(\mu_x, \mu_y)_i$ sont les coordonnées moyennes des frames 0 (visage neutre) de toutes les séquences et i indique le numéro du point.

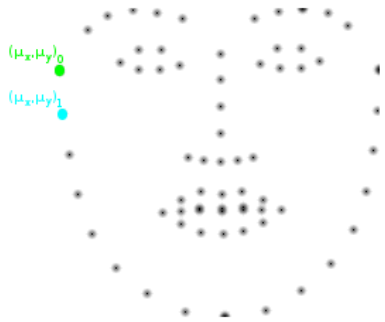

Fig. 4.21 Représentation graphique du visage neutre moyen

Le calcul des nouvelles coordonnées de chaque séquences se fait en deux étapes. La première étape consiste à calculer le déplacement entre chaque point du frame 0 de la séquence et le visage neutre moyen calculé précédemment. On écrit alors :

$$(\delta_x, \delta_y)_i^k = (\mu_x - x_0, \mu_y - y_0)_i^k \tag{4.21}$$

Où :
— i indique le numéro du point.
— k indique le numéro de la séquence (vidéo).
— x_0 et y_0 sont les coordonnées du point i dans le frame 0.
— μ_x et μ_y sont les coordonnées du point i dans le visage neutre moyen.
Ainsi on peut écrire les nouvelles coordonnées normalisées d'un même point comme suit :

$$p_i'^k = \{(x_0+\delta_x, y_0+\delta_y)_i^k, (x_1+\delta_x, y_1+\delta_y)_i^k, (x_2+\delta_x, y_2+\delta_y)_i^k, ..., (x_{29}+\delta_x, y_{29}+\delta_y)_i^k\} \tag{4.22}$$

On peut donc écrire :

$$S'^k = \{p_0'^k, p_1'^k, p_2'^k, ..., p_{67}'^k\} \tag{4.23}$$

Et la nouvelle matrice des coordonnées normalisées devient :

$$\begin{array}{c} \\ 68Points \end{array} \begin{array}{c} \overset{\text{30 Frames}}{\overbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}} \\ \begin{bmatrix} (x_0+\delta_x, y_0+\delta_y)_0^k & (x_1+\delta_x, y_1+\delta_y)_0^k & \cdots & (x_{29}+\delta_x, y_{29}+\delta_y)_0^k \\ (x_0+\delta_x, y_0+\delta_y)_1^k & (x_1+\delta_x, y_1+\delta_y)_1^k & \cdots & (x_{29}+\delta_x, y_{29}+\delta_y)_1^k \\ \vdots & \vdots & \ddots & \vdots \\ (x_0+\delta_x, y_0+\delta_y)_{67}^k & (x_1+\delta_x, y_1+\delta_y)_{67}^k & \cdots & (x_{29}+\delta_x, y_{29}+\delta_y)_{67}^k \end{bmatrix} \end{array}$$

De cette façon nous avons pu normaliser toutes les coordonnées de toutes les séquences d'émotions. L'étape qui suit consiste à extraire les vecteurs de données caractéristiques (ou

"features") de ces séquences.

Extraction des vecteurs de données caractéristiques ("features")

Il existe deux types de vecteurs de données à extraire de chaque séquence vidéo :
— Le premier vecteur de caractéristiques est **un vecteur de déplacement** qui décrit l'évolution de chaque point indépendamment des autres.
— Le deuxième vecteur est **un vecteur angulaire** qui décrit l'évolution de chaque point relativement aux autres.
Le frame 0 (visage neutre) de chaque séquence est pris comme référence.

Pour simplifier nous pouvons réécrire la formule 4.22 comme suit :

$$p_i'^k = \{(x\prime_0, y\prime_0)_i^k, (x\prime_1, y\prime_1)_i^k, (x\prime_2, y\prime_2)_i^k, ..., (x\prime_{29}, y\prime_{29})_i^k\} \tag{4.24}$$

Le vecteur de déplacement est calculé en prenant chaque séquence à part et en calculant la différence des coordonnées d'un point d'un frame i et du frame 0 :

$$\delta p_i'^k = \{(0,0)_i^k, (x\prime_1 - x\prime_0, y\prime_1 - y\prime_0)_i^k, (x\prime_2 - x\prime_0, y\prime_2 - y\prime_0)_i^k, ..., (x\prime_{29} - x\prime_0, y\prime_{29} - y\prime_0)_i^k\} \tag{4.25}$$

