

Suivi d'expressions faciales occultées

Occluded facial expression tracking

Hugo Mercier¹

Julien Peyras²

Patrice Dalle¹

¹Institut de Recherche en Informatique de Toulouse
118, route de Narbonne, F-31062 Toulouse Cedex 9

²Dipartimento di Scienze dell'Informazione
via Comelico 39/41, I-20135 Milano

{mercier,peyras,dalle}@irit.fr

Résumé

Le travail présenté ici s'inscrit dans le domaine de l'analyse assistée d'expressions faciales observées sur des séquences vidéos de langue des signes. Dans ce contexte, le visage du signeur est fréquemment occulté, principalement par ses mains. Nous utilisons un modèle à apparence active (AAM) pour modéliser le visage et ses déformations expressives. Un algorithme d'adaptation robuste d'AAM est utilisé pour prendre en compte des données aberrantes. La contribution principale est la détection automatique des occultations manuelles, par une modélisation des images des résidus dans le cas non-occulté et une détection des occultations comme étant ce qui n'est pas expliqué par le modèle. Le modèle est choisi en fonction de l'itération actuelle de l'algorithme d'adaptation, ce qui améliore considérablement la détection d'occultations, comparé à l'utilisation d'un modèle fixé. Nous proposons de plus une stratégie de suivi robuste permettant de réinitialiser correctement l'AAM quand les occultations sont trop importantes.

Mots Clef

Modèle à apparence active, occultation, expressions faciales, suivi robuste

Abstract

The work presented here takes place in the field of computer aided analysis of facial expressions displayed in sign language videos. In this context, the signer's face is frequently occluded, mainly by hands. We use Active Appearance Models to model a face and its deformations caused by expressions. A robust AAM fitting algorithm is used to take outliers into consideration. Our main contribution is the automatic detection of hand occlusions, by means of a residual image model in the unoccluded case, and a detection of occlusions as being

what is not explained by this model. The model is chosen with respect to the fitting iteration, which greatly improves occlusion detection compared to the use of a fixed model. We also propose a robust tracking strategy to ensure a good AAM initialization when occlusions are too important.

Keywords

Active Appearance Model, occlusion, facial expression, robust tracking

1 Introduction

Nous utilisons le formalisme des modèles à apparence active (*Active Appearance Models – AAM*) pour modéliser un visage et ses déformations dues aux expressions, en termes de déformations d'un ensemble de points d'intérêt. Les points d'intérêts peuvent être suivis avec précision à condition que le visage ne soit pas occulté et ait été appris au préalable.

Nous nous intéressons ici à l'analyse de vidéos de langue de signes. En langue des signes, les expressions faciales jouent un rôle important et de nombreux signes sont produits près du visage du signeur. De plus, le crâne du signeur effectue fréquemment des rotations hors-plan. Ceci implique, du point de vue de l'interlocuteur (ici remplacé par un système de captation vidéo) que le visage n'est généralement pas vu en entier.

Des travaux récents se sont intéressés à des variantes robustes des algorithmes d'adaptation d'AAM ([1], [2]) capables de considérer des données aberrantes (*outliers*). Nous suivons l'approche proposée dans [3] de manière à détecter automatiquement la localisation des occultations, considérées comme données aberrantes pour l'algorithme. La principale contribution est de tirer avantage de la nature itérative de l'algorithme d'adaptation pour améliorer la détection.

Nous présentons en section 2 les modèles à apparence active et de quelle manière ils peuvent être utilisés pour suivre les déformations faciales avec un algorithme qui peut prendre les occultations en compte par le biais d’une carte de confiance des pixels. Dans la section 3, nous montrons comment calculer de manière optimale la carte de confiance des pixels pour détecter les occultations. La section 4 décrit une stratégie robuste permettant le suivi d’expressions faciales tout au long d’une vidéo.

2 Modèles à apparence active

Un modèle à apparence active (AAM) décrit un objet d’une classe prédéfinie comme étant une forme et une texture. Chaque objet, pour une classe donnée, peut être représenté par sa forme, à savoir un ensemble de coordonnées 2D d’un nombre fixé de points d’intérêt, et une texture, à savoir l’ensemble des pixels inclus dans l’enveloppe convexe de la forme.

