Machine à Vecteurs Supports Multi-Noyau pour la détection de points caractéristiques du visage

Vincent Rapp[1], Thibaud Senechal[1], Kevin Bailly[1], Lionel Prevost[2]

[1]ISIR - CNRS UMR 7222
Université Pierre et Marie Curie, Paris
{rapp, senechal, bailly}@isir.upmc.fr

[2]LAMIA - EA 4540
Université des Antilles et de la Guyanne
lionel.prevost@univ-ag.fr

Résumé

Dans cet article, nous présentons une méthode robuste et précise pour détecter 17 points caractéristiques du visage sur des images expressives. Une nouvelle architecture multi-résolution basée sur les récents algorithmes multi-noyau est introduite. Les patches de faibles résolutions codent les informations globales du visage donnant lieu à une détection grossière mais robuste du point désiré. Les patches de grandes résolutions quant à eux utilisent les détails locaux afin d'affiner cette localisation. En combinant une détection indépendante de points et des informations *a priori* sur les distributions de points, nous proposons un détecteur robuste aux changements d'expressions ainsi qu'aux variations d'illuminations. Ce système a été évalué sur plusieurs bases de données de la littérature. Les résultats présentés améliorent les performances des détecteurs de l'état de l'art.

Mots-clefs

Analyse faciale, localisation de points caractéristiques, SVM, apprentissage multi-noyaux, multi-résolution

Abstract

In this paper we present a robust and accurate method to detect 17 facial landmarks in expressive face images. We introduce a new multi-resolution framework based on the recent multiple kernel algorithm. Low resolution patches carry the global information of the face and give a coarse but robust detection of the desired landmark. High resolution patches, using local details, refine this location. This process is combined with a bootstrap process and a statistical validation, both improving the system robustness. Combining independent point detection and prior knowledge on the point distribution, the proposed detector is robust to variable lighting conditions and facial expressions. This detector is tested on several databases and the results reported can be compared favorably with the current state of the art point detectors.

Keywords

Facial analysis, facial feature localization, SVM, multiple kernel learning

1 Introduction

La détection de points caractéristiques est une étape clef des systèmes d'analyse automatique du visage. (identification, reconnaissance, suivi de regard, etc.). Bien que plusieurs solutions aient été proposées, localiser des points caractéristiques d'un visage dans une image reste un problème non résolu pour des applications qui nécessitent d'opérer sur des visages présentant de fortes variations de poses, d'expressions, d'occultations, etc. Le développement de tels détecteurs est un compromis permanent entre robustesse et précision.

De manière générale, les méthodes de détection de points caractéristiques faciaux peuvent être classées en trois catégories :

- Les méthodes d'alignement, prenant en compte tous les points comme une forme globale et essayant de converger vers la forme la plus cohérente pour n'importe quel nouveau visage.
- Les méthodes par détection conjointe des points, détectant directement un modèle global.
- Les méthodes destinées à détecter les points indépendamment sans utiliser de modèle global.

Les méthodes d'alignement apprennent explicitement un modèle de visage et ont pour objectif de segmenter, lors d'un processus itératif, un visage d'une nouvelle image. Ces méthodes incluent les modèles actifs de formes (ASM pour Active Shape Model) [1], ou d'apparence (AAM pour Active Appearance Model) [2]. Plusieurs systèmes proposent des évolutions de ces méthodes d'alignement. Par exemple, Milborrow et Nicolls [3] apportent plusieurs améliorations aux ASM et les utilisent pour localiser des points caractéristiques dans des vues frontales de visages. Cristinacce et Cootes [4] utilisent quant à eux des représentations locales de la texture autour des points à l'aide d'analyse en composantes principales (ACP) et les combinent avec des ASM. Ces méthodes, largement utilisées pour l'analyse de visage, obtiennent de bons résultats. Mais elles présentent des problèmes de convergence vers des minimums locaux pouvant notamment découler de l'initialisation du modèle.

Les méthodes par détection conjointe des points permettent de détecter un ensemble de points via un modèle implicitement appris. Par exemple, [5] and [6] utilisent les niveaux

de gris des pixels comme entrées d'un réseau de neurones, ceci afin de détecter les points caractéristiques. Ainsi, la relation spatiale entre les points est implicitement apprise par le réseau de neurones. Ces méthodes utilisent un modèle contraint dont l'utilisation est limitée à certains cas. En effet, du fait de leur rigidité et de l'absence de corrélation entre la texture et l'apparence, ces modèles perdent de leur efficacité devant des problèmes importants de poses ou de déformations dans des applications réelles.

Les systèmes indépendants détectent chacun des points du visage indépendamment. Vukadinovic et Pantic [7] détectent 20 points faciaux en utilisant un classifieur GentleBoost appris sur des caractéristiques extraites avec des filtres de Gabor. Ces méthodes, de par l'absence de relation entre les points, peuvent introduire des valeurs aberrantes donnant lieu de gros problèmes de robustesse.

Récemment, des systèmes combinant ces deux types d'approches ont été proposée. Par exemple, les approches par modèles hiérarchiques (Pictorial Structure Matching, PSM) de Felzenszwalb et Huttenlocher [8] apprennent des détecteurs pour un ensemble de points manuellement labélisés et un arbre de données pour les relations spatiales entre les paires de points. Valstar *et al.* [9] quant à eux proposent une méthode basée sur des SVR (Support Vector Regression) pour détecter indépendamment chaque point tout en utilisant des Champs de Markov pour exploiter les relations entre ces points.

