Capture d'Inscriptions pour la Réalité Augmentée

David Thevenin, François Bérard, Joëlle Coutaz

Laboratoire CLIPS-IMAG,
Equipe IIHM
BP 53 38041 Grenoble Cedex 9 France
{david.thevenin, francois.berard, joelle.coutaz}@imag.fr

RÉSUMÉ

Certains systèmes de Réalité Augmentée tels que le Bureau Digital nécessitent la capture d'inscriptions sur une surface plane physique (un tableau, une feuille de papier). La capture comprend l'acquisition d'images au moyen d'une caméra suivie de l'extraction à haute résolution des inscriptions. Les systèmes actuels de capture sont expérimentaux et fonctionnent dans des environnements contraints. Dans cet article, nous proposons un outil de capture pour de grandes surfaces utilisables comme support à des systèmes de Réalité Augmentée en environnement non contraint. Cet outil, appliqué au cas du Tableau Magique, inclut un algorithme de seuillage couplé à une technique de mosaicing. Tandis que le seuillage assure une extraction robuste des inscriptions, le mosaicing satisfait au besoin de haute résolution.

MOTS-CLES Réalité augmentée, Vision par ordinateur, Seuillage d'images, *clustering, mosaicing*, tableau électronique.

INTRODUCTION

Si les principes fondamentaux de l'Interaction Homme-Machine restent inchangés, l'évolution massive de la technologie force l'invention de nouveaux procédés. Réalisme, réalité et physicalité en sont les maîtres mots. L'amplification des objets qui nous sont familiers [7], les bits tangibles [6] et les interfaces palpables [4] en sont de bonnes illustrations. Tous ces procédés se fondent sur l'association de deux mondes, le réel et le virtuel, dont l'efficacité interactionnelle tient à l'existence de passerelles techniques bien pensées. Par exemple, dans le Bureau Digital [17], le passage du monde électronique au monde physique est assuré par un vidéo-projecteur. Dans l'autre sens, la passerelle est plus complexe à mettre à œuvre : il convient de capturer les inscriptions sur le papier et d'interpréter convenablement les gestes de l'utilisateur, que celui-ci emploie ses doigts ou les instruments du métier (crayon, gomme, pinceau).

Dans cet article, nous nous intéressons à la mise en œuvre d'une passerelle technique répondant aux contraintes interactionnelles d'un tableau augmenté, le Tableau Magique [1]. Comme le montre la figure 1, le Tableau Magique est constitué d'un tableau blanc réel manipulable de manière ordinaire : des inscriptions peuvent se faire et se défaire avec les stylos feutre et l'effaceur usuels. Un projecteur relié à la sortie vidéo d'un ordinateur affiche sur le tableau les retours d'information du système ; une caméra observe et interprète les gestes humains. Dès lors, le tableau est amplifié de services informatiques tels que couper, coller, déformer, imprimer, qui peuvent s'exprimer avec les mains. Dans ce projet, le tableau magique est envisagé pour une situation particulière d'usage : le "brainstorming", activité de réflexion collective conduisant à la production d'idées et à leur organisation. Cette activité doit être conduite en local ou bien à distance sur deux sites distincts.

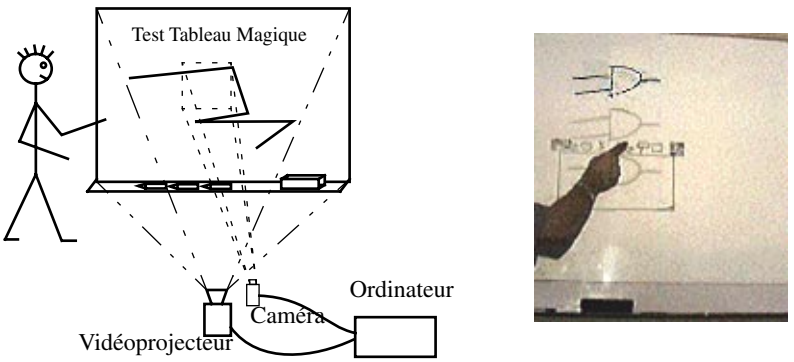


Figure 1: Le Tableau Magique. A gauche, l'équipement. A droite, une version Magicien d'Oz en action.

Cet article traite de l'acquisition des inscriptions physiques sur le tableau en condition naturelle : l'éclairage varie, les utilisateurs masquent temporairement la surface, le tableau présente des reflets et des salissures, les marqueurs sont de différentes couleurs et plus ou moins usagés. Dans la section qui suit, nous présentons brièvement le problème de l'acquisition et sa décomposition en deux classes de traitements complémentaires : le seuillage et le mosaicing. Le détail de chacun de ces traitements est ensuite traité.