La forme peut être décrite par :

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^n \mathbf{p}_i^s \mathbf{s}_i \quad (1)$$

où \mathbf{s}_0 est la forme moyenne, les \mathbf{s}_i sont des vecteurs de déformations et \mathbf{p}_i^s les coefficients pondérateurs de ces déformations. Ceci peut être écrit sous forme matricielle par $\mathbf{s} = \mathbf{s}_0 + \mathbf{S}\mathbf{p}^s$.

La texture est décrite par :

$$\mathbf{t} = \mathbf{t}_0 + \sum_{i=1}^m \mathbf{p}_i^t \mathbf{t}_i \quad (2)$$

ou, en notation matricielle $\mathbf{t} = \mathbf{t}_0 + \mathbf{T}\mathbf{p}^t$

Le modèle est construit à partir d’une base de visages où une forme, *i.e.*, les coordonnées 2D d’un ensemble de points d’intérêt, est associée à chaque image. Les formes sont extraites à partir de l’ensemble d’apprentissage et les déformations géométriques globales sont différenciées des déformations internes au moyen d’une analyse de Procrustes. Il en résulte une forme moyenne \mathbf{s}_0 et des formes qui diffèrent de la forme moyenne seulement par des déformations internes.

Les pixels inclus à l’intérieur de la forme de chaque visage sont extraits et transformés vers la forme moyenne, par une transformation affine par morceaux, de manière à construire la texture associée à chaque visage.

Une analyse en composantes principales (ACP) peut alors être appliquée aux formes et textures ainsi normalisées; les vecteurs propres forment alors les matrices \mathbf{S} et \mathbf{T} . Dans notre cas, nous retenons suffisamment de vecteurs pour expliquer 95% de la variance de forme et de texture (ce qui correspond à 12 vecteurs de forme et 15 vecteurs de texture).

Un visage (extrait de la base d’apprentissage ou proche) peut alors être représenté par des paramètres de forme \mathbf{p}^s et des paramètres de texture \mathbf{p}^t .

2.1 Algorithme d’adaptation robuste

Le but de l’algorithme d’adaptation d’AAM est de trouver \mathbf{p}^s et \mathbf{p}^t qui décrivent au mieux le visage vu sur une image d’entrée. Les paramètres de forme et de texture sont optimisés au moyen d’une image des résidus $E(\mathbf{x})$ qui représente les différences entre l’estimation actuelle du visage et le visage présent sur l’image d’entrée I :

$$E(\mathbf{x}) = \mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^m \mathbf{p}_i^t \mathbf{t}_i(\mathbf{x}) - I(W(\mathbf{x}; \mathbf{p}^s)), \forall \mathbf{x} \in \mathbf{s}_0 \quad (3)$$

La fonction $W(\mathbf{x}; \mathbf{p}^s)$ représente ici la transformation affine par morceaux qui projette les pixels de la forme moyenne \mathbf{s}_0 sur l’image I .

Au lieu de la norme euclidienne utilisée classiquement en optimisation, il est possible d’utiliser une distance pondérée :

$$\sum_{\mathbf{x}} Q(\mathbf{x}) E(\mathbf{x})^2$$

où $Q(\mathbf{x})$ pèse l’influence du pixel \mathbf{x} .

Nous utilisons le processus d’optimisation présenté dans [4], appelé algorithme à composition inverse (*inverse compositional algorithm*). Son avantage principal réside dans le fait que la matrice jacobienne peut être dérivée analytiquement, plutôt qu’estimée par différenciation numérique (comme dans [5]).

Parmi toutes les variantes proposées par les auteurs, nous avons choisi l’algorithme dit *simultané* (*simultaneous inverse compositional algorithm*). C’est une variante qui optimise à la fois les paramètres de forme et de texture de manière précise. Ce n’est pas la variante la plus efficace des algorithmes à composition inverse qui peuvent prendre en compte des variations de texture (voir par exemple l’algorithme *project-out* présenté dans [4]), mais celui donnant les résultats les plus précis.