Cet article s'inscrit dans cette optique : combiner des détections indépendantes de chaque point avec des modèles explicites. Cette approche possède les avantages de chacune des méthodes présentées précédemment : l'utilisation d'un modèle contraignant les détections afin d'éviter les valeurs aberrantes, tout en évitant les problèmes d'initialisation ou de convergence des AAM. Nos contributions portent sur trois points :

1. Un nouveau détecteur de points caractéristiques utilisant à la fois des Machines à Vecteurs Supports et une validation statistique.
2. Une nouvelle méthodologie d'apprentissage, basée sur un apprentissage combiné à une procédure de bootstrap spécifique à notre problème, améliorant la capacité du système à généraliser.
3. Un nouveau détecteur multi-résolution utilisant les récents algorithmes multi-noyau pour la SVM.

La suite de cet article sera organisée de la manière suivante : la section 2 regroupe les différents éléments de notre méthode (extraction des caractéristiques, apprentissage des détecteurs et validation des détections). Dans la section 3, nous présenterons nos résultats expérimentaux sur plusieurs bases de données couramment utilisées dans la littérature. Enfin, la section 4 conclura cet article.

2 Méthodologie

Le système que nous présentons ici se décompose en deux étapes. Dans un premier temps, chaque point caractéristique du visage est détecté indépendamment. Dans un second temps, ces détections sont validées lors d'un processus de validation statistique.

Ce détecteur utilise un classifieur de type SVM qui apprend à discriminer un point caractéristique des autres zones de l'image. Pour caractériser l'information locale autour d'un point, nous utilisons les niveaux de gris du voisinage. Sur ces niveaux de gris, sont extraits des patches de plusieurs résolutions introduisant différents niveaux d'informations spatiales. La pondération de chacune de ces résolutions est apprise à l'aide des algorithmes multi-noyau appliqués aux machines à vecteurs supports, combinés avec une procédure de bootstrap adaptée à notre problème. En parallèle, la distribution des points du visage est modélisée par un mélange de gaussiennes sur l'ensemble des images de la base d'apprentissage.

En phase de test la SVM calcule, pour chacun des pixels candidats contenus dans une région d'intérêt, un score pouvant être interprété comme un indice de confiance sur la possibilité que le pixel corresponde au point recherché. On cherche ensuite la combinaison de points qui maximise les sorties des SVM.

2.1 Extraction des caractéristiques

Le détecteur de points caractéristiques du visage proposé utilise des patches de différentes tailles en entrée du classifieur. La taille de ces patches est un choix sensible. En effet, une taille trop petite ne prendra en compte que l'information locale (perdant donc des informations globales) et introduisant des détections fines mais ambiguës alors qu'une taille trop grande n'extraira que des informations grossières et peu précises.

Pour pallier à ce problème, nous proposons une architecture pyramidale de patches multi-résolutions extrayant différents niveaux d'informations. Pour un pixel i, nous prenons un premier patche p_i^1 de taille suffisamment grande pour capturer un maximum d'information globale. Les autres patches $(p_i^2, p_i^3, ..., p_i^N)$ sont quant à eux extraits en sélectionnant une zone de plus en plus petite, récupérant ainsi des informations de plus en plus locales et fines autour du point. Tous les patches sont ensuite re-dimensionnés à la taille du plus petit (fig. 2).

Ainsi, ce système sera capable, grâce à ces patches de petites tailles, de récupérer les informations locales et les petits détails, comme le canthus ou la position de la pupille. Les patches de grandes tailles se focaliseront sur l'extraction d'informations globales permettant ainsi de différencier un oeil droit d'un oeil gauche en utilisant par exemple la forme du nez ou des oreilles.

Nous proposons ici d'utiliser un détecteur par point. Etant dans un problème de classification, l'apprentissage est réalisé, pour chacun des points, à l'aide d'exemples positifs et négatifs. Nous extrayons 9 patches multi-résolutions comme exemples positifs (target) : le premier centré sur la vérité terrain et les 8 autres sur les 8 positions connexes. Pour les exemples négatifs (non-target) nous utilisons, au

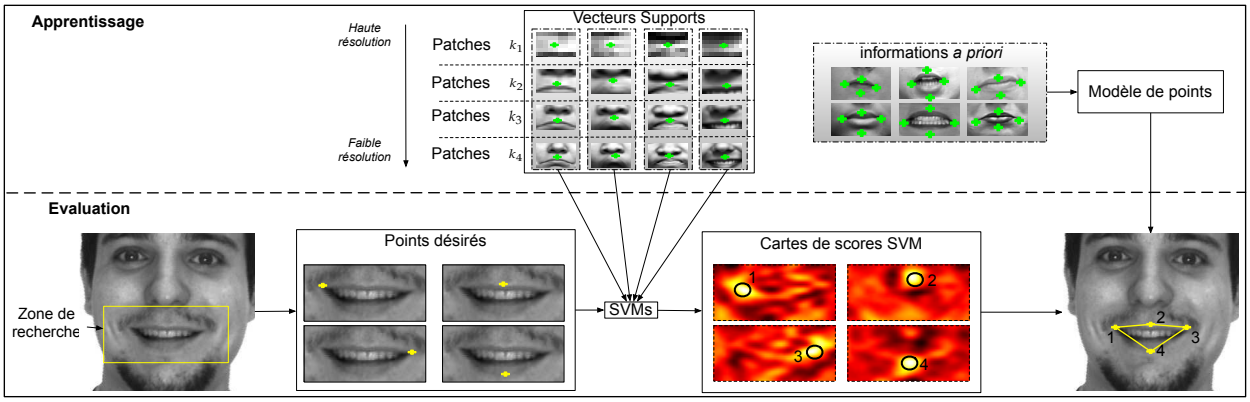

FIGURE 1 – Aperçu de la méthode proposée.