LE PROBLÈME ET SA DÉCOMPOSITION

La caméra acquiert les images de la scène réelle. L'œil humain y distingue sans difficulté le contenu informationnel inscrit au feutre sur le tableau, divers instruments comme l'effaceur, les mouvements d'individus, tout ou partie d'une personne (un visage, une main), les retours d'information du système (résultats d'un calcul, image d'une copie de zone du tableau, mise en évidence de zone sélectionnée), etc. Un système de vision artificielle doit être capable d'opérer ces mêmes distinctions.

Le problème revient à extraire les inscriptions produites au moyen des feutres (l'encre). Cette extraction, qui consiste à décider pour chaque pixel s'il correspond à de l'encre ou au fond du support, s'appelle étiquetage. L'étiquetage peut être réalisé par la technique du *seuillage*.

L'extraction d'inscription doit être pratiquée sur toute la surface du tableau. Une image qui contiendrait l'ensemble du tableau ne présenterait pas nécessairement la résolution requise : pour un tableau de 1,5 m de large et une image caméra de 640x480 pixels, la résolution serait de 3 mm par pixel, soit juste la largeur d'un trait de feutre pour un pixel. A l'évidence, cette résolution ne satisfait pas les requis de qualité d'un service d'impression du contenu du tableau. Nous améliorons la résolution en photographiant le tableau par petits morceaux et en jouant sur le facteur de zoom de la caméra. Nous obtenons ainsi une suite d'images qu'il faut assembler pour reconstituer fidèlement le tableau. Ce traitement s'appelle *mosaicing*.

Les deux sections qui suivent présentent en détail la mise en œuvre technique du seuillage et du mosaicing.

SEUILLAGE

L'objectif visé par le seuillage est la production d'une image nettoyée (figure 2b) à partir d'une image brute (figure 2a). Pour cela, chaque pixel de l'image est étiqueté comme pixel d'inscriptions ou pixel de fond. Les pixels de fond sont ensuite supprimés.

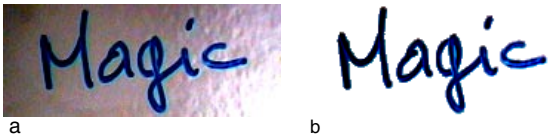


Figure 2: Image brute (a) et image nettoyée (b).

Principe général du seuillage. Lorsque l'éclairage de la surface est uniforme et que la surface est parfaitement plane, les pixels du fond ont en théorie la même valeur. Il suffit alors d'identifier cette valeur et de supprimer les pixels correspondants. Étant donné que la surface occupée par les inscriptions est largement inférieure à la surface occupée par le fond, la valeur de pixel de fond est choisie comme la valeur la plus fréquemment rencontrée dans l'image.

En pratique, les pixels de fond n'ont pas tout à fait la même valeur. L'histogramme de l'intensité lumineuse des pixels de la figure 3 le confirme : il exprime, pour chaque valeur de luminosité, le nombre de pixels présents dans l'image. On y observe deux bosses, l'une correspondant à l'encre, l'autre au fond. En supposant que le fond est clair (cas du tableau blanc ou de la feuille de papier) et que les inscriptions sont foncées (les feutres sont noirs, bleus, etc.), la valeur du seuil est comprise entre l'intensité lumineuse du fond la plus sombre et l'intensité des inscriptions les plus claires. En conséquence, tout pixel dont la valeur est supérieure au seuil est considéré comme appartenant au fond. Les autres pixels correspondent à l'encre. La difficulté revient au choix de la valeur du seuil.

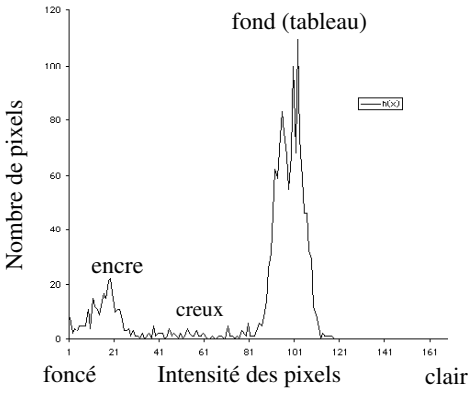


Figure 3: histogramme de l'intensité lumineuse des pixels. La bosse de gauche correspond aux inscriptions sombre, et la bosse de droite, aux fond clair du tableau. Entre les deux bosses, un creux. Le seuil doit être choisi dans ce creux.

Dans un environnement non contrôlé comme une salle de réunion, l'éclairage de la surface n'est pas uniforme. La figure 2a) illustre bien le phénomène : en bas à gauche, on note une partie surexposée en raison des reflets des

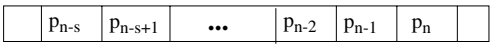
lampes de la pièce ou de la lumière du jour. D'autres facteurs interviennent dans la variation lumineuse comme l'irrégularité de la surface et les ombres des personnes et des objets de l'environnement. En raison de ces variations, le principe général de seuillage exposé jusqu'ici ne suffit pas à remplir sa mission : dans les zones fortement éclairées, les pixels d'encre peuvent avoir des intensités lumineuses supérieures aux pixels des zones du fond les plus sombres. Il convient donc de découper la surface en zones et de calculer un seuil adapté à chaque zone : c'est le seuillage adaptatif par zone.