La mise à jour itérative est donnée par (les détails de calcul sont présentés dans [1] et [6]) :

$$[\Delta \mathbf{p}^s, \Delta \mathbf{p}^t] = -H_Q^{-1} \sum_{\mathbf{x}} Q(\mathbf{x}) [G_s(\mathbf{x}), G_t(\mathbf{x})] E(\mathbf{x})$$

avec

$$\begin{aligned} G_s(\mathbf{x}) &= \left[(\nabla \mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^m \mathbf{p}_i^t \nabla \mathbf{t}_i(\mathbf{x})) \frac{\partial W}{\partial \mathbf{p}_1^s}, \right. \\ &\quad \dots, \\ &\quad \left. (\nabla \mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^m \mathbf{p}_i^t \nabla \mathbf{t}_i(\mathbf{x})) \frac{\partial W}{\partial \mathbf{p}_n^s} \right] \\ G_t(\mathbf{x}) &= [\mathbf{t}_1(\mathbf{x}), \dots, \mathbf{t}_m(\mathbf{x})] \\ H_Q &= \sum_{\mathbf{x}} Q(\mathbf{x}) [G_s(\mathbf{x}), G_t(\mathbf{x})]^T [G_s(\mathbf{x}), G_t(\mathbf{x})] \end{aligned}$$

Les paramètres de forme sont ensuite mis à jour par inversion et composition :

$$W(\mathbf{x}; \mathbf{p}^s) \leftarrow W(\mathbf{x}; \mathbf{p}^s) \circ W(\mathbf{x}; \Delta \mathbf{p}^s)^{-1}$$

Les paramètres de texture sont quant à eux mis à jour de manière additive par $\mathbf{p}^t \leftarrow \mathbf{p}^t + \Delta \mathbf{p}^t$. Les étapes essentielles de cet algorithme, appelé *simultané pondéré* sont présentées sur la figure 1.

Dans nos expérimentations, nous utilisons un AAM spécifique à une personne, indiquant que l'ensemble d'apprentissage ne contient des expressions que de la seule personne à étudier. Un AAM plus générique serait moins précis et plus difficile à contrôler.

3 Détection d'occultations

La carte de confiance $Q(\mathbf{x})$ utilisée dans la variante pondérée de l'algorithme d'adaptation d'AAM doit être aussi proche que possible de la carte des occultations. Si $M(\mathbf{x})$ est la carte des occultations réelles *i.e.*, une image binaire de même dimensions que I où chaque pixel vaut 1 si $I(\mathbf{x})$ est occulté et 0 sinon, alors la carte de confiance idéale à utiliser à chaque itération est $\mathbf{1} - M(W(\mathbf{x}; \mathbf{p}^s)), \forall \mathbf{x} \in \mathbf{s}_0$.

Le problème est de calculer la meilleure carte de confiance sans connaissance sur la localisation des occultations réelles. Nous proposons pour ce faire de modéliser le comportement de l'image des résidus dans le cas non-occulté et de détecter les occultations comme étant ce qui n'est pas bien expliqué par le modèle, suivant l'approche présentée dans [3].

3.1 Modèles paramétriques des résidus

Nous utilisons des modèles paramétriques de l'image des résidus. Nous proposons de tester différents calculs de la carte de confiance :

$$Q_1(\mathbf{x}) = \begin{cases} 1 & \text{if } \min(\mathbf{x}) \leq E(\mathbf{x}) \leq \max(\mathbf{x}) \\ 0 & \text{else} \end{cases}$$

$$Q_2(\mathbf{x}) = \frac{1}{\sigma(\mathbf{x})\sqrt{2\pi}} e^{\left(-\frac{E(\mathbf{x})^2}{2\sigma(\mathbf{x})^2}\right)}$$

$$Q_3(\mathbf{x}) = \begin{cases} 1 & \text{if } |E(\mathbf{x})| \leq 3\sigma(\mathbf{x}) \\ 0 & \text{else} \end{cases}$$

$$Q_4(\mathbf{x}) = \begin{cases} 1 & \text{if } |E(\mathbf{x})| \leq 4\sigma(\mathbf{x}) \\ 0 & \text{else} \end{cases}$$

$$Q_5(\mathbf{x}) = e^{\left(-\frac{E(\mathbf{x})^2}{2\sigma(\mathbf{x})^2}\right)}$$

Dans les fonctions ci-dessus, $\min(\mathbf{x})$ est la valeur minimale du pixel \mathbf{x} sur toutes les images des résidus, $\max(\mathbf{x})$ est la valeur maximale et $\sigma^2(\mathbf{x})$ la variance.