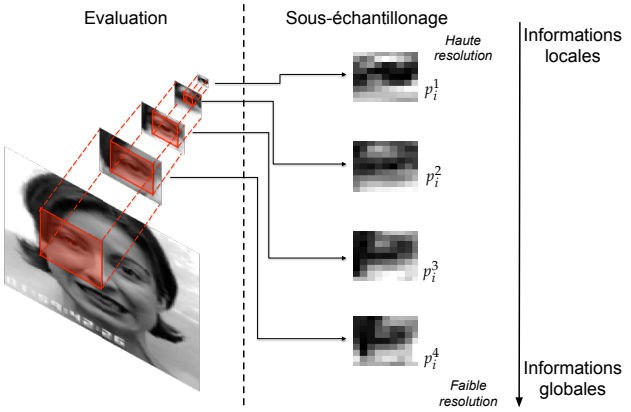

FIGURE 2 – Pyramide de patches multi-résolutions pour un pixel i. A chaque étape, différents niveaux d'informations sont mis en valeur.

départ, 16 patches multi-résolutions distribués uniformément autour de la vérité terrain (fig. 3).

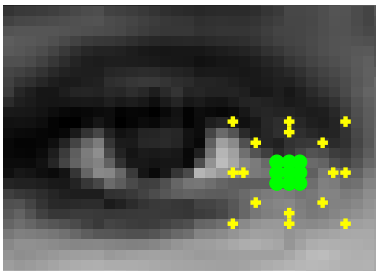

FIGURE 3 – Les exemples positifs sont extraits sur les 9 points les plus proches de la vérité-terrain. Les exemples négatifs sont ici représentés par des croix.

2.2 Apprentissage Multi-Noyau.

Afin de procéder à cette classification, nous utilisons les machines à vecteurs supports. Un avantage des SVM est que la modélisation du problème est toujours une optimisation convexe, de fait n'importe quelle solution locale est aussi un optimum global.

Phase d'apprentissage. Etant donné $X = (x_1, x_2, ..., x_m)$ un ensemble d'apprentissage de m exemples associés aux étiquettes $y_i \in \{-1, 1\}$ (positifs ou négatifs), la fonction de classification de la SVM associe un score s à un nouvel exemple (ou pixel candidat) $x = (p_i^1, ..., p_i^N)$ avec p_i les patches et N le nombre de résolutions :

$$s = \left(\sum_{i=1}^{m} \alpha_i k(x_i, x) + b \right) \quad (1)$$

Avec α_i la représentation dans l'espace dual du vecteur normal à l'hyperplan [11]. La fonction k est la fonction noyau résultant du produit scalaire dans l'espace des caractéristiques.
Dans le cas des SVM multi-noyau, le noyau k peut être toutes combinaisons convexes de fonctions semi-définies positives.

$$k(x_i, x) = \sum_{j=1}^{K} \beta_j k_j \text{ avec } \beta_j \geq 0, \sum_{j=1}^{K} \beta_j = 1 \quad (2)$$

Dans notre cas, nous avons une fonction noyau pour chaque résolution :

$$k = \sum_{j=1}^{N} \beta_j k_j(p_i^j, p^j) \quad (3)$$

Les poids α_i et β_j sont fixés afin d'avoir un hyperplan optimal dans l'espace des caractéristiques induit par k. Cet hyperplan sépare les deux classes d'exemples et maximise la marge : la distance minimum d'un exemple à l'hyperplan. Il a été prouvé que ce problème d'optimisation est conjointement convexe en α_i et β_j [12]. De fait, il n'y a qu'un unique minimum global pouvant être trouvé efficacement. $\beta_1 ... \beta_N$ représentent les poids associés à chacune des résolutions. Ainsi, le système est capable de trouver la meilleure combinaison d'entrée (utilisant les différentes résolutions) maximisant la marge. Nous avons utilisé les al-

gorithmes SimpleMKL [13] pour l'apprentissage de nos SVM multi-noyau.

Il s'agit ici d'une nouvelle façon d'utiliser les apprentissages multi-noyau. Généralement, ils sont utilisés pour combiner différents types de fonctions noyaux, comme les fonctions à bases radiales ou polynomiales. Ici, nous utilisons ces algorithmes pour combiner l'information à différentes résolutions spatiales.

Phase d'évaluation. En phase d'évaluation, nous devons choisir, parmi tous les pixels de la zone de recherche, celui correspondant au point désiré. Dans un cas parfait, nous devrions avoir $s > 0$ quand le pixel est proche du point désiré, $s < 0$ sinon. Dans le cas réel, quand nous avons zéro ou plus d'un candidat avec un score positif, nous utilisons la valeur de s pour prendre notre décision. Ce score, donné par la SVM et correspondant à la distance à l'hyperplan, peut ainsi être vu comme un indice de confiance sur le fait que le pixel soit le point que l'on souhaite détecter ou non.