Seuillage adaptatif par zone. Le problème du seuillage adaptatif d'image a été abondamment traité dans la littérature [10]. Parmi les méthodes avancées, le seuillage adaptatif par région convient à notre problème. Son principe s'appuie sur la continuité de la variation lumineuse dans l'espace. Il s'agit de partitionner la surface à seuiller en zones de taille assez petite pour que les variations de luminosité soient faibles à l'intérieur de chaque zone. Dans ces conditions, l'approche générale présentée précédemment peut être appliquée, le problème se ramenant à trouver la taille de chaque zone et la valeur du seuil pour chaque zone.

Dans ce qui suit, nous présentons les deux techniques que nous avons mises en œuvre. La première, basée sur la moyenne courante de l'intensité des pixels, a été conçue par Wellner pour le Bureau Digital [16]. Elle a été également utilisée dans le développement du système BrightBoard [14]. Les limitations de cette solution nous ont amenés à développer une nouvelle technique basée sur une modélisation plus précise de la distribution des intensités de pixels.

Moyenne courante

Principe. La méthode proposée par Wellner [16] consiste à calculer un seuil pour chaque pixel en fonction de la moyenne des pixels qui le précèdent sur une ligne de l'image. Soit la ligne de pixels p_{n-s} à p_n. Le seuil choisi

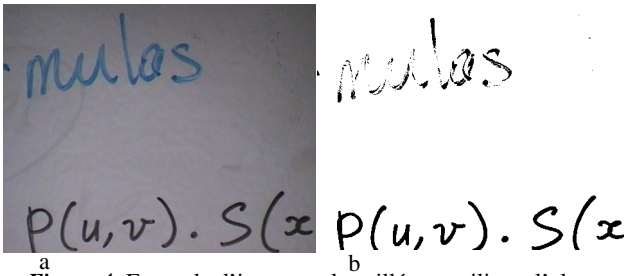


au point p_n est calculé en faisant la moyenne des intensités lumieuses des s pixels précédents. Si $p_n > seuil - (seuil * t)$, alors p_n est un pixel du fond. Dans le cas contraire, il s'agit d'un pixel d'encre. t est un pourcentage choisi empiriquement.

Evaluation de la moyenne courante. La technique de la moyenne courante est très rapide : sur un PowerMacintosh G3 à 400 Mhz, le seuillage d'une image de 640x480 pixels est de l'ordre du dixième de seconde. Par contre, la qualité du nettoyage n'est pas satisfaisante. La figure 4 en offre une illustration.

La technique de la moyenne courante s'avère trop simpliste pour répondre à la contrainte de robustesse :

1. Wellner ne suggère aucune heuristique pour le choix du seuil t qui doit être déterminé au cas par cas de manière expérimentale. Wellner utilise 15%, mais notre expérience révèle que t dépend de l'éclairage.
2. Le calcul de la moyenne peut inclure des pixels d'encre qui déplacent cette moyenne vers les intensités lumineuses foncées. Elle est donc malencontreusement influencée par de l'encre.



a b

Figure 4: Exemple d'image mal seuillée en utilisant l'algorithme de Wellner. Image initiale (a). Image seuillée (b). Le texte du bas est noir : il est bien extrait. Le texte du haut est bleu : il est mal extrait.

Il convient donc de déterminer de manière plus robuste le seuil et de tenir compte de l'influence de l'encre dans les statistiques. Dans ce but, nous proposons une nouvelle approche de seuillage basée sur une extraction de modes.

Extraction de modes

En statistique, un mode est une courbe gaussienne dont la surface est multipliée par un facteur d'échelle. Les deux bosses de la figure 3 sont modélisables chacunes par un mode. La technique que nous proposons en réponse aux limitations de la moyenne courante consiste à extraire ces modes.

Principe. Comme le montre l'histogramme de la figure 3, les pixels d'encre et les pixels de fond se regroupent chacun sous la forme d'une distribution gaussienne. Cette observation suggère d'approximer l'histogramme par une somme de gaussiennes (ou de modes). Le mode correspondant au fond est celui dont la moyenne des intensités lumineuses est la plus élevée (les pixels de fond sont les plus clairs de l'image). Disposant, par cette technique, d'une estimation précise de la moyenne μ_f et de la variance σ_f d'intensité du fond, nous définissons le seuil par :

$$s = \mu_f - 2 \cdot \sigma_f$$

Cette valeur de seuil permet d'extraire 98% des pixels du fond. Les descriptions mathématiques qui justifient ces choix sont fournies en annexe.