Pour simplifier la formule 4.25 nous pouvons écrire pour chaque point d'une même séquence :

$$\delta p_i'^k = \{(0,0)_i^k, (\delta x\prime_1, \delta y\prime_1)_i^k, (\delta x\prime_2, \delta y\prime_2)_i^k, ..., (\delta x\prime_{29}, \delta y\prime_{29})_i^k\} \tag{4.26}$$

Par conséquent le vecteur de déplacement d'une séquence entière est :

$$\delta S'^k = \{\delta p_0'^k, \delta p_1'^k, \delta p_2'^k, ..., \delta p_{67}'^k\} \tag{4.27}$$

Chaque vecteur de déplacement de chaque séquence a une taille de 68 (couples de données) qui est le nombre de points caractéristiques sur le visage ("Landmarks").

le vecteur angulaire est calculé par rapport à chaque deux couples de points, où on détermine l'angle θ [1] que forme les deux segments et la distance d [2] entre leurs centres [3].

Pour commencer nous calculons le vecteur angulaire du frame 0 (visage neutre) par la suite nous calculons le changement du vecteur angulaire des autres frames par rapport au frame 0 d'une même séquence.

1. $angle(radian) = \arccos \frac{\vec{AB}.\vec{CD}}{||\vec{AB}||.||\vec{CD}||}$.
2. $distance = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$
3. $centre = (\frac{x_A + x_B}{2}, \frac{y_A + y_B}{2})$

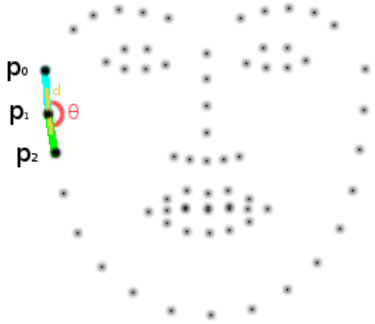

Fig. 4.22 Calcul de l'angle θ et de la distance d entre les deux couples de points (p_0, p_1) et (p_1, p_2)

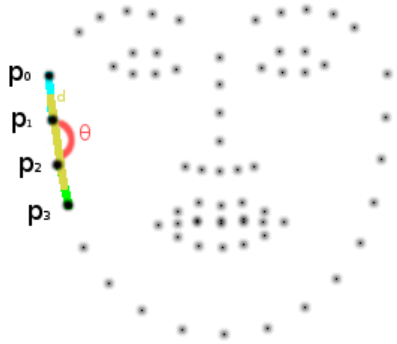

Fig. 4.23 Calcul de l'angle θ et de la distance d entre les deux couples de points (p_0, p_1) et (p_2, p_3)

Nous écrivons alors le vecteur angulaire de chaque deux couples $(i,\ j)$ de points comme suit :

$$A_{i,j}^{k} = \{(d_0, \theta_0)_{i,j}^{k}, (d_1, \theta_1)_{i,j}^{k}, (d_2, \theta_2)_{i,j}^{k}, ..., (d_{29}, \theta_{29})_{i,j}^{k}\} \tag{4.28}$$

Ensuite nous calculons le δA c'est à dire le changement de chaque frame par rapport au frame 0. On a :

$$(\delta d_l, \delta\theta_l)_{i,j}^{k} = (d_l - d_0, \theta_l - \theta_0) \tag{4.29}$$

Où :

— i est le premier couple de points.
— j est le deuxième couple de points.
— l est le numéro du frame.
— k est le numéro de la séquence.

Ce qui nous donne :

$$\delta A_{i,j}^{k} = \{(d_1 - d_0, \theta_1 - \theta_0)_{i,j}^{k}, (d_2 - d_0, \theta_2 - \theta_0)_{i,j}^{k}, ..., (d_{29} - d_0, \theta_{29} - \theta_0)_{i,j}^{k}\} \tag{4.30}$$

Pour simplifier nous pouvons écrire :

$$\delta A_{i,j}^{k} = \{(\delta d_1, \delta\theta_1)_{i,j}^{k}, (\delta d_2, \delta\theta_2)_{i,j}^{k}, ..., (\delta d_{29}, \delta\theta_{29})_{i,j}^{k}\} \tag{4.31}$$

Pour chaque séquence le vecteur angulaire est de taille $L * (L-1)/2$ où L est le nombre de points caractéristiques autrement dit dans notre cas $L = 68$. Donc le vecteur angulaire pour une seule séquence est de taille 2278 couples de données.