L'apprentissage des paramètres pourrait être effectué sur un ensemble quelconque d'images des résidus générés quand l'algorithme d'adaptation est utilisé sur des images non occultées.

Cependant, une image des résidus générée quand le modèle est éloigné de la solution est très différente d'une image générée quand le modèle est proche de la solution.

Ainsi les paramètres des modèles des résidus dépendent de la distance du modèle à la solution : ils doivent être permissifs quand le modèle est éloigné de la solution et stricts quand le modèle est proche de la solution.

3.2 Ensembles d'apprentissage partitionnés

De manière à expliciter le lien entre les paramètres et la distance à la solution, nous avons procédé au test suivant.

Un ensemble d'images des résidus est généré : l'algorithme d'adaptation d'AAM (non-pondéré) est lancé à partir des formes optimales perturbées pendant 15 itérations jusqu'à convergence. Pour initialiser l'AAM, les coordonnées de chaque point de la forme optimale sont perturbées par un bruit gaussien ayant 10 variances différentes (entre 5 et 30). L'algorithme est lancé sur 25 images qui font partie de l'ensemble d'apprentissage de l'AAM. La distance à la solution, calculée par la distance euclidienne moyenne du modèle de forme au modèle de forme optimal, et l'image des résidus sont stockées à chaque itération.

Au lieu de calculer les paramètres ($\min(\mathbf{x})$, $\max(\mathbf{x})$ et $\sigma(\mathbf{x})$) sur toutes les images des résidus (au nombre de $15 \times 10 \times 25 = 3750$), nous formons 15 partitions en regroupant les images des résidus qui ont mené à convergence par rapport à leur distance à la solution. Chaque partition P_i contient alors 210 images des résidus (pour un total de 3150 images des résidus) et peut être caractérisée par sa distance minimale d_i^- et maximale d_i^+ à la solution. Les paramètres sont alors appris, pour chaque pixel \mathbf{x} , sur les résidus de chaque partition.

Sur la figure 2 sont représentés les écart-types $\sigma(\mathbf{x})$ appris sur chacune des partitions. Pour des raisons de visualisation, seul l'écart-type moyen σ , calculé en moyennant sur l'ensemble des pixels \mathbf{x} , est affiché.

3.3 Approximation des paramètres

Quand l'algorithme d'adaptation est lancé sur une image de test, la distance du modèle à la solution est difficile à estimer. En effet, la seule information disponible est l'image des résidus qui peut donner une estimation de la distance à la solution seulement dans le cas non-occulté. Une telle information n'est pas fiable dans le cas occulté, puisque les résidus reflètent aussi