Nous extrayons les patches pyramidaux pour chaque pixel de la zone d'intérêt, cette dernier pouvant être la région entière de la détection de visage donnée par Viola-Jones. Cependant, dans le but de réduire les temps d'extraction des patches, nous utilisons deux régions d'intérêt : une pour les yeux et une pour la bouche. La position et la taille de ces régions sont déterminées statistiquement durant la phase d'apprentissage. Ainsi, ces ROI (régions d'intérêt) seront suffisamment grandes pour prendre en compte les variations de poses. Ensuite, les patches pyramidaux de chacun des pixels de la ROI sont passés dans le classifieur. Cela nous donne en sortie le score pour chacun des candidats. Une carte de score est donc créée durant cette procédure représentant, pour chacun des pixels, un indice de confiance sur l'appartenance du pixel aux exemples positifs (Fig. 1). Nous attendons de cette sortie que son maximum corresponde à la position du point recherché.

2.3 Bootstrap & Patches négatifs

Afin de proposer un détecteur robuste, les exemples négatifs doivent être les plus pertinents possibles. Comme expliqué dans la partie 2.1, ce détecteur est appliqué sur de grandes régions d'intérêt (une pour les yeux et une pour la bouche). Des patches pris aléatoirement dans ces régions ne seraient pas nécessairement représentatifs. C'est pourquoi nous utilisons une procédure de bootstrap ayant pour but l'ajout de fausses alarmes pertinentes dans la procédure d'apprentissage. Le système est donc itérativement ré-appris avec deux ensembles d'apprentissages mis à jour, contenant les fausses alarmes produites après que le processus de détection ait été exécuté.

La base d'apprentissage est divisée en trois bases différentes, et ce sans mélanger les sujets : A, B et C. A et B sont utilisés pour l'apprentissage, tandis que C est utilisé en cross-validation. Cette procédure de bootstrap a été implémentée de la façon suivante :

1. Apprentissage sur A et test B.
2. Récupération des fausses alarmes produites sur B et ajout de ces dernières dans l'ensemble des exemples négatifs de B.
3. Validation sur C. Si le score de détection n'augmente plus, aller à l'étape 5. Sinon, aller à l'étape 4.
4. Inverser A et B et aller à l'étape 1.
5. Concaténation des exemples positifs et négatifs de A et B et réalisation de l'apprentissage final.

Cette procédure originale de bootstrap ajoute des fausses alarmes à chaque itération sans mélanger les exemples de A et B. Cette stratégie permet de limiter les risques de sur apprentissage : les fausses alarmes détectées sur les visages de A n'étant ajoutées que dans A. Ces fausses alarmes forcent la SVM, lors de l'itération suivante, à affiner l'hyperplan optimal. De plus, cette procédure permet d'ajouter des patches pertinents aux exemples négatifs. Dans la fig.4, nous désirons détecter le coin intérieur de l'oeil droit, mais des fausses détections apparaissent : ces fausses alarmes sont ajoutés à l'ensemble d'exemple négatif de la même base de données. A la seconde itération, des fausses alarmes déjà détectées lors de l'itération précédente peuvent revenir . Ainsi, un nombre important de fausses alarmes, éventuellement redondantes, est sélectionné. Cette redondance pouvant permettre d'ajouter un poids supplémentaire sur les fausses alarmes significatives.

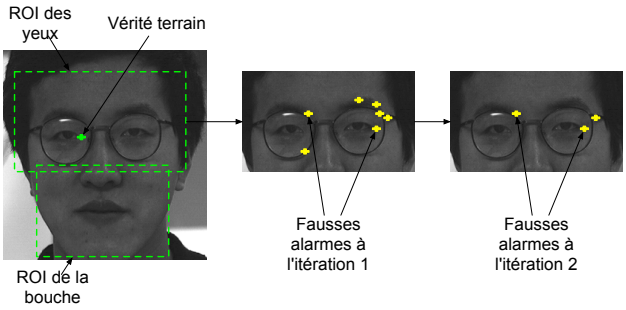

FIGURE 4 – Les fausses alarmes pertinentes sont ajoutées à l'ensemble de patche négatif à chaque itérations.

2.4 Modèle statistique & Validation

En sortie des SVMs, nous obtenons donc une carte de scores par point que l'on souhaite localiser. Les cartes de chacun des points étant calculées indépendamment les unes des autres, aucune relation spatiale particulière entre les points détectés n'est prise en compte. Par conséquent, des positions de points aberrantes peuvent éventuellement apparaître. Nous effectuons donc, en sortie des SVM, une étape de validation permettant de restreindre la distribution des points localisés à un ensemble de formes plausibles.

Estimation. Pour ce faire, nous divisons le modèle de distribution de points en plusieurs modèles locaux : oeil droit, oeil gauche et bouche. Ces modèles doivent être suffisamment flexibles pour gérer les variations d'expressions, de poses, de morphologies ou d'identités. Pour cela,

nous utilisons des mélanges de gaussiennes pour chaque modèle pouvant prendre en compte des distributions complexes [14]. Les modèles de mélange gaussiens (GMM) s'expriment comme une combinaison linéaire de K gaussiennes :

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k) \quad (4)$$

Ou $N(x|\mu_k, \Sigma_k)$ correspond à la fonction de densité de probabilité, ayant pour moyenne μ et pour covariance Σ. En supposant que suffisamment de composantes soient utilisées, un tel mélange de gaussiennes peut approximer toutes distributions arbitraires. Le but étant d'avoir un nombre restreint de composantes permettant d'avoir une "assez bonne" estimation. Nous utilisons l'algorithme EM pour adapter nos mélanges de gaussiennes à nos ensembles de données.