Création des modèles de références pour chaque émotion

À la fin de l'extraction des vecteurs de données ("features") chaque séquence d'une émotion aura $2278 + 68 = 2346$ couples de données.

Les séquences utilisées sont déjà étiquetées, c'est à dire qu'on connaît déjà l'émotion exprimée dans la séquence. Soit U l'ensemble des séquences des émotions que nous avons traitées précédemment, et soit $c \in \{\text{joie, peur, colère, surprise, dégoût, tristesse}\}$. U_c est le sous ensemble des séquences qui expriment la même émotion (U_{joie} est l'ensemble des séquences de l'émotion de la joie).

Dans cette étape nous devons créer pour chaque ensemble de séquences d'une même émotion un modèle de référence du **vecteur de déplacement** et du **vecteur angulaire**, ceci en utilisant les vecteurs générés durant l'étape précédente. Pour cela nous calculons la médiane pour chaque valeur de chaque point pour un même frame. L'utilisation de la médiane à la place de la moyenne, permet d'éviter la déformation qui peut être provoquée par des valeurs aberrantes.

Soit c une des six émotions de base, U_c est l'ensemble de séquences appartenant à la même classe d'émotion c et soit k la séquence d'émotion tel que $k \in U_c$. Pour chaque frame l et pour chaque point i nous calculons la médiane du **vecteur de déplacement** sur l'ensemble des séquences :

$$P(\delta p_i^l) = \underset{k \in U_c}{median}((\delta x_l, \delta y_l)_i^k) \tag{4.32}$$

Nous obtenons alors la matrice de déplacement prototype suivante :

$$\begin{array}{c} \\ 68 Points \end{array} \begin{array}{c} \overset{\text{30 Frames}}{\begin{bmatrix} \underset{k \in U_c}{median}((\delta x_0, \delta y_0)_0^k) & \underset{k \in U_c}{median}((\delta x_1, \delta y_1)_0^k) & \cdots & \underset{k \in U_c}{median}((\delta x_{29}, \delta y_{29})_0^k) \\ \underset{k \in U_c}{median}((\delta x_0, \delta y_0)_1^k) & \underset{k \in U_c}{median}((\delta x_1, \delta y_1)_1^k) & \cdots & \underset{k \in U_c}{median}((\delta x_{29}, \delta y_{29})_1^k) \\ \vdots & \vdots & \ddots & \vdots \\ \underset{k \in U_c}{median}((\delta x_0, \delta y_0)_{67}^k) & \underset{k \in U_c}{median}((\delta x_1, \delta y_1)_{67}^k) & \cdots & \underset{k \in U_c}{median}((\delta x_{29}, \delta y_{29})_{67}^k) \end{bmatrix}} \end{array}$$

Le **vecteur angulaire prototype** est calculé de la même façon sauf qu'au lieu d'avoir le déplacement $(\delta x, \delta y)$ pour chaque point, nous avons le changement de la distance entre les centres et le changement de l'angle $(\delta d, \delta\theta)$ pour chaque deux couples de points i et j. Ainsi nous avons :

$$P(\delta a_{i,j}^l) = \underset{k \in U_c}{median}((\delta d_l, \delta\theta_l)_{i,j}^k) \tag{4.33}$$

Nous obtenons alors la matrice angulaire prototype suivante :