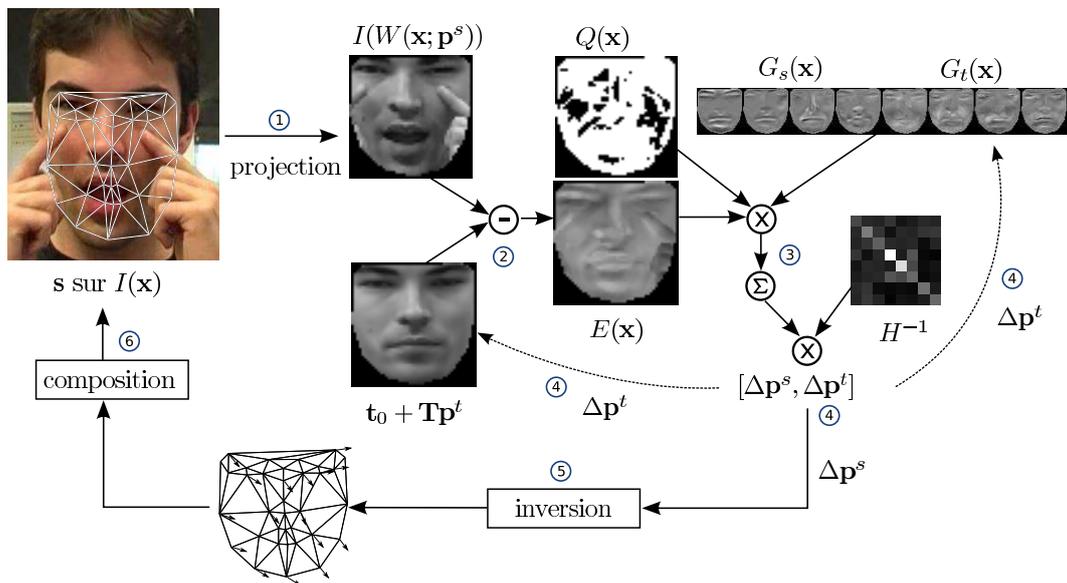


FIG. 1 – Etapes essentielles de l’algorithme simultané pondéré. Les nombres indiquent la chronologie des étapes pour une itération.

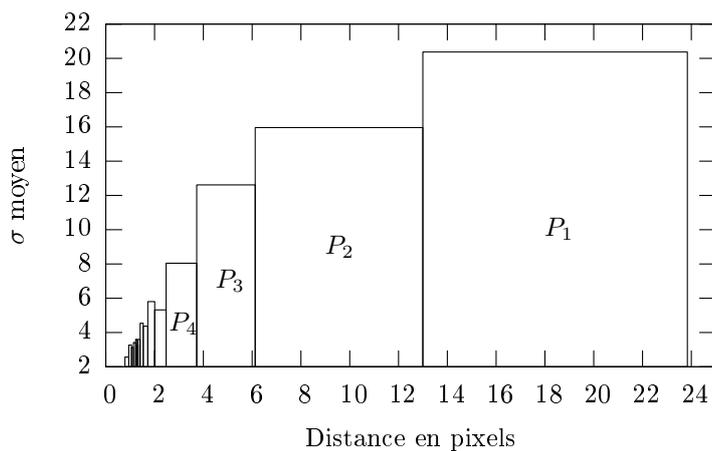


FIG. 2 – Ecart-type moyen appris pour chaque partition.

bien les erreurs de mauvais placement que les erreurs dues aux occultations.

Cependant, nous supposons que nous pouvons considérer le numéro d’itération de l’algorithme pour sélectionner la partition appropriée, en particulier si la distance du modèle à la solution dans le cas occulté est plus petite que la distance maximale utilisée pour regrouper les résidus de la première partition.

Pour valider cette hypothèse, nous avons procédé au test suivant. En utilisant les variances calculées sur chacune des 15 partitions, nous avons testé l’algorithme pondéré lancé pendant 20 itérations à partir de positions optimales perturbées par un bruit gaussien (avec une variance de 20) sur des images occultées (25% de l’image est couverte de blocs de 8×8 pixels d’intensité aléatoire). Il est à noter que les perturbations de la forme sont ici moins importantes que celles utilisées lors de la construction des partitions. Parmi toutes les fonctions $Q_i(\mathbf{x})$, nous utilisons $Q_3(\mathbf{x})$ pour calculer la carte de confiance à chaque itération. Un autre choix aurait pu être fait, puisque nous sommes seulement intéressé par la manière de calculer son paramètre, non par sa performance. Différentes manières de sélectionner la variance à chaque itération sont testées :

- S_{real} : sélection à partir de P_i où la distance réelle à la solution d_{model} est bornée par l’intervalle de distances de P_i : $[d_i^-, d_i^+]$; pour comparaison;
- S_{it} : sélection à partir de P_i où i est l’itération actuelle (et $i = 15$ pour les itérations 15 à 20);
- S_f : sélection à partir de P_1 ;
- S_m : sélection à partir de P_7 ;
- S_l : sélection à partir de P_{15} .

Les résultats sur la figure 3 montrent que le meilleur choix pour le calcul des paramètres du modèle des résidus est S_{real} . Cependant, il n’est pas utilisable dans la pratique (la forme optimale n’est pas connue *a priori*), mais il est possible de se rabattre sur S_{it} comme approximation. Pour comparaison, les résultats sont donnés pour le cas non-occulté et pour les variances fixes (S_f , S_m et S_l), qui donnent de mauvais résultats comparées à S_{real} ou S_{it} .