Evaluation. Pour notre problème, ces modèles doivent représenter des formes expressives empiriques. Nous utilisons 3 gaussiennes pour chacun des modèles : 3 pour l'oeil droit, 3 pour l'oeil gauche et 3 pour la bouche, afin de représenter les principales formes expressives (bouche ouverte, bouche ferme, ou sourire par exemple). Ensuite, nous procédons à la validation statistique en utilisant la distance de Mahalanobis d pour chaque modèle M :

$$d_k^M = \min_k[(x - \mu_k)\Sigma_k^{-1}(x - \mu)] \quad (5)$$

Avec x l'hypothèse (constitué de l'ensemble des points détectés) à valider. Nous voulons que d_k^M soit inférieure à un seuil T_k^M choisi de telle sorte que 95% des formes d'apprentissages soient acceptées. Lors de la phase d'évaluation, nous avons, pour chacun des points caractéristiques faciaux, un score SVM pour différentes localisations possibles. A partir de ces dernières, nous voulons trouver l'ensemble des pixels candidats formant une hypothèse validée par le modèle. Tout d'abord, nous testons l'hypothèse ayant la meilleure somme des scores de SVM. Si des points aberrants sont détectés par la SVM, d ne sera pas inférieure à T. On évalue alors la deuxième meilleure combinaison des sorties de SVM, et ainsi de suite jusqu'à trouver une hypothèse validant le modèle.

3 Experimentations

3.1 Bases d'apprentissage

Ce détecteur de points caractéristiques du visage à été entraîné à l'aide de deux bases différentes.

La base de données Cohn-Kanade [15] contenant 486 séquences d'images de personnes actant des expressions telles que la joie, la peur, la colère, etc. Les séquences d'images commencent par l'état neutre de l'expression pour finir avec le maximum de l'expression (apex). Pour notre apprentissage, nous utilisons la première image (image neutre) et la dernière image (image expressive à l'apex) de 209 exemples choisis aléatoirement parmi les 486 séquences.

La base PIE (Pose, Illumination, Expression) de la CMU [16] présente plus de 40 000 images de visages de 68 personnes. Chaque sujet a été capturé sous 13 différentes poses, 43 différentes illuminations, et 4 expressions différentes. Pour notre apprentissage nous avons utilisé 108 exemples, choisis aléatoirement.

Notre détecteur a été entrainé sur ces deux bases (317 visages au total, détectés à l'aide du détecteur de Viola-Jones). La distance inter-oculaire varie entre 40 et 50 pixels. Avant la première itération de bootstrap, nous avons 9 exemples positifs et 16 exemples négatifs par point, extraits à 4 résolutions différentes : 9x9 pixels, 17x17 pixels, 25x25 pixels, et enfin 33x33, et re-dimensionnés à 9x9 pixels. Le bootstrap est appliqué pour chaque point, de fait le nombre de fausses alarmes ajoutées peut varier selon les points. Néanmoins, nous avons pu évaluer un ajout moyen de 30% d'exemples négatifs.

3.2 Mesure de performances

Le succès de la détection se mesure en fonction de la distance entre les points détectés et leur vérité terrain, préalablement labellisées à la main. L'erreur d'un point i est définie comme la distance euclidienne d_i entre le point détecté (x_i, y_i) et sa vérité terrain $(x_i^{vt}, y_i vt)$. La moyenne est donnée suivant la formule :

$$m_e = \frac{1}{n d_{IOD}} \sum_{i=1}^{n} \sqrt{(x_i - x_i^{vt})^2 + (y_i - y_i vt)^2} \quad (6)$$

Avec d_{IOD} la distance inter-oculuaire et n le nombre total d'images. La détection est déclarée comme bonne si $m_e < 0.10$ (10% de d_{IOD}). Les évaluations des différents points de notre méthode sont présentés soit en erreur de localisation moyenne, soit en taux de bonne détection à 10%.

3.3 Evaluation Multi-Noyaux

La Fig. 5 étudie l'impact des patches multi-résolutions et de l'apprentissage multi-noyau sur la détection de points caractéristiques faciaux. Nous pouvons voir les cartes de décisions obtenues avec chacun des noyaux $k_1, ..., k_4$. L'étude de ces cartes nous montre que chacun des noyaux k_i extrait différents niveaux d'information. En effet, le noyau k_1 traite le problème localement, donnant lieu à plusieurs zone de détections fines mais ambiguës et peu robustes. A l'opposé, le noyau k_4 utilise des informations globales engendrant des détections beaucoup plus robustes mais aussi plus grossières. La combinaison de chacun des noyaux, à travers le processus d'apprentissage multi-noyau, nous donne une carte de décision finale utilisant tous ces différents niveaux d'information.