$$
\begin{array}{c} \\ 68Points \end{array}
\begin{array}{c}
\begin{array}{cccc} \multicolumn{4}{c}{30\ \text{Frames}} \end{array} \\
\begin{bmatrix}
\underset{k\in U_c}{median}((\delta d_0, \delta\theta_0)^k_{0,1}) & \underset{k\in U_c}{median}((\delta d_1, \delta\theta_1)^k_{0,1}) & \cdots & \underset{k\in U_c}{median}((\delta d_{29}, \delta\theta_{29})^k_{0,1}) \\
\underset{k\in U_c}{median}((\delta d_0, \delta\theta_0)^k_{0,2}) & \underset{k\in U_c}{median}((\delta d_1, \delta\theta_1)^k_{0,2}) & \cdots & \underset{k\in U_c}{median}((\delta d_{29}, \delta\theta_{29})^k_{0,2})) \\
\vdots & \vdots & \ddots & \vdots \\
\underset{k\in U_c}{median}((\delta d_0, \delta\theta_0)^k_{67,68}) & \underset{k\in U_c}{median}((\delta d_1, \delta\theta_1)^k_{67,68}) & \cdots & \underset{k\in U_c}{median}((\delta d_{29}, \delta\theta_{29})^k_{67,68}))
\end{bmatrix}
\end{array}
$$

De cette façon nous avons pu construire un modèle prototype pour chaque émotion et pour chaque vecteur de données (**vecteur de déplacement** et **vecteur angulaire**). Ces modèles prototypes seront utilisés pour permettre la classification et donc déterminer l'émotion à partir des expressions faciales.

Fig. 4.24 Représentation de l'intensité maximale des expressions faciales prototypes (frame 30) de gauche vers la droite : colère, joie, dégoût, peur, tristesse et surprise.

4.4.2 Création du classificateur "Dynamic Time Warping"

Les étapes précédentes nous ont permis de créer des modèles de références à utiliser pour la comparaison, c'est à dire que nous pouvons maintenant calculer le taux de ressemblance d'une nouvelle séquence d'émotion avec les modèles générés et en déduire l'émotion.

Les modèles prototypes générés sont des modèles types d'une durée d'une seconde (30 frames), et les nouvelles séquences d'émotions peuvent ne pas être de la même durée (une même émotion peut avoir une durée variable d'une personne à une autre). Pour cela nous nous sommes basés sur l'algorithme de **la déformation temporelle dynamique**[13] (ou "**Dynamic Time Warping (DTW)**").

Cet algorithme permet le calcul du taux de ressemblance entre deux séquences qui varient en vitesse et en longueur (la reconnaissance vocale est une des applications de cet algorithme).

Pour chaque nouvelle séquence nous extrayons les vecteurs de données (**vecteur de déplacement** et **vecteur angulaire**), que nous comparons avec les vecteurs prototypes générés précédemment en utilisant DTW.

La comparaison se fait en calculant la distance DTW entre la séquence prototype et la séquence à comparer. Pour cela nous construisons une matrice de distances (fig 4.26) de taille $(n, 30)$, où n est le nombre de frames de la séquences à comparer et 30 est celui de la

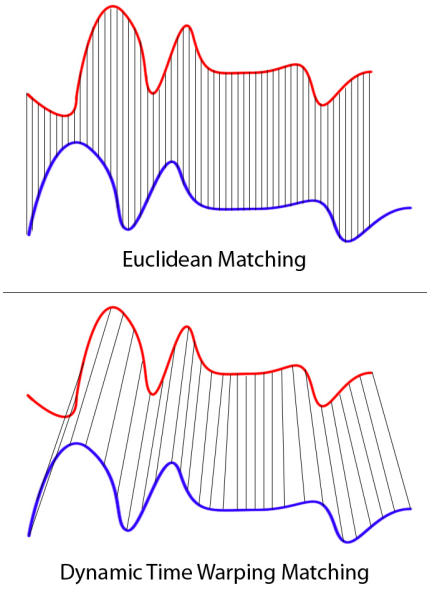

Fig. 4.25 Correspondance euclidienne et correspondance DTW

séquence prototype.

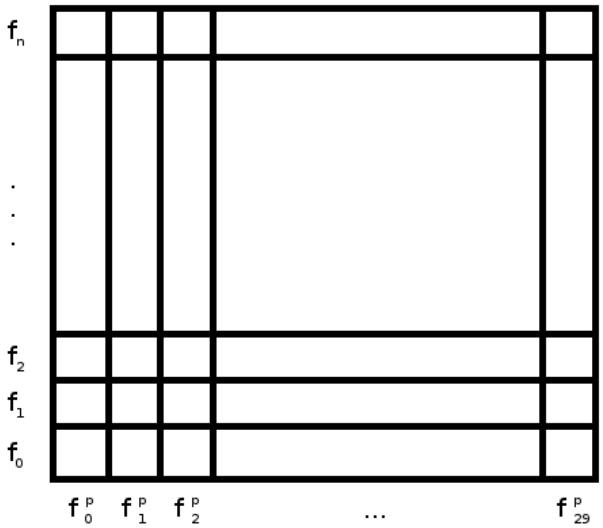

Fig. 4.26 Matrice de distances DTW

Dans chaque case $(i,j)_{0\leq i<n, 0\leq j<30}$ nous calculons la distance entre le frame i de la séquence à classifier et le frame j de la séquence prototype comme suit :