3.4 Choix du modèle paramétrique

Avec les résultats précédents, nous pouvons tester quelle est la meilleure façon de calculer la carte de confiance utilisée à chaque itération.

Pour ce faire, nous procédons au test suivant : la version pondérée de l’algorithme d’adaptation est lancée sur les images de la base d’apprentissage de l’AAM, couvertes avec un pourcentage variable d’occultations, depuis des formes perturbées par une gaussienne (nous utilisons une variance de 20 pour chaque coordonnée). Nous testons chacune des fonctions de calcul Q_i de la carte de confiance.

La fréquence de convergence est déterminée en cal-

culant le nombre d’adaptations qui résultent en une forme ayant une distance moyenne à la forme optimale inférieure à 2 pixels.

Les résultats sont résumés en FIG. 4. La fonction Q_4 montre clairement les meilleurs résultats. Toutes les autres fonctions donnent des résultats moindres, excepté pour la fonction Q_1 qui semble être un bon détecteur dans le cas d’un taux d’occultations faible et un très mauvais dans le cas d’un taux d’occultations important. La fonction Q_1 repose sur le calcul de valeurs minimales et maximales, qui sont des mesures très bruitées, comparées à la variance. C’est pourquoi le comportement de Q_1 n’est pas toujours fiable.

4 Stratégie de suivi robuste

Le but de l’algorithme de suivi est de prendre le plus possible en compte les occultations. Cependant, sur certaines images, les occultations sont trop importantes pour s’attendre à une bonne adaptation du modèle, parce que très peu d’informations fiables existent. Dans une telle situation, l’algorithme d’adaptation a généralement un comportement divergent résultant en une forme qui serait une mauvaise initialisation si elle était utilisée directement dans l’image suivante.

C’est pourquoi nous proposons d’utiliser une mesure de divergence et un AAM rigide pour initialiser le modèle.

Le but est d’éviter les mauvaises configurations du modèle de forme de manière à ne pas perturber le processus d’adaptation sur les images suivantes. Nous détectons ces mauvaises configurations comme étant celles mal expliquées par la statistique. Pour ce faire, nous comparons les paramètres de forme \mathbf{p}^s à leurs écarts types σ_i , qui ont été préalablement appris depuis la base d’apprentissage des formes. La divergence est décidée si :

$$\frac{1}{n} \sum_{i=1}^n \frac{|\mathbf{p}_i^s|}{\sigma_i} > \rho_1 \text{ ou } \max_{i=1, \dots, n} \left\{ \frac{|\mathbf{p}_i^s|}{\sigma_i} \right\} > \rho_2$$

Les seuils ρ_1 et ρ_2 sont déterminés empiriquement et peuvent être hauts (nous choisissons ici $\rho_1 = 2.5$ et $\rho_2 = 7.0$). Les seuils sont testés seulement après dix itérations, puisque les fortes déformations du modèle des premières itérations peuvent amener à convergence.

Sur chaque image, si la convergence est détectée, la configuration finale est stockée et elle sert comme initialisation pour l’image suivante.

Si une divergence est détectée, un modèle robuste est utilisé pour initialiser l’image suivante : un AAM construit en ne retenant que les vecteurs de déformation géométrique. Il s’agit d’un modèle représenté par la forme moyenne (et la texture moyenne) qui ne peut varier qu’en facteur d’échelle, rotation (dans le plan) et position mais pas en déformations faciales. Un tel

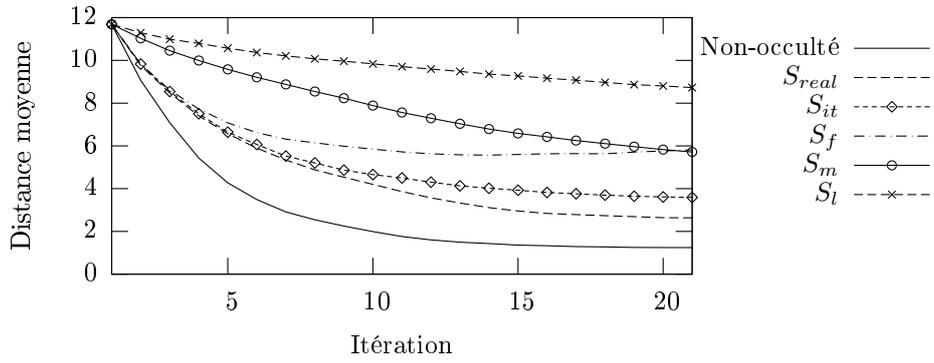


FIG. 3 – Comportement moyen de l’algorithme d’adaptation pour le cas non-occulté de référence et pour différents calculs de la variance dans le cas occulté.