Cette approche donne de bons résultats dans le cas général, mais elle peut échouer. Par exemple, l'hypothèse de base du seuillage adaptatif par zone n'est pas respectée dans le cas d'une ombre franche : la zone contient une forte variation d'intensité lumineuse, quelle que soit sa taille. L'exemple de la figure 5 illustre ce problème : les pixels du fond se répartissent selon deux modes, fond clair et fond ombré.

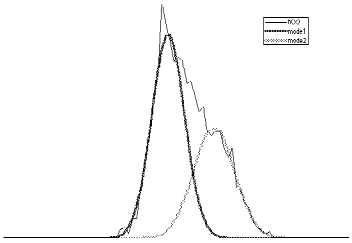


Figure 5: Cas d'une ombre franche. Les pixels de fond se répartissent selon deux modes.

La figure 6 illustre une seconde source d'échec : le phénomène d'interférences dû aux capteurs de la caméra à la frontière d'un trait noir. On observe l'apparition d'un mode dans les intensités lumineuses claires. Alors, le mode le plus clair n'est plus celui du fond.

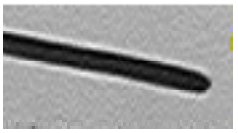


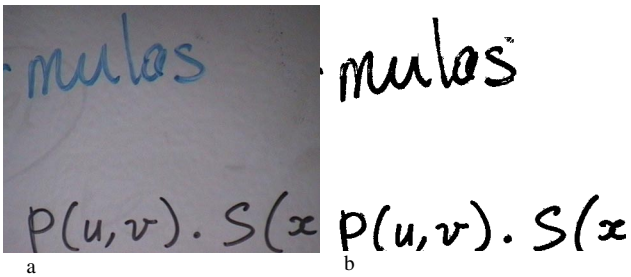
Figure 6: Interférence à la frontière d'un trait noir. Ce phénomène crée un mode dans l'intensité lumineuse claire.

Nous avons traité ces cas particuliers par des heuristiques:

- si les deux modes les plus lumineux sont très proches et qu'ils ont le même écart type à 10% près, alors ils représentent les pixels du fond,
- si l'histogramme contient deux modes lumineux proches ainsi que d'autres modes moins lumineux mais éloignés, alors les deux modes lumineux représentent les pixels du fond.

Evaluation de la technique d'extraction de modes. Notre expérience confirme la robustesse de notre approche dans des conditions très variables. Les cas traités correctement par la technique de la moyenne courante le sont également par notre technique. L'extraction de modes a l'avantage d'être autonome pour le choix du seuil et de traiter avec succès de nombreux cas d'échec rencontrés avec la technique par moyenne courante. A titre d'exemple, on comparera les résultats de notre technique présentés dans la figure 7 au regard du même exemple de la figure 4 avec la moyenne courante.

Notre approche n'est cependant pas parfaite. Elle est coûteuse en temps de calcul : le traitement d'une image



a b

Figure 7: Seuillage par extraction de modes : image brute (a) et image seuillée (b).

est dix fois moins rapide que la technique par moyenne courante (le traitement d'une image de taille 640x480 est de l'ordre de la seconde). La technique d'extraction échoue systématiquement lorsque la zone à traiter ne contient que des pixels d'inscription : l'unique mode de l'histogramme est considéré comme représentant le fond. Nous pensons nous affranchir de cette limitation par un algorithme de plus haut niveau d'abstraction qui applique une contrainte de continuité du seuil entre zones voisines.

Ceci conclut notre étude sur le seuillage adaptatif. Nous présentons maintenant le traitement permettant la constitution d'une image en haute résolution.

MOSAÏQUE D'IMAGES

Construire une mosaïque d'images d'une grande surface plane comme un tableau, consiste à capturer des sous-images zoomées du tableau, puis à les assembler. La capture automatique des sous-images implique de contrôler le mouvement et le facteur de zoom de la caméra. Nous présentons notre mise en œuvre du contrôle de la caméra puis une nouvelle approche de construction de la mosaïque fondée sur l'utilisation du vidéoprojecteur.

Contrôle de la caméra

Contrôler la caméra, c'est définir son point de vue. Cette opération consiste à commander les paramètres de la caméra : facteur de zoom, inclinaisons horizontale (*le pan*) et verticale (*le tilt*). La stabilité de la caméra (le pied de la caméra est fixe) et la précision du système de motorisation facilitent la mise en œuvre du contrôle. Celui-ci comprend deux phases : l'initialisation et l'estimation des paramètres de la caméra.

Initialisation. L'initialisation du système de contrôle consiste à créer deux tables de correspondance : l'une entre des points prédéfinis du tableau et les inclinaisons horizontale et verticale de la caméra ; l'autre entre la surface à acquérir et le facteur de zoom.