$$M[i,j] = dist(f_i, f_j^p) + \min(dist(f_{i-1}, f_j^p), dist(f_i, f_{j-1}^p), dist(f_{i-1}, f_{j-1}^p)) \tag{4.34}$$

Pour chaque frame, que ce soit un frame prototype ou un frame d'une séquence à classifier,

nous avons une matrice écrite sous la forme $f_i = \begin{pmatrix} x_0^i & y_0^i \\ x_1^i & y_1^i \\ \vdots & \vdots \\ x_n^i & y_n^i \end{pmatrix}$ et $f_j = \begin{pmatrix} x_0^j & y_0^j \\ x_1^j & y_1^j \\ \vdots & \vdots \\ x_n^j & y_n^j \end{pmatrix}$. La distance entre ces deux frames f_i et f_j est définie comme suit :

$$dist(f_i, f_j) = \sum_{k=0}^{n}((x_k^i - x_k^j)^2 + (y_k^i - y_k^j)^2) \tag{4.35}$$

Une fois que les valeurs de la matrice sont calculées nous récupérons la valeur de la dernière case $(n, 30)$ de la matrice, cette valeur indique la distance finale entre la séquence à classifier et la séquence prototype. Ensuite nous comparons cette distance avec toutes les autres distances calculées en utilisant les autres séquences prototypes. La plus petite distance nous permet de déduire la classe à laquelle appartient la séquence à classifier, autrement dit l'émotion exprimée.

Ce calcul est fait une première fois en utilisant le **vecteur de déplacement** ce qui nous renvoie une première valeur de distance, puis une deuxième fois selon le **vecteur angulaire**. La moyenne des deux valeurs est utilisée pour la comparaison et donc pour la classification.

Grâce à ces algorithmes, nous avons pu réaliser un détecteur d'émotions à partir des expressions faciales. Ce système peut être intégré à l'assistant culinaire pour qu'il renvoie une information supplémentaire sur l'état émotionnel de la personne utilisant la cuisine, et donc ceci permettra de mieux savoir la guider.

Chapitre 5

Tests, résultats, perspectives et conclusion

La réalisation de ce travail a été faite en utilisant la base de données [15] des émotions. Cette base de données offre 593 séquences de différentes émotions de base. Ces émotions sont exprimées par 123 sujets de différentes ethnies et de différents sexes. Toutefois nous avons divisé cette base de données en deux parties :

— Une première partie qui représente les 2/3 des données a été utilisée pour l'apprentissage et la création des modèles types des émotions.

— Une deuxième partie qui représente le 1/3 des données a été utilisée pour les tests.

Cependant nous ne nous sommes pas limité au seulement le 1/3 de la base de données pour faire les tests, nous avons aussi lancé le test sur les 2/3 utilisés pour l'apprentissage.

5.1 Tests et résultats

5.1.1 Détection du visage et placement des points caractéristiques

La première partie du test consiste à détecter le visage, les yeux et la bouche sur l'ensemble des images dont on dispose. Ceci est fait en lançant un apprentissage ASM pour permettre la création du modèle de points caractéristiques. Donc en utilisant l'outil d'apprentissage de ASM et en lui passant toutes les images disponibles dans la base de données nous avons obtenus ce qui suit :

> *10692 faces detected (99.85%), 10692 facedets in the shapefile, 0 facedets in the images 0 facedet false positives, 0 shapes with missing landmarks 10692 faces actually used (valid facedet and all points) (99.85% of the shapes in the shapefile) 3 missing both eyes, 1 missing just left eye, 28 missing just right eye, 127 missing mouth*

Dans certaines des images le système n'a pas pu détecter les yeux parce qu'ils étaient fermés et donc en lançant l'algorithme Haar-cascades il lui ai impossible de satisfaire les conditions nécessaires pour dire qu'il trouve les yeux. Concernant la bouche, dans certains cas la bouche est cachée par les mains, c'est ce qu'il fait que le système n'arriver pas à détecter la bouche.