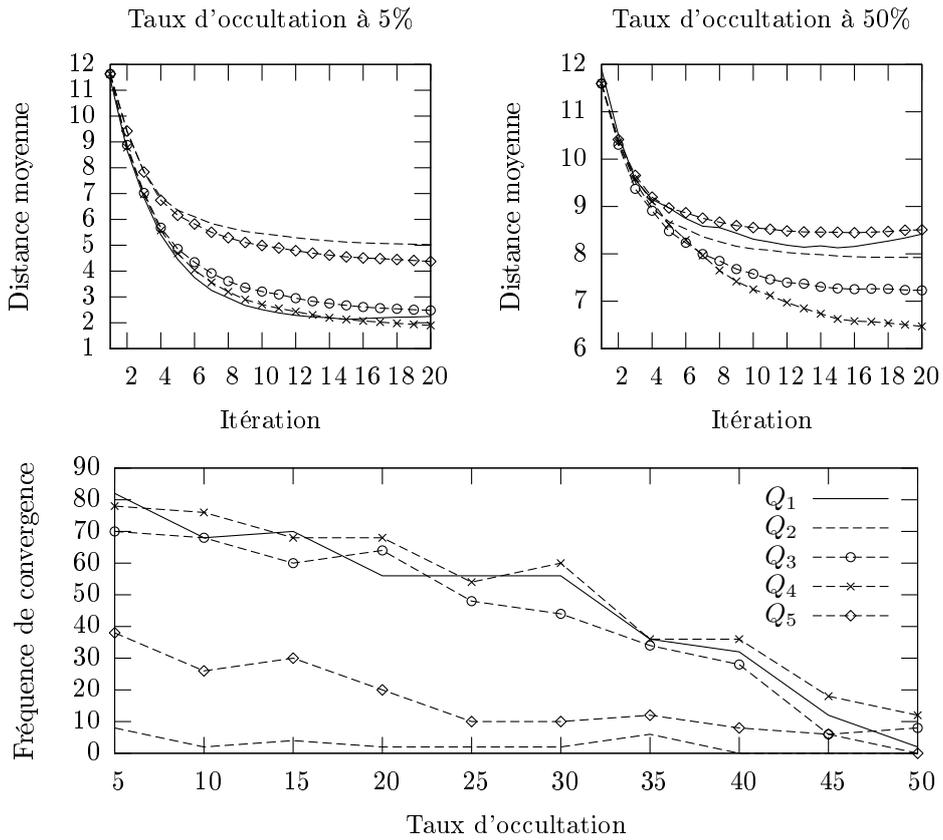


FIG. 4 – Caractérisation des calculs de la carte de confiance. La distance moyenne à la solution par itération pour 5% et 50% d’occultations (courbes du haut) et fréquence de convergence (courbe du bas).

