Fusion Audio/Vidéo pour la reconnaissance d'objets et la navigation des robots mobiles

Loic Lachèze, Ryad Benosman, Yan Guo, Bruno Gas
Institut des Systèmes Intelligents et Robotique
Université Pierre et Marie Curie
4 place jussieu
75252 Paris cedex 05, France
ryad.benosman@upmc.fr

Résumé
La reconnaissance de formes et d'objets en robotique s'effectue principalement en utilisant la modalité vision. Il existe de nombreuses autres modalités possibles comme le toucher, l'audition, et les données proprioceptives. Celles-ci sont cependant rarement utilisées alors que leur utilisation apporterait des informations vitales qui faciliteraient la reconnaissance. Cet article présente une approche de fusion de données entre vision et audition, elle repose sur l'utilisation d'une caméra et d'un microphone qui observent une série d'objets mobiles et sonores. Nous discuterons à travers les tests expérimentaux sur une large base de données les effets de la fusion sur différents cas de problèmes usuels tels que les occlusions et les brouillages audio.
Mots-clé - Découpage en patch, reconnaissance de formes, fusion audio-video, sacs de mots

I. INTRODUCTION

L'intégration de plusieurs modalités sensorielles a été définie comme une étape nécessaire pour simplifier plusieurs taches de reconnaissance difficiles et permettre une perception robuste de l'environnement. [8]. La complémentarité des modalités simplifie plusieurs taches réputées difficiles dans une grande variété d'applications : vision [25], reconnaissance d'objects [7], localisation de robots [16], etc. Les travaux existants traitant du problème de la fusion de l'audio et de la vidéo sont généralement liés au domaine de l'interaction homme-machine [17], [23]. Certains systèmes existants combinent aussi les modalités audio et vidéo avec d'autres capteurs tels que lasers pour détecter des visages [13], audio pour localiser les sources sonores dans les scènes [5] parfois aussi en utilisant des réseaux de microphones [19], [20].

D'un point de vue théorique la plupart des méthodes de fusion existantes impliquent l'utilisation de l'information mutuelle entre modalités [14], [10], elles peuvent aussi être basées sur l'utilisation de techniques statistiques [15], [6]. Le but de ce travail est de porter l'attention sur le cas de la fusion audio-vidéo et d'étudier ce qui devrait être extrait d'images et de sons pour obtenir une fusion simple des deux modalités.

Dans ce qui suit l'extraction une méthode d'extraction de primitives images sera introduite, elle repose sur une approche originale qui découpe l'image par un processus itératif utilisant toute l'information contenue dans une image et non pas à quelques endroits saillants comme c'est souvent le cas dans ce type d'applications. Des expériences pratiques ont été menées sur une base de données d'objets en mouvement et pourvus d'une signature sonore. Ce dispositif expérimental permet alors d'étudier l'interaction possible entre l'image et le son, le point important étant de déterminer quels sont les éléments de corrélation entre modalités et quelles sont les échelles temporelles qui les lient les uns aux autres, et enfin dans quelle mesure ces différentes échelles temporelles définissent un élément d'interaction entre ces deux modalités. Les cas de fusion présentés considèrent aussi des scénarios d'altérations de la base de données par l'ajout d'occlusions visuelles et sonores.

II. PRÉSENTATION DE LA PRIMITIVE VISUELLE

L'objectif de cette section est de présenter une nouvelle méthode d'extraction de primitives visuelles qui représente l'une des contributions de ce travail. Son principe de fonctionnement repose sur la répartition de l'information dans l'image. Cette approche s'inspire des méthodes de division de l'image utilisées par la segmentation en région. L'objectif est de pouvoir décomposer des objets de manière identique quelles que soient les modifications de l'arrière plan de l'image.
Cette décomposition de l'image en patches est couplée à une description particulière des textures pour former le système visuel.
Le principe de son fonctionnement est le suivant :

- Division de l'image en zones ou sous-régions.
- Calcul pour chaque zone échantillonnée de sa description.
- Codification de l'image en fonction des labels de chaque mot reconnu.