Sur 10 708 images l'algorithme a pu détecter 10 692 visages ce qui représente un taux de réussite de 99.85%.

5.1.2 Détection des émotions

Le premier test a été fait sur les 2/3 de données utilisées pour l'apprentissage. Ces mêmes données ont été utilisées comme information d'entrée pour le système. La classification a été faite en comparant par rapport aux modèles prototypes générés. Le tableau suivant 5.1 résume les résultats obtenus :

	Colère	**Joie**	**Tristesse**	**Peur**	**Dégoût**	**Surprise**
Colère	90%	0%	6.66%	0%	3.33%	0%
Joie	2%	94%	2%	2%	0%	0%
Tristesse	5%	0%	95%	0%	0%	0%
Peur	0%	13.33%	6.66%	80%	0%	0%
Dégoût	2.27%	0%	2.27%	0%	95.45%	0%
Surprise	0%	0%	4.83%	6.45%	0%	88.70%

Tab. 5.1 Résultat de reconnaissance des émotions et taux de confusion avec les autres émotions (données utilisées pour l'apprentissage)

Le deuxième test effectué se base sur les données qui n'ont pas été utilisées pour l'apprentissage. Le tableau 5.2 suivant résume les résultats obtenus :

	Colère	**Joie**	**Tristesse**	**Peur**	**Dégoût**	**Surprise**
Colère	86.66%	0%	13.33%	0%	0%	0%
Joie	5.26%	94.73%	0%	0%	0%	0%
Tristesse	12.5%	0%	87.5%	0%	0%	0%
Peur	0%	0%	10%	90%	0%	0%
Dégoût	13.33%	0%	0%	0%	86.66%	0%
Surprise	0%	0%	0%	5%	0%	95%

Tab. 5.2 Résultat de reconnaissance des émotions et taux de confusion avec les autres émotions (données non utilisées pour l'apprentissage)

Ces résultats montrent que le taux de reconnaissance le plus faible est de 86.66% sur des données qui n'ont pas été utilisées pour faire de l'apprentissage.

En analysant ces résultats, nous remarquons que des fois le système confond une émotion avec une autre (par exemple Colère/Joie ou Surprise/Tristesse). Ceci est dû au fait que ces émotions sont très proches (selon le modèle des émotions de Plutchik) et sont généralement soit adjacentes directement, soit adjacentes à une émotion près. La roue des émotions de

Plutchik (fig. 1.1) vient appuyer ceci en illustrant ces émotions. Cependant le taux d'erreur reste relativement faible.

En plus du résultat du détection des émotions, un autre test a été fait et qui concerne le temps d'exécution. En moyenne la normalisation des données d'une séquence à classifier prend **0.01 seconde** (la durée d'une séquence varie et peut atteindre le 90 frames). L'extraction du **vecteur de déplacement** prend en moyenne **0.015 seconde** et l'extraction du **vecteur angulaire** prend **0.5 seconde**.

Concernant la classification en moyenne, la classification selon le **vecteur de déplacement** seulement **0.04 seconde**. Par contre la classification selon le **vecteur angulaire** seulement est légèrement plus longue et elle nécessite un temps moyen de **1.5 seconde**.

La classification est faite d'une façon séquentielle et en utilisant toutes les caractéristiques extraites, c'est ce qui explique ce temps d'exécution qui peut parfois être long.

5.2 Perspectives

Il existe plusieurs améliorations possibles pour ce système, et ce que ce soit pour le rendre plus performant ou plus rapide. Parmi ces améliorations :

- Paralléliser la classification en utilisant **TBB**.
- Minimiser le nombre de caractéristiques extraites des vecteurs de données (**vecteur de déplacement** et **vecteur angulaire**), et ce en utilisant **Adaboost** pour sélectionner celles qui sont les plus pertinentes pour la classification.
- Utiliser une base de données plus grande qui offrent plus de séquences et plus de sujets.
- Utiliser une classification plus performante qui se base sur **SVM**[24] (Machine à Vecteurs de Support, ou "Support Vector Machines").
- Utiliser un réseau de neurones multicouches pour l'apprentissage des émotions et la classification[24].
- Rajouter d'autres informations telles que la voix, les signes vitaux, les mouvements du corps et les capteurs de la cuisine, pour mieux donner une estimation de l'émotion exprimée.
- Rajouter la détection des émotions de plusieurs personnes en même temps.