Du fait de l'utilisation d'un SVM par points que l'on veut détecter, l'apprentissage nous donne un ensemble de poids $\beta_1, \beta_2, \beta_3, \beta_4$ attribués à chacun des noyaux et donc à chacune des résolutions, et ce pour tous les points de notre

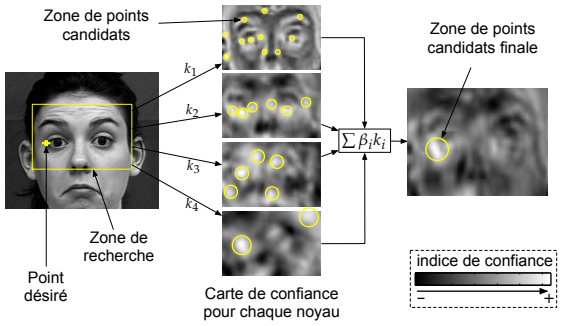

FIGURE 5 – Carte de décision de chacun des noyaux pour le coin extérieur de l'oeil droit.

modèle. Les moyennes des poids sont reportées dans le tableau 1

Modèle	β_1	β_2	β_3	β_4
Sourcils (6 pts)	0.1863	0.2774	0.1823	0.3541
Yeux (8 pts)	0.2140	0.2229	0.1654	0.3977
Bouche (4 pts)	0.1496	0.2639	0.2380	0.3484

TABLE 1 – Moyennes des poids associés à chacun des noyaux.

Nous pouvons voir que les patches de faible résolution (correspondant au noyau k_4) ont les β les plus élevés. Ceci signifie qu'un poids plus important est attribué, lors de l'apprentissage, aux informations globales. La zone de recherche étant assez grande, il est très important d'avoir une vue d'ensemble afin de pouvoir, par exemple, différencier les yeux droits des yeux gauches.
Ces patches de petites résolutions aident donc à trouver une localisation grossière du point dans la zone de recherche. Ensuite, les autres patches, utilisant des informations de plus en plus précises et locales, affinent cette localisation.

3.4 Fonction noyaux

Cette évaluation se concentre sur les résultats obtenus, en cross-validation, avec différents noyaux. Comme introduit dans la partie 2.2, une fonction noyau k peut être toutes combinaisons convexes de fonctions semi-définies positives (eq.2). Pour rappel, dans notre cas nous avons une même fonction noyau par type d'entrée (i.e pour chaque résolution r_1, r_2, r_3 et r_4) :

$$k = \beta_{r_1} k(p_j^{r_1}, p^{r_1}) + ... + \beta_{r_4} k(p_j^{r_4}, p^{r_4}) \qquad (7)$$

Nous avons évalué trois fonctions noyaux :

1. fonction linéaire : $k(x, x) = x'.x^t$
2. fonction RBF : $k(x, x') = exp(\frac{-\|x-x'\|}{2\sigma^2})$
3. fonction polynomial ordre n : $k(x, x') = (1 + \langle x, x' \rangle)^n$

Les résultats de cette étude sont reportés dans le tableau 2. L'utilisation de fonctions plus complexes qu'une simple fonction linéaire permet de diminuer le nombre de vecteurs supports. En effet, une fonction RBF ou polynomial projettera les données dans un espace de dimension plus grande qu'une fonction linéaire. De fait, les données seront plus facilement séparables et le nombre de vecteurs supports utilisés sera moindre. Concernant l'erreur moyenne, cette projection dans un espace de plus grande dimension permet également d'améliorer les résultats.

Fonctions noyaux	Linéaire	RBF	Polynomial n=5
Erreur moyenne	4.7%	4.3%	**3.9%**
Nb Vect. Supports	1579	851	**612**

TABLE 2 – Erreurs moyennes et nombres de vecteurs supports selon différentes fonctions noyaux.

3.5 Evaluation de la procédure de bootstrap.

Afin d'évaluer le processus de bootstrap ainsi que son impact sur les performances du système, nous avons effectué plusieurs tests. Lors du premier test, les exemples négatifs sont sélectionnés à l'aide de la procédure complète de bootstrap. Dans le deuxième test, les exemples négatifs sont choisis aléatoirement, sans aucune procédure de bootstrap. Nous avons aussi évalué les réponses du système avec des exemples ajoutés via une seule itération de bootstrap. Les résultats de chacun de ces tests sont reportés dans le tableau 3.

Méthode	Taux de bonne détection à 10%
MKL SVM + patches aléatoires	88%
MKL SVM + Bootstrap (1 it)	92%
MKL SVM + Bootstrap complet (6 it)	**97%**

TABLE 3 – Mesures de performance de différentes procédures de bootstrap.

Afin de réaliser un test pertinent, nous avons ajouté autant d'exemples aléatoires que le nombre d'exemples ajouté via la procédure de bootstrap. Cette procédure de bootstrap, adaptée à notre problème, permet un gain de performance assez conséquent. En effet, avec un ajout d'exemples négatifs aléatoirement sélectionnés nous obtenons 88% de bonnes détections. Une seule itération de notre bootstrap permet d'obtenir 92%. Enfin, lorsque la procédure complète est exécutée, nous améliorons les résultats de 5% pour obtenir 97% de bonnes détections. Ces évaluations ont été menées sur la base AR Face [17].