Cette section est organisée de la manière suivante. Dans un premier temps la description des primitives est présentée. L'obtention de cette primitive s'inscrit dans le domaine des dictionnaires visuels et présente l'intérêt de fonctionner en temps réel tout en étant robuste aux changements d'illumination. Dans un second temps la décomposition de l'image est décrite. Elle s'inspire du découpage par Quadtree des méthodes de partage et fusion de région.

A. Représentation des textures

Le but étant d'implanter ce système visuel dans un robot, nous utiliserons, comme représentation de texture, un descripteur simple mais rapide à calculer à l'inverse des différentes approches utilisées pour définir habituellement les textons.

1) Principe: Le descripteur se base sur un calcul d'histogramme des différences entre les valeurs des pixels de l'image. Soit une image I, chaque valeur de cet histogramme de différence h_I est donné par :

$$h_I(i) = \sum_{x,y,x',y' \in I}^{x \neq x' \vee y \neq y'} diff(I, x, y, x', y', i), \;\; i \in [0, 255]$$

avec

$$diff(I, x, y, x', y', i) = \begin{cases} 1 & \text{si } |I(x,y) - I(x',y')| = i \\ 0 & \text{sinon} \end{cases}$$

Dans un second temps l'histogramme h_I est normalisé, pour assurer son invariance par rapport à la taille de I.

2) Algorithme de calcul: Pour une image de taille $n \times n$ le temps de calcul d'un histogramme des différences est de l'ordre de n^4 puisqu'il faut calculer la différence d'un pixel avec tout les autres pixels de l'image. Nous introduisons alors une optimisation permettant de diminuer de manière importante le temps de calcul. En effet il est possible de calculer l'histogramme de différence à partir de l'histogramme en niveau de gris de l'image. Soit h_d l'histogramme de différence et h_{int} l'histogramme de niveaux de gris :

$$h_d(i) = \sum_{j=0}^{255} \sum_{k=0}^{255} F(h_{int}(j), h_{int}(k), i)$$

Avec

$$F(h_{int}(j), h_inT(k), i) = \begin{cases} h_{int}(j) * h_{int}(k) & \text{si } |h_{int}(j) - h_{int}(k)| = i \text{ et } j \neq k \\ h_{int}(j) * (h_{int}(k) - 1) & \text{si } j = k \\ 0 & \text{sinon} \end{cases}$$

Cet algorithme permet de calculer la description de la texture à partir de l'histogramme de niveau de gris en un temps constant quel que soit la taille de l'image ou de la sous-image à traiter. La seule variabilité

en temps de calcul de l'algorithme vient de l'obtention de l'histogramme de niveau gris. Or son calcul est de l'ordre de n^2 pour une image de $n \times n$ pixels, ce qui est plus avantageux que le n^4 précédent.

B. Génération du dictionnaire visuel

L'espace des représentations de textures peut être partitionné en un ensemble de classes. Chaque classe est considérée comme un mot visuel et l'ensemble de ces mots comme le dictionnaire ou le vocabulaire. Pour générer ce dictionnaire, la méthode utilisée est la suivante.
Soit $F_z(I)$ une fonction permettant de décomposer une image I en un certain nombre de patches de texture z_i :

$$F_z(I) = z_0, z_1, ..., z_n \text{ avec } I = \bigcup_{i=0}^{n} z_i$$

Soit $T = h_{z_0}, h_{z_1}, ...h_{z_n}$ l'ensemble contenant toute les descriptions de textures des patches z_i de I. L'idée est d'échantillonner T pour réduire le nombre de descripteurs à $m \leq n$. Une fonction métrique exprimée par $dist(h_{z_i}, h_{z_j})$ est ajoutée à T, ainsi qu'une texture de référence h_{ref}. Cette référence est représentée par un patch ne contenant qu'une seule et même couleur, ce qui correspond à une surface colorimétrique uniforme. Dans un second temps toutes les représentations de patches contenues dans T sont comparées à h_{ref} et triées en fonction de leurs ressemblances. L'ensemble T_s correspondant à l'ensemble trié de T s'écrit :

$$T_s = h_{ref}, h'_{z_0}, h'_{z_1}, ...h'_{z_n} \text{ avec } dist(h_{ref}, h'_{z_i}) \leq dist(h_{ref}, h'_{z_j}) \text{ si } i < j$$