Mis à part les améliorations possibles, d'autres tests peuvent être faits avec ce système par exemple :

- Tester le système avec des sujets ayant un traumatisme cranio-cérébral. Pour cela nous avons besoin de construire une base de données d'émotions spécifique à cette catégorie de personnes.
- Comparer le système avec un autre qui se base sur la reconnaissance d'une émotion à partir d'une image seule et non pas d'une séquence.

5.3 Conclusion

Les émotions sont des informations importantes à récupérer et à traiter. Le visage de la personne est un des supports de ces informations. Grâce au système développé, nous avons pu décoder les expressions faciales et en déduire les émotions de la personne, tout ceci en gardant un taux d'erreur faible. Ce système permettra de renvoyer une information, à ne pas négliger, à l'assistant culinaire, qui pourrait utiliser pour mieux savoir guider la personne et surtout anticiper les accidents et les éviter.

Notre système peut être considéré comme étant un système préventif, il peut être intégré dans n'importe quelle partie de la maison intelligente. Sa structure modulaire permet d'apporter des améliorations d'une façon indépendante à chacun des modules.

La réalisation de ce projet a demandé non seulement des connaissances en informatique pour le développement et l'optimisation, mais aussi des connaissances en imagerie, en intelligence artificielle, en mathématiques et en psychologie.

Ce projet m'a permis, non seulement, de combiner deux domaines qui me fascinent beaucoup et qui sont l'informatique et l'intelligence artificielle d'un côté et la psychologie d'un autre côté, mais aussi d'apporter ma contribution dans le projet de l'assistant culinaire.

Pour conclure je suis satisfait de mon expérience au sein du laboratoire DOMUS, où j'ai pu approfondir mes connaissances et consolider mes acquis en premier lieu, et aussi participer au partage de connaissances avec tous les autres membres du laboratoire. Tout cela m'a permis de vivre une expérience très enrichissante.

Bibliographie

[1] Aitor Azcarate, Felix Hageloh, Koen van de Sande et Roberto Valenti. "Automatic facial emotion recognition". In : (2005) (cf. p. vi).

[2] Jeffrey Cohn, Zara Ambadar et Paul Ekman. "Observer-based mezasurement of facial expression with the Facial Action Coding System". In : *The handbook of emotion elicitation and assessment. Oxford University Press Series in Affective Science*. J. A. Coan & J. B. Allen, 2006 (cf. p. 10).

[3] T.F. Cootes, G.J. Edwards et C.J. Taylor. "Active Appearance Models". In : (1998) (cf. p. 16).

[4] T.F. Cootes, C.J. Taylor, D.H. Cooper et J. Graham. "Active Shape Models - Their Training and Application". In : *Computer And Image Understanding Vol. 61, No 1 January, pp 38-59* (1995) (cf. p. 16, 35, 38).

[5] Tim Cootes. "An Introduction to Active Shape Models". In : () (cf. p. 16).

[6] Douglas DeCarlo et Dimitris Metaxas. "The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation". In : (1996) (cf. p. 16).

[7] Paul Ekman. *Emotions Revelead Recognizing Faces and Feelings to Improve Communication and Emotional Life*. Times Books Henry Holt et Company, 2003. isbn : 0-8050-7275-6 (cf. p. vi).

[8] Paul Ekman, Wallace V. Friesen et Joseph C. Hager. *Facial Action Coding System*. Research Nexus division of Network Information Research Corporation, 2002. isbn : 0-931835-01-1 (cf. p. 10–12).

[9] Paul Ekman, Thomas S. Huang, Terrence J. Sejnowski et Josep C. Hager. *Final Report To NSF of the Planning Workshop on Facial Expression Understanding*. 1992.

[10] Deepak Ghimire et Joonwhoan Lee. "Geometric Feature-Based Facial Expression Recognition in Image Sequences Using Multi-Class AdaBoost and Support Vector Machines". In : *Sensors* (2013) (cf. p. 43).