La première *table de correspondance point/inclinaison*, est construite en projetant, au moyen d'un vidéo-projecteur, un ensemble de points prédéfinis. Pour un point donné, la caméra est déplacée itérativement jusqu'à

ce qu'elle soit centrée sur lui. Ce faisant, nous obtenons les coordonnées *pan/tilt* de la caméra correspondant au point. L'opération est répétée pour chaque point. Le nombre de points nécessaires à ce calibrage dépend de la taille de la surface à traiter et de la précision recherchée. Pour un tableau de 1,5m x 1m, nous utilisons 64 points formant un quadrillage uniforme du tableau.

Les points projetés ont une surface de plusieurs pixels (5x5 pixels) de façon à améliorer la précision du centrage de la caméra en chaque point : le centre est le barycentre des pixels correspondant au point. La détection du point s'appuie sur une différence d'images, seuillage et analyse de connexité : une première image est acquise sans projection de point, la seconde avec le point projeté. La différence entre les deux images identifie le changement. Ce résultat est seuillé pour supprimer le bruit éventuel. L'analyse de connexité calcule les surfaces connexes de l'image seuillée. La plus grande de ces surfaces, dont on calcule le barycentre, correspond au point.

La deuxième *table de correspondance surface/zoom*, est déterminée à partir de la projection d'un carré dont la surface est connue et d'une succession de captures en utilisant, pour chacune, un facteur de zoom différent. Ainsi, en fonction de la surface connue du carré et de sa surface dans l'image capturée, nous construisons la table de facteur d'échelles. La détection du carré se fait comme pour les points de la première table (différence d'images, seuillage, connexité).

Estimationdes des paramètres de la caméra.
Soit une région de surface *S* centrée sur le point *p(x, y)* du tableau. Il revient au système d'estimer les paramètres de la caméra pour acquérir cette région. Cette estimation comprend deux étapes :

1. Interpolation de l'inclinaison de la caméra à partir des quatre points les plus proches de *p(x, y)* dans la table de correspondance point/inclinaison caméra.
2. Interpolation du zoom à partir de la table de correspondance surface/zoom caméra.

Grâce aux tables de correspondance, le système contrôle la caméra de manière autonome. La surface du tableau est couverte en appliquant de manière itérative l'algorithme de mises en correspondance. Pour un tableau de 1,5m x 1m, nous utilisons 9 zones. Il suffit maintenant d'assembler la suite d'images ainsi obtenues (en anglais, *mosaicing*)

Assemblage des images (ou *mosaicing*)

Le principe du *mosaicing* est d'assembler plusieurs images (les images sources) en une unique image (l'espace image cible). Pour deux images sources, l'assemblage se décompose en deux étapes : trouver la position d'une image par rapport à l'autre, puis corriger l'effet de déformation entre les deux images. Cette déformation résulte du changement de point de vue de la caméra lors des prises de vues. Ces deux étapes peuvent être résolues par une seule opération en calculant la transformation de chaque image par rapport à un espace image cible. C'est ce que nous détaillons maintenant.

La déformation d'une image prise par une caméra est de type perspective. Comme le montre la figure 8, un rectangle capturé par la caméra est déformé en un quadrilatère quelconque. La déformation perspective est modélisable par une transformation projective notée M. Chaque image caméra subit une transformation M_1. Pour un espace image cible donné, il faut calculer la transformation M_2 de l'image caméra vers l'espace image cible (c.f. figure 8). Dans notre cas, l'image cible est l'image du tableau entier projetée par le vidéo projecteur ; M_2 est donc l'inverse de M_1.

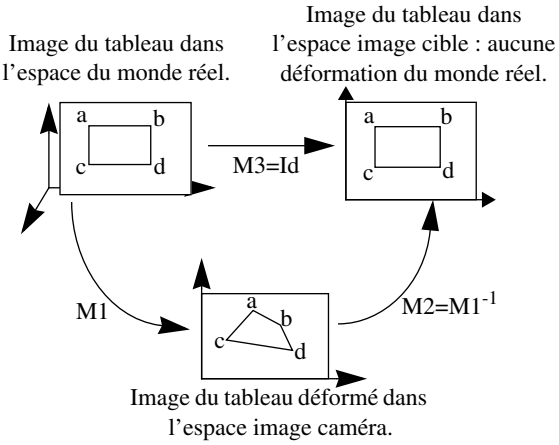


Figure 8: Transformation M1 de type projective.

Calcul de la déformation. La transformation M_1 est représentée par la matrice suivante [11] :

$$M_1 = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix}$$

Chaque point (x, y) de l'image caméra est projeté en (x', y') dans l'image cible par l'équation suivante :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad x' = \frac{a_1 x + a_2 y + a_3}{a_7 x + a_8 y + a_9} \qquad y' = \frac{a_4 x + a_5 y + a_6}{a_7 x + a_8 y + a_9}$$

Déterminer la matrice M_1 implique la résolution d'un système d'équations [[5] pp. 142]. La résolution de ce système à neuf inconnues, nécessite de déterminer au moins quatre points dans l'image source et leurs corre-

spondants dans l'image cible. C'est ce que nous détaillons au paragraphe suivant.