Un exemple de T_s est donné par la figure.1. Pour illustrer le fonctionnement de la génération du vocabulaire, un objet simple est considéré, l'image est découpée en carrés de tailles égales. Le classement des patches fait apparaître clairement que ces derniers sont triés en fonction de leurs textures. Une distance de mahalanobis est utilisée comme métrique et cela restera valable pour tout le reste de ce document. L'utilisation d'une seule distance et la projection de chaque représentation de texture sur un seul axe présente plusieurs avantages :

- Faciliter la création du vocabulaire, puisque les mots à choisir sont classés sur un même axe.
- Minimiser le temps de calcul pour apparier un mot du dictionnaire et une nouvelle primitive. Le calcul est direct puisqu'il consiste en une mesure de distance entre h_{ref} et la nouvelle primitive.
- Analyser les mots de vocabulaire obtenus en fonction du mot de référence. Ainsi si la primitive de référence utilisée représente une surface d'image plane, la primitive la plus éloignée devrait représenter une surface d'image très accidentée, comme l'illustre la figure.1.

Une fois triée, T_s contient toujours trop de descripteurs, il est alors échantillonné en m éléments de même taille. Pour chacun de ces éléments, seul le patch médian est sélectionné. Le résultat donne le vocabulaire visuel V :

$$V = h_{ref}, h'_{z_0}, h'_{z_1}, ...h'_{z_m},\ V \subset T_s$$

qui correspond à l'ensemble des patches les plus représentatifs. Il serait possible de considérer un échantillonnage plus complexe, qui extrait plus de patches représentatifs des parties de T_s où il y a beaucoup de patches et moins de celles qui sont vides, mais les résultats obtenus avec une telle approche se sont révélés moins satisfaisant globalement.

C. Codage de l'image en patches de V

Une image acquise I_{acq} est décomposée en z_{acq_i} patches. Chaque description calculée à partir de ses patches doit être comparée avec le contenu de V. Nous définissons alors une fonction $Reco$ qui transforme les patches de I_{acq} en patch du vocabulaire V :

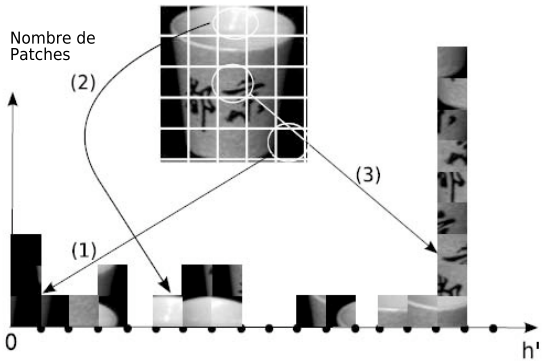

Fig. 1. Exemple de décomposition statique d'une image en patches. Les patches extraits sont comparés à un patch uniforme et triés en fonction de la complexité de la texture. Le patch de référence (uniforme) est placé en 0 sur l'axe des h'. La distance de mahalanobis est utilisée comme métrique pour la comparaison. Des exemples de patches du moins texturé (1) au plus texturé (3) sont présentés. Le patch (2) contient une réflexion lumineuse qui est assimilée à de la texture après la normalisation.