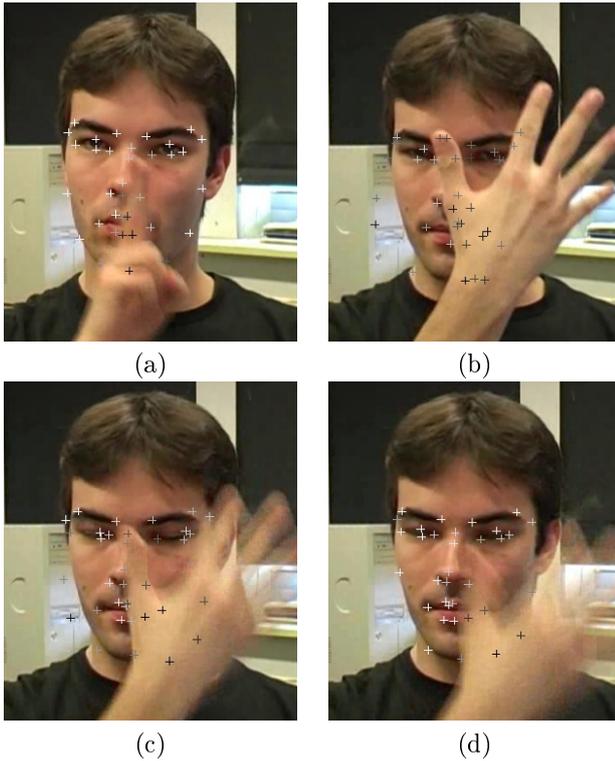


FIG. 5 – Résultats de suivi sur séquence vidéo. (a) Exemple d’une bonne détection d’occultations. (b) Exemple de divergence. Divergence sur une image (c) et convergence sur l’image suivante (d).

modèle donne une estimation de la forme du visage qui peut être utilisée comme initialisation pour l’AAM non-rigide. Il empêche le modèle non-rigide d’être attiré par des minima locaux (présents dans l’arrière-plan de l’image par exemple). L’adaptation du modèle rigide utilise aussi une carte de confiance pour traiter les occultations. Cependant, celle calculée pour l’AAM non-rigide est trop stricte pour le modèle rigide, c’est pourquoi nous utilisons une carte plus permissive (dont la variance est calculée sur la deuxième partition par exemple).

L’adaptation du modèle rigide est lancé pendant 5 itérations à partir de la dernière configuration obtenue après convergence. Le modèle non-rigide est ensuite lancé depuis la position résultante.

Nous avons testé cet algorithme de suivi sur une séquence vidéo d’environ 500 images où des signes viennent fréquemment occulter le visage du locuteur.

Quelques résultats typiques sont représentés en figure 5. Chaque point de la forme est affiché avec un niveau de gris calculé à partir de la carte de confiance. Comparé à un suivi naïf, l’AAM converge ici toujours quand il est lancé sur des images non occultées.

5 Conclusion

Nous avons présenté une manière de suivre les déformations faciales qui apparaissent sur une vidéo en prenant en compte les occultations manuelles via l’utilisation d’un AAM facial, d’une adaptation robuste qui pondère l’influence des pixels en présence d’occultations, d’une manière optimale de calculer la carte de confiance des pixels et d’une stratégie de suivi robuste basée sur une mesure de divergence et un AAM rigide. La séquence vidéo utilisée pour tester l’algorithme de suivi ne contenait que de faibles rotations hors-plan. C’est pourquoi l’AAM rigide 2D donne une bonne configuration initiale pour l’adaptation du modèle non-rigide. Sur des vidéos de langue des signes réalistes, les rotations hors-plan sont plus importantes et nous aurions dans ce cas à utiliser un modèle rigide capable de considérer la pose 3D. En particulier, ce cas peut être abordé en considérant un modèle rigide capable de se déformer selon une homographie du plan.

Nous avons utilisé le plus précis mais aussi le plus coûteux des algorithmes à composition inverse. Nous devons encore étudier l’influence sur la précision des approximations utilisées par exemple dans [1], [6] ou [3] qui permettent un suivi à la cadence vidéo.

Références

- [1] S. Baker, R. Gross, I. Matthews, and T. Ishikawa, “Lucas-Kanade 20 years on : A unifying framework : Part 2,” Tech. Rep. CMU-RI-TR-03-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, February 2003.
- [2] R. Gross, I. Matthews, and S. Baker, “Active appearance models with occlusion,” *Image and Vision Computing*, vol. 24, no. 6, pp. 593–604, 2006.
- [3] B.-J. Theobald, I. Matthews, and S. Baker, “Evaluating error functions for robust active appearance models,” in *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 149 – 154, April 2006.
- [4] I. Matthews and S. Baker, “Active appearance models revisited,” *International Journal of Computer Vision*, vol. 60, pp. 135 – 164, November 2004.
- [5] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 681–685, 2001.
- [6] S. Baker, R. Gross, and I. Matthews, “Lucas-Kanade 20 years on : A unifying framework : Part 3,” Tech. Rep. CMU-RI-TR-03-35, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, November 2003.