3.6 Résultats expérimentaux & Comparaisons

Plusieurs bases de données ont été utilisées pour évaluer ce détecteur. Tout d'abord, nous l'avons évalué sur une partie

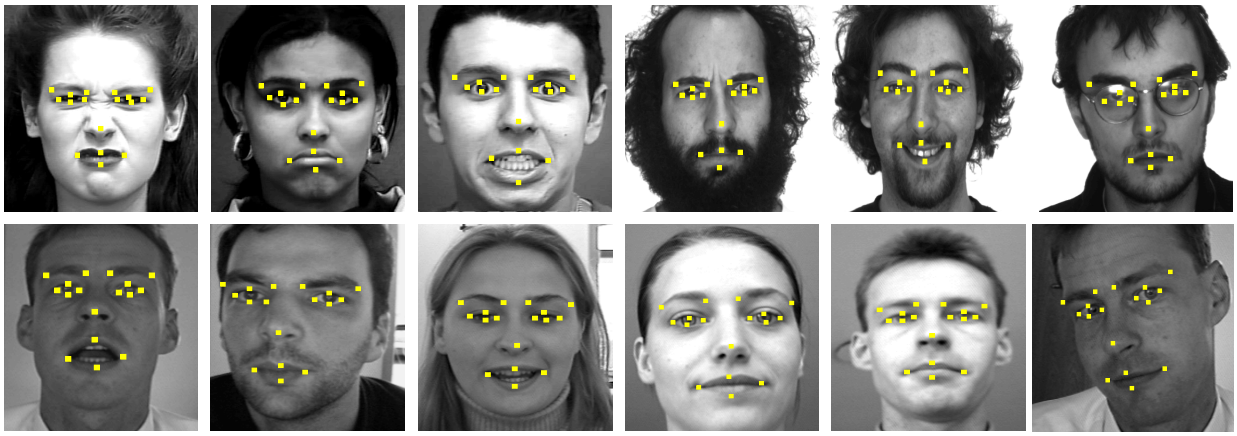

FIGURE 6 – Exemple de résultats sur les bases Cohn Kanade, AR Face et BioID.

test de la base Cohn-Kanade. Les résultats de cette évaluation sont présentés dans le tableau 4. Comme nous pouvons le voir à l'aide des scores de classification, tous les points de notre modèle sont détectés avec une grande précision, et ce même si la base de données inclut des images expressives. Seuls les points des sourcils ont un score plus faible. Ceci s'explique par le fait que la localisation manuelle de ces points n'est pas évidente. Suivant les annotateurs, le milieu du sourcil peut par exemple être placé à différentes positions (des tests réalisés par nos soins ont montré une erreur d'étiquetage de l'ordre de 6%).

Point	**C**	**m**	**e**
Coin extérieur de l'oeil droit	99.1%	4.0%	2.5%
Dessus de l'oeil droit	99.2%	3.9%	2.5%
Coin intérieur de l'oeil droit	99.3%	4.2%	2.8%
Dessous de l'oeil droit	99.5%	3.4%	2.4%
Coin extérieur du sourcil droit	89.4%	6.0%	5.1%
Coin extérieur du sourcil gauche	84.1%	7.1%	6.1%
Coin intérieur de l'oeil gauche	99.6%	5.0%	2.4%
Dessus de l'oeil gauche	100%	3.1%	2.5%
Coin extérieur de l'oeil gauche	100%	3.1%	2.4%
Dessous de l'oeil gauche	100%	4.5%	2.4%
Coin intérieur du sourcil gauche	84.5%	6.6%	6.5%
Coin extérieur du sourcil gauche	91.1%	5.0%	3.5%
Bout du nez	98.5%	4.0%	5.8%
Coin droit de la bouche	96.2%	3.8%	4.7%
Milieu de la lèvre supérieur	94.3%	4.9%	6.6%
Coin gauche de la bouche	95.5%	5.3%	4.5%
Milieu de la lèvre inférieur	95.0%	5.2%	4.2%

TABLE 4 – Résultats sur 266 images de la base Cohn-Kanade jamais vues en apprentissage. C correspond au pourcentage de bonne détection à 10% de la distance inter-oculaire. La moyenne m ainsi que l'écart type e sont mesurés en pourcentage de la distance inter-oculaire d_{IOD}

Deux autres bases de données ont aussi été utilisées pour évaluer, cette fois-ci, notre méthode en généralisation. La base AR Face [17] contenant des vues frontales de visage présentant différentes expressions faciales et illuminations. Nous avons aussi appliqué notre détecteur sur la base BioID [18] contenant 1521 vues frontales de visage présentant aussi plusieurs variations que ce soit au niveau de l'illumination, de la scène de fond, de la taille du visage ou encore de la variation de la pose de la tête. En se basant sur les résultats de la littérature [3, 4, 5, 9], cette base de données est considérée comme très difficile. De fait, cette base constitue une vraie base de comparaison avec l'état de l'art.

La figure 7 présente la distribution des erreurs cumulées mesurée sur ces deux bases de données. Nous pouvons voir que ce détecteur présente de bonne capacité de généralisation. Sur la base AR Face, nous pouvons observer que sur 80% des images de la base, notre détecteur présente une erreur inférieure à 5% de la distance inter-oculaire, soit une erreur inférieure à environ 2 pixels par points.

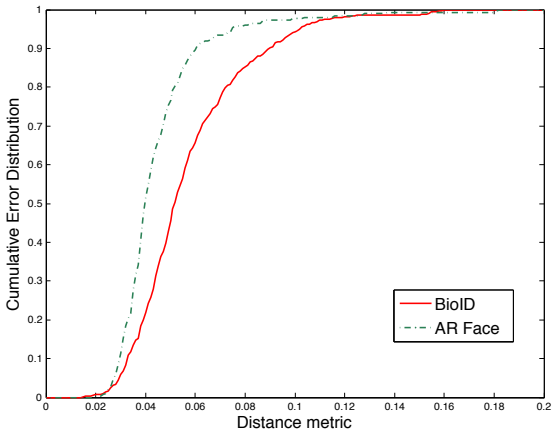

FIGURE 7 – Distribution des erreurs cumulées sur les bases de données BioID et AR Face.