Correspondance entre points source et cible. Le principe est de déterminer à partir des coordonnées de n points connus de l'image source, les coordonnées des n points correspondants dans l'image cible. L'approche couramment rencontrée consiste à capturer des images consécutives qui se recouvrent partiellement (la partie A et B respectivement de la première et deuxième image, représentent la même scène c.f. figure 9).

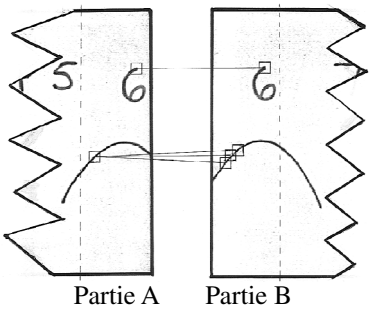


Figure 9: Correspondance par annalyse de pixels. Cet exemple, extrait de [11], illustre le principe de correspondance d'image. La partie droite de l'image (A) et la partie gauche de l'image (B) se superposent. L'algorithme détecte la correspondance du haut des deux "six" ainsi qu'une partie de la courbe.

La mise en correspondance de points entre deux images est un problème difficile en raison du manque de fiabilité de la détection des points [11], [12]. Nous proposons une solution spécifique mais fiable qui tire avantage de l'existance du vidéoprojecteur.

Notre approche consiste à projeter une mire de 4 points connus non alignés (les points de l'image cible), de capturer une image de cette mire et d'en extraire les points correspondants (les points de l'image source). Les points de l'image caméra sont extraits par différence d'images, seuillage et analyse de connexité (c.f. dans "Contrôle de la caméra", "Initialisation"). Cette technique nous permet de réaliser les quatre mises en correpondance nécessaires au calcul de la transformation. Connaissant la matrice de transformation, nous pouvons projeter les images sources vers l'image cible.

Projection des images. La projection peut être réalisée par l'une des deux techniques suivantes :

1. pour chaque pixel de l'image cible, on détermine son correspondant dans l'image source (ici, l'image caméra). Cette technique permet d'éviter des "trous" dans l'image cible.
2. chaque pixel de l'image source est projeté vers l'image cible.

Dans notre cas, seuls les pixels d'encre sont pertinents. C'est pourquoi nous préférons la deuxième technique qui permet de nous limiter à la projection des pixels d'encre. Cette approche a l'avantage de réduire le nombre de projections si l'image cible contient plus de pixels que la somme des pixels d'encre de toutes les images sources de la mosaïque. Cette hypothèse est vraie pour le Tableau Magique : en pratique, la surface couverte par l'encre est largement inférieure à celle du fond.

Sur un écran, un pixel de coordonnées (x,y) est représenté par une surface centrée sur le point (x,y). Projeter un pixel revient à projeter une surface. Or en règle générale, la surface projetée se retrouve à cheval sur quatre pixels. Il est donc nécessaire de distribuer l'intensité lumineuse de la surface projetée, sur les quatre pixels (c.f. figure 11), pour éviter un phénomène d'*aliasing* (c.f. figure 10b). Il s'agit de faire une interpolation. Dans notre cas, nous utilisons une adaptation de l'interpolation bilinéaire de la valeur du pixel [[2] pp. 115] (c.f. figure 10c).

Figure 10: Effet d'une interpolation sur les niveaux de gris lors d'une transformation géométrique d'une image. Fig a : objet initial; Figb : objet transformé sans interpolation; Fig c : transformation avec interpolation bilinéaire.

Le principe est le suivant : soit le pixel représenté par une surface centrée sur le point (x,y) et sa projection sur une surface centrée sur le point (x',y'). L'intensité lumineuse du pixel p(x,y) de l'image source est répartie sur le voisinage du point (x',y') dans l'image cible, comme l'illustre la figure 11. L'intensité lumineuse est répartie selon le rapport des distances entre le point *(x',y')* et le centre des surfaces des pixels avoisinants.

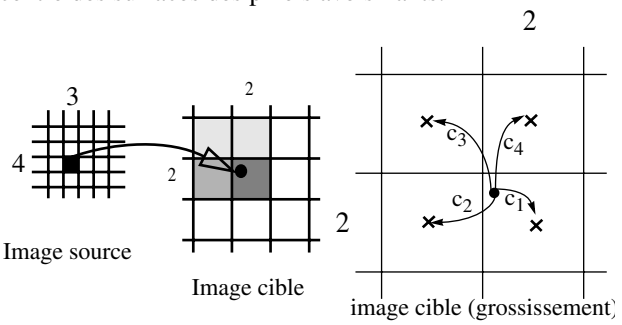


Figure 11: Le pixel (3, 4) se projette sur une surface centrée sur le point (1.7, 1.8) dans l'image cible. L'intensité lumineuse du pixel est répartie dans le voisinage du point (1.7, 1.8), c.-à-d. dans les pixels {(1,1), (1,2), (2,1), (2,2)}

PERSPECTIVES

Nos algorithmes, on l'a vu, utilisent l'intensité lumineuse, c'est-à-dire les niveaux de gris. Or la couleur faciliterait le seuillage pour l'extraction d'encre et permettrait de recoloriser l'image seuillée. Nous

projetons d'utiliser un modèle de couleur [8] pour détecter les feutres et trouver la couleur d'un trait.