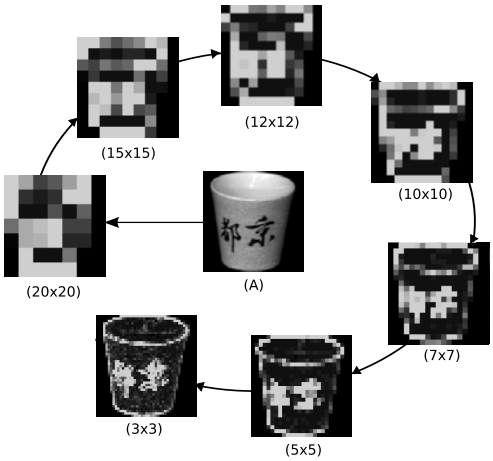

Fig. 2. Exemples de codification d'une même image (A) en fonction d'une taille variable de patches (de 20×20 pixels à 3×3 pixels).

$$Reco(z_{acq_i}, V) = j \text{ si } \exists j, h'_{z_j} \in V \ / \ dist(h'_{z_j}, h_{z_{acq_i}}) = min_k(dist(h'_{z_k}, h_{z_{acq_i}}))$$
$$\text{alors } h_{z_{acq_i}} = h'_{z_j} \text{ sinon } V = V \cup h_{z_{acq_i}}$$

Dans le cas où un nouveau patch est détecté, il est ajouté dans le dictionnaire comme une nouvelle entrée. L'image acquise I_{acq} est alors codifiée en utilisant les patches du vocabulaire. L'image résultante I_{cod_i} est donnée par :

$$I_{cod_i} = Reco(z_{acq_0}, V), Reco(z_{acq_1}, V), ..., Reco(z_{acq_m}, V)$$

La figure.2 présente différentes codifications dépendantes de la taille donnée aux patches. Plus la couleur du patch est proche du blanc, plus il est éloigné de h_{ref}. Il est intéressant de noter qu'il y a une relation directe entre la taille des patches et la qualité de la codification de l'image.

D. Autres descriptions utilisées

En plus de la description de textures présentée précédemment, il est possible d'adjoindre d'autres descripteurs simples, comme l'intensité moyenne des pixels du patch, les couleurs moyennes en rouge, vert et bleu, ainsi qu'une mesure d'entropie du contenu du patch.

E. Décomposition optimale

Le découpage dense d'images par des carrés de même taille 2 possède le défaut majeur de ne pas être robuste à la translation. Ce type de décomposition n'est pas unique et a pour conséquence de ne pas permettre de reconnaître des objets appris dans une image. Une décomposition efficace doit produire un partitionnement de l'image optimal et possiblement unique. De plus il pourrait être intéressant de produire moins de patches, mais de tailles différentes qui pourraient couvrir les zones homogènes de textures.

1) Principe: Pour réaliser cette génération optimale de patches, un algorithme de quadtree est utilisé. Ce type d'algorithme divise l'image en sous-images, puis celles-ci en d'autres sous-images de manière récursive. A partir de l'image initiale, chaque sous-image est divisée en 4 sous-images de même taille. L'idée de la décomposition optimale est de guider le découpage du QuadTree en fonction d'une mesure d'entropie. Le point de découpe est la position dans l'image qui minimise la différence de quantité d'information entre chaque sous-image. Cette approche doit permettre d'obtenir de grandes zones là où il y a peu de modifications de l'intensité des pixels et de nombreuses petites zones aux endroits qui sont caractérisés par de forts changements.

2) Algorithme de calcul du point de découpage: La mesure d'entropie sur une image I en niveau de gris s'écrit :

$$H(I) = -\sum_{c=0}^{c=255} P(I=c)\log P(I=c)$$

Où $P(c)$ est la probabilité d'apparition du niveau d'intensité c dans I. Dans notre cas cette mesure est légèrement différente et s'exprime par :

$$H'(I) = -\sum_{c=0}^{c=255} Occ(I=c)\log P(I=c)$$

Où $Occ(I=c)$ est le nombre de fois où un pixel de couleur c apparaît dans I. Ce changement se justifie par un découpage plus stable avec H' lorsque l'on translate un objet sur un fond uniforme. En effet H' conserve mieux le point de découpage sur l'objet que H (erreur de 2.9 pixels en moyenne pour H' cont 70.8 pixels pour H).