Nous avons également comparé notre détecteur avec les détecteurs de l'état de l'art [2, 9, 4, 3]. Le tableau 5 montre l'erreur de chacun de ces détecteurs à 10% de la distance inter-oculaire. Nous pouvons voir que notre méthode se situe au niveau, voir au-dessus, de l'état de l'art actuel.

La figure 6 montre quelques résultats qualitatifs sur des

Méthode	Taux de bonne détection à 10%
AAM [2]	85%
BoRMaN [9]	95%
CLM [4]	90%
STASM [3]	95%
SVM multi-noyau	**95%**

TABLE 5 – Comparaison des différents systèmes de l'état de l'art sur la base BioID.

visages des bases de données Cohn-Kanade, AR Face et BioID

4 Conclusion

Dans cet article, nous avons présenté une méthode robuste et précise pour détecter de façon totalement automatique des points caractéristiques sur des visages expressifs.
Ce système se base sur l'exploitation d'informations à différentes résolutions. Des patches de petites résolutions permettent d'avoir la structure globale du visage, donnant lieu à des détections robustes mais peu précises. Les patches de grandes résolutions extraient quant à eux l'information dans une zone réduite du visage, amenant plusieurs zones de détections précises, dans l'une desquelles se trouve le point que l'on souhaite détecter. L'apprentissage multi-noyau nous permet de combiner ces différents niveaux d'informations de façon pertinente. Ce système est entrainé à l'aide d'une procédure originale de bootstrap. Les évaluations nous prouvent l'efficacité de ce bootstrap consistant à ajouter des exemples pertinents à notre base d'apprentissage. Enfin, en combinant les détections données par la SVM à une validation statistique, ce système corrige les exemples aberrants et assure des résultats robustes.
Ce détecteur, entrainé sur les bases Cohn-Kanade et PIE, est robuste aux variations d'illumination, aux expressions faciales et aux occultations causées par les lunettes ou les cheveux. Les évaluations effectuées sur les bases AR Face et BioID démontrent que ce détecteur est largement comparable, voire supérieur, à l'état de l'art.
La précision de notre méthode permet d'avoir une compréhension et une analyse des micro-mouvements faciaux. Nos travaux étant axés sur la détection d'émotions, nous planifions d'utiliser ce détecteur comme une première étape d'extraction de caractéristiques pertinentes pour cette problématique.

5 Remerciements

Ce travail a été partiellement supporté par l'Agence National de la Recherche (ANR) dans le cadre du programme de recherche technologique CONTINT (IMMEMO, projet numéro ANR-09-CORD-012).

Références

[1] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models-their training and application," *IEEE Conf. Comp. Vision and Pattern Recognition (CVPR'95)*, vol. 61, no. 1, p. 38, 1995.

[2] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *Proc. IEEE European Conference on Computer Vision (ECCV '98)*, p. 484, 1998.

[3] S. Milborrow and F. Nicolls, "Locating facial features with an extended active shape model," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR '08)*, p. 504, 2008.

[4] D. Cristinacce and T. Cootes, "Automatic feature localisation with constrained local models," *Pattern Recognition*, vol. 41, no. 10, pp. 3054–3067, 2008.

[5] T. Senechal, L. Prevost, and S. Hanif, "Neural Network Cascade for Facial Feature Localization," *Fourth Int'l Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR '10)*, p. 141, 2010.

[6] S. Duffner and C. Garcia, "A Connexionist Approach for Robust and Precise Facial Feature Detection in Complex Scenes," *Image and Signal Processing and Analysis*, 2005.

[7] D. Vukadinovic and M. Pantic, "Fully automatic facial feature point detection using Gabor feature based boosted classifiers," *Proc. IEEE Conf. Systems, Man and Cybernetics (SMC'05)*, vol. 2, p. 1692, 2005.

[8] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[9] M. Valstar, B. Martinez, X. Binefa, and M. Pantic, "Facial Point Detection using Boosted Regression and Graph Models," *IEEE Conf. Comp. Vision and Pattern Recognition (CVPR'10)*, 2010.

[10] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 57, no. 2, p. 137, 2002.

[11] B. Scholkopf and A. Smola, *Learning with kernels*. Cambridge, MIT Press, 2002.

[12] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan, "Learning the kernel matrix with semidefinite programming," *The Journal of Machine Learning Research*, vol. 5, p. 27, 2004.

[13] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *Journal of Machine Learning Research*, vol. 9, p. 2491, 2008.

[14] T. Cootes and C. Taylor, "A mixture model for representing shape variation," *Image and Vision Computing*, vol. 17, no. 8, pp. 567–573, 1999.

[15] T. Kanade, Y. Tian, and J. Cohn, "Comprehensive database for facial expression analysis," *Proc. IEEE Conf. Face and Gesture Recognition (FG'00)*, p. 46, 2000.

[16] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, p. 1615, 2003.

[17] A. Martinez and R. Benavente, "The AR face database," tech. rep., CVC Technical report, 1998.

[18] O. Jesorsky, K. Kirchberg, and R. Frischholz, "Robust face detection using the hausdorff distance," in *Audio-and Video-Based Biometric Person Authentication*, p. 90, 2001.