Le seuillage de Wellner est très rapide mais manque de robustesse alors que notre technique, bien que plus robuste, est de l'ordre de dix fois plus lente. Nous pensons exploiter les caractérisiques de chacune de ces deux méthodes en fonction des services attendus par l'utilisateur. Par exemple pour une impression, attendre une seconde ou un centième de seconde importe peu mais une bonne qualité est nécessaire. Inversement, pour une opération de copier/coller, le système doit réagir rapidement. Selon les requis, le système utilisera l'algorithme qui convient : par exemple, un préseuillage à-la Wellner suivi de notre seuillage dès que le système dispose des ressources de calcul.

CONCLUSION

La capture de qualité d'une surface plane nécessite un seuillage pour supprimer tout parasite visuel et une acquisition d'images à haute résolution. Nous avons conçu et mis en œuvre une technique de seuillage adaptatif et une technique de *mosaicing* innovantes. La modélisation, par une somme de gaussiennes, de la distribution des pixels selon leurs intensités lumineuses, assure un seuillage robuste car indépendant des variations lumineuses sur la surface traitée. L'utilisation d'un vidéoprojecteur et d'une caméra motorisée couplée à un algorithme de contrôle de la caméra assure un mosaicing autonome, fiable et rapide. En intégrant le paramètre couleur et en utilisant notre seuillage ou celui de Wellner en fonction des besoins, nous espérons faire un pas vers la mise en œuvre d'un système de réalité augmentée, de type Tableau Magique, réellement utilisable.

REMERCIEMENTS

Nous remercions James L. Crowley pour les informations d'ordre technique qui nous ont aidés à l'aboutissement de ce travail. Nous remercions William Astier pour les travaux réalisés sur les nouvelles possibilités d'interaction du Tableau Magique, ayant ainsi attisé l'intérêt pour ce travail. Ce projet a été repris par Sébastien Annedouche, Benoît Loup, Michel Prodhomme. Nous les remercions pour leur aide et leurs apports techniques dans la réalisation technique d'un Tableau Magique utilisable.

ANNEXE

Modélisation par gaussienne

Soit $h(x)$ l'histogramme d'intensité lumineuse des pixels ($h(x)$ est le nombre de pixels dont l'intensité est égale à x). Notre objectif est de trouver une approximation $M(x)$ de $h(x)$, étant donné une somme de k modes :

$$M(x) = \sum_{j=1}^{k} m_j(x)$$

où chaque mode m_j est une gaussienne centrée sur μ_j, d'écart-type σ_j, et de facteur d'échelle (scale) s_i. Le facteur d'échelle modélise les surfaces respectives des modes : la somme des facteurs d'échelle est égale au nombre de pixels dans l'image.

$$m_j(x) = \frac{1}{\sigma_j\sqrt{2\pi}} \cdot e^{\frac{-(x-\mu_j)^2}{2\sigma_j^2}} \cdot s_j$$

Estimation du nombre de modes . L'algorithme est appliqué itérativement pour un mode, puis deux modes et ainsi de suite. L'algorithme s'arrête lorsque l'erreur de modélisation E est inférieure à un seuil donné.

$$E = \int (M(x) - h(x))^2$$

Estimation des centres et écarts types. Notre stratégie est de partitionner la contribution de chaque pixel entre les modes. La contribution $c_j(x)$ d'un pixel de niveau x à un mode m_j est donnée par sa probabilité d'appartenance à ce mode, pondérée par la somme de ses probabilités d'appartenance à tous les modes :

$$c_j(x) = \frac{P(x|m_j)}{\sum_{l=1}^{k} P(x|m_l)} = \frac{m_j(x)}{\sum_{l=1}^{k} m_l(x)}$$

Nous définissons également C_j, la somme des contributions au mode m_j de tous les pixels, c'est-à-dire la contribution au modèle du mode m_j :

$$C_j = \int (c_j(x) \cdot h(x))$$

Les calculs du centre et de l'écart type d'un mode m_j sont pondérés en fonction des contributions $c_j(x)$ et C_j. Les contributions à l'itération t sont utilisées pour déterminer le centre et l'écart type à l'itération $t+1$:

$$\mu_j^{t+1} = \frac{\int (x \cdot c_j^t(x) \cdot h(x))}{C_j^t}$$

$$\sigma_j^{t+1} = \sqrt{\frac{\int (x - \mu_j^{t+1})^2 \cdot c_j^t(x) \cdot h(x)}{C_j^t}}$$

Le processus est répété jusqu'à stabilisation des centres (lorsque la somme des variations des centres est inférieure à 1% de la somme des centres).