[11] J. C. Gower. "Generalized Procrustes analysis". In : *Psychometrika* (1975) (cf. p. 35).

[12] Jay P. Kapur. "Face Detection in Color Images". In : *EE499 Capstone Design Project Spring 1997* (1997) (cf. p. 15).

[13] Daniel Lemire. “Faster Retrieval with a Two-Pass Dynamic-Time-Warping Lower Bound”. In : *Elsevier* (2009) (cf. p. 49).

[14] Yang Li. “Protractor : A Fast and Accurate Gesture Recognizer”. In : (2010) (cf. p. 35).

[15] Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih et Zara Ambadar. “The Extended Cohn-Kanade Dataset (CK+) : A Complete dataset for action unit and emotion-specified expression”. In : *CVPR for Human Communicative Behavior Analysis* (2010) (cf. p. vi, 31, 33, 43, 52).

[16] B. Menser et F Muller. “Face Detection In Color Images Using Principal Components Analysis”. In : () (cf. p. 15).

[17] Stephen Milborrow et Fred Nicolls. “Active Shape Models with SIFT Descriptors and MARS”. In : *VISAPP* (2014). `http://www.milbo.users.sonic.net/stasm` (cf. p. 16, 39).

[18] Margarita Osadchy, Yann Le Cun et Matthew L. Miller. “Synergistic Face Detection and Pose Estimation with Energy-Based Models”. In : *Journal Of Machine Learning 8* (2007) (cf. p. 16).

[19] Allan Pease. *Body Language How to read others' thoughts by their gesture*. Sheldon Press, 1984. isbn : 0-85969-406-2 (cf. p. 6).

[20] de l'évaluation et de la statistique du ministère de la Famille et des Aînés Direction de la recherche. *Les aînés du Québec*. 2012 (cf. p. iv).

[21] Stephen Robbins, Timothy Judge et Véronique Tran. *Comportements organisationnels*. Pearson Education, 2011. isbn : 2744074845 (cf. p. 7).

[22] Moritz Storring, Hans J. Andersen et Erik Granum. “Skin colour detection under changing lighting conditions”. In : (1999) (cf. p. 15).

[23] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato et Lior Wolf. “DeepFace : Closing the Gap to Human-Level Performance in Face Verification”. In : *Conference on Computer Vision and Pattern Recognition (CVPR)* () (cf. p. 16).

[24] Pang-Ning Tan, Michael Steinbach et Vipin Kumar. *Introduction to Data Mining*. Pearson Education, 2006. isbn : 0-321-32136-7 (cf. p. 17, 54).

[25] Imen TAYARI, Nhan LE THANH et Chokri BEN AMAR. *Modélisation des états émotionnels par un espace vectoriel multidimensionnel*. 2009. url : `http://www.i3s.unice.fr/~mh/RR/2009/RR-09.19-I.TAYARI.pdf` (cf. p. 8).

[26] Paul Viola et Michael Jones. “Rapide Object Detection using a Boosted Cascade of Simple Features”. In : 2001 (cf. p. 16, 25, 28, 30).

[27] Paul Viola et Michael J. Jones. “Robust Real-Time Face Detection”. In : *Internation Journal of Computer Vision* (2003) (cf. p. 16, 25).

[28] Laurenz Wiskott, Jean-Marc Fellous, Norbert Kruger et Christoph von der Malsburg. “Face Recognition by Elastic Bunch Graph Matching”. In : *Intelligent Biometric Techniques in Fingerprint and Face Recognition* (1999) (cf. p. 16).

[29] Jacob O. Wobbrock, Andrew D. Wilson et Yang Li. “Getures Without Libraries, Toolkits or Training : A $1 Recognizer for User Interface Prototypes”. In : (2007) (cf. p. 35).

[30] Yue Wu, Zuoguan Wang et Qiang Ji. “Facial Feature Tracking under Varying Facial Expressions and Face Poses based on Restricted Boltzmann Machines”. In : *IEEE* (2013). URL : `http://www.cv-foundation.org/openaccess/content_cvpr_2013/papers/Wu_Facial_Feature_Tracking_2013_CVPR_paper.pdf` (cf. p. 43).