Estimation des facteurs d'échelle. Nous utilisons une propriété intéressante d'un mode en son centre :

$$m_j(\mu_j) = \frac{1}{\sigma_j\sqrt{2\pi}} \cdot s_j \qquad s_j = m_j(\mu_j) \cdot \sigma_j\sqrt{2\pi}$$

Il suffit d'estimer la valeur d'un mode j en son centre pour calculer le facteur d'échelle. Pour cela, la valeur de l'histogramme au centre du mode j est partitionnée selon la "demande" de chaque mode à cet endroit :

$$s_i^{t+0.5} = s_i^t \cdot \frac{h(\mu_i^t)}{\sum_{j=1}^{k} m_j^t(\mu_i^t)}$$

Dans une deuxième étape, chaque facteur d'échelle estimé à $t+0.5$ est pondéré par la somme de tous les facteurs d'échelle pour assurer que la somme des "demandes" ne dépasse pas le nombre n de pixels dans l'image.

$$s_j^{t+1} = \frac{s_j^{t+0.5}}{\sum_{i=1}^{k} s_i^{t+0.5}} \cdot n$$

Calcul du seuil. Le but du seuil est de supprimer le maximum de pixels appartenant au fond. Le fond est modélisé par un mode qui est une gaussienne multipliée par un facteur d'échelle. Supprimer un pourcentage n du fond revient à supprimer ce pourcentage de la surface du mode du fond, soit n% de la surface d'une gaussienne :

$$n = \int_k^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{-1}{2} \cdot x^2} dx$$

Pour k = 2, n = 0,98.

Ainsi en prenant le seuil $s = \mu_f - 2 \cdot \sigma_f$ nous supprimons 98% des pixels du fond.

RÉFÉRENCES

1. W. Astier, Manipulation d'information en Réalité Augmentée, DEA Systèmes d'Information I.M.A.G. & Diplôme Européen de IIIeme Cycle en Systèmes d'information MATIS, juin 1998.
2. K.R. Castleman, Digital Image Processing. Eds. Prentice Hall, 1996.
3. M. J. Black, A. Rangarajan, The Outlier Process: Unifying Line Processes and Robust Statistic. Dans IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94), Seattle, June 1994.
4. B. L. Harrison, K. P. Fishkin, A. Gujar, C. Mochon, R. Want, Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces. Proceeding de CHI'98, Los Angeles, Avril 1998.
5. R. Horaud, O. Monga, Vision par ordinateur : outils fondamentaux. Eds. Hermes, 1995.
6. H. Ishii et B. Ullmer, 1997, "Tangible Bits : Towards Seamless Interfaces between People, Bits and Atoms", CHI 97, Pub ACM, P 234-241
7. W. Mackay, G. Velay, K Carter, C. Ma et D. Pagani, 1993, "Augemnting Reality : Computational Dimensions to paper", Communication of the ACM n° 7, Juillet 1993, Pub ACM, P 96-97
8. N. Oliver, A.P. Pentland, F. Bérard, LAFTER: Lips and Face Real Time Tracker. Dans IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico
9. E. R. Pedersen, K. McCall, T. P. Moran, F. G. Halasz, Tivoli: An Electronic WHiteboard for Informal Workgroup Meetings. Proceedings of ACM INTERCHI'93.
10. P.K. Sahoo, S. Soltani, A.K.C. Wong and Y.C. Chen, A Survey of Thresholding Techniques. Dans Computer Vision, Graphics, and Image Processing, 1988, n°41, pp. 233-260.
11. E. Saund, Image Mosaicing and a Diagrammatic User Interface for an Office Whiteboard Scanner. Xerox Palo Alto Research Center. http://www.parc.xerox.com/spl/members/saund/zombieboard-public.html.
12. H.Y Shum, R. Szeliski, Panoramic Image Mosaics,. Microsoft Research Technical Report MSR-TR-97-23, http://www.research.microsoft.com/.
13. Q. Stafford-Fraser, Video-Augmented Environments Thèse de l'université de Cambridge, février 1996, 96 pages, http://www.uk.research.att.com/~qsf/thesis.pdf.
14. Q. Stafford-Fraser, P. Bobinson, BrightBoard: A Video-Augmented Environment. Proceeding of ACM CHI'96, Vancouver, BC Canada.
15. R. Szeliski, Video Mosaics for Virtual Environnements. In IEEE Computer Graphics And Application, March 1996, pp. 22-30.
16. P. D. Wellner, Adaptive Thresholding for the DigitalDesk. EuroPARC Technical Report EPC-93-110.
17. P. D. Wellner. Interacting with paper on the DigitalDesk. *Communication of the ACM n° 7*, p. 87-96, Juillet 1993.