Pour estimer le point qui minimise la variance des différences des quantités d'informations contenus dans les quatre sous-images de I, le principe de l'image intégrale introduit par [12] est utilisé.

Soit $q(i,j)$ la quantité d'information du pixel $I(i,j)$ avec $q(i,j) = \log(P(I(i,j)))$.
La quantité d'information intégrale de $I(x,y)$ est définie comme

$$QI(x,y) = \sum_{i\leq x, j\leq y} q(i,j)$$

Cette somme est calculée en une itération sur l'image entière ou la sous-image considérée. Soit $R(x,y-1)$ la quantité d'information intégrale de l'image ou la sous-image de coordonnées $(0,0)$ à $(x,y-1)$. Le principe de calcul est donné par :

$$R(x,y-1) = QI(x,y-1) - Q(x-1,y-1)$$

Finalement la quantité d'information intégrale pour (x,y) est :

$$QI(x,y) = QI(x-1,y) + R(x,y-1) + q(i,j)$$

Une fois QI calculée, la valeur de la variance pour chaque pixel est implicite. Pour cela il est important de calculer la valeur moyenne de la quantité d'information contenue dans les quatre sous-images. Elle est donné directement par la quantité d'information intégrale de l'image ou la sous-image de taille $m \times n$ pixels divisée par 4 :

$$QI_{moy} = QI(m,n)/4$$

La figure.3 présente le calcul de la quantité d'information de chaque zone à partir de QI :

$$QI_{11}(x,y) = QI(x,y)$$
$$QI_{12}(x,y) = QI(m,y) - QI(x,y)$$
$$QI_{21}(x,y) = QI(x,n) - QI(x,y)$$
$$QI_{22}(x,y) = QI(m,n) - QI_{21} - QI_{12} - QI_{11}$$

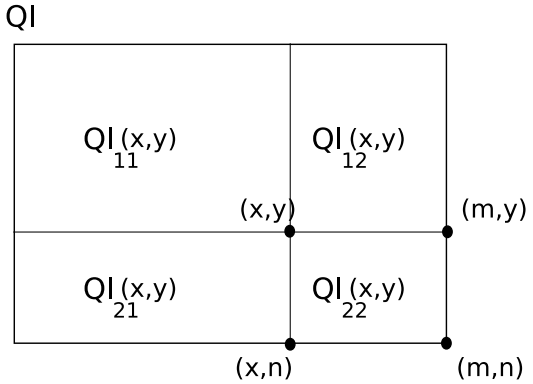

Fig. 3. Calcul de la quantité d'information contenue dans les quatres sous-images de QI, pour une position de découpage en (x, y).

Finalement la position optimale de découpage (x, y) est celle qui minimise la somme des différences suivante :

$$\exists(x,y)/min_{x,y}(\sum_{a=1,b=1}^{a=2,b=2}(QI_m - QI_{ab}(x,y))^2)$$

Une illustration de l'algorithme est donnée par la figure 4. Il apparaît clairement que la figure est décomposée de manière plus précise que le découpage de la figure.2. D'un point de vue visuel, si on compare les résultats de la figure.2 de l'image 10×10 et de la figure 4 de l'image (E), la méthode de découpage optimal fournit une décomposition plus précise grâce à la création de patches de taille spécifique. Il est intéressant de noter que les caractéristiques comme les contours et les coins apparaissent plus clairement.

Pour illustrer la stabilité de décomposition de cette approche, la figure.5 présente les patches obtenus pour un même objet mais translaté dans deux images. On peut vérifier la stabilité de la décomposition en notant que dans les deux cas les patches couvrent les mêmes zones en dépit de l'importante translation de l'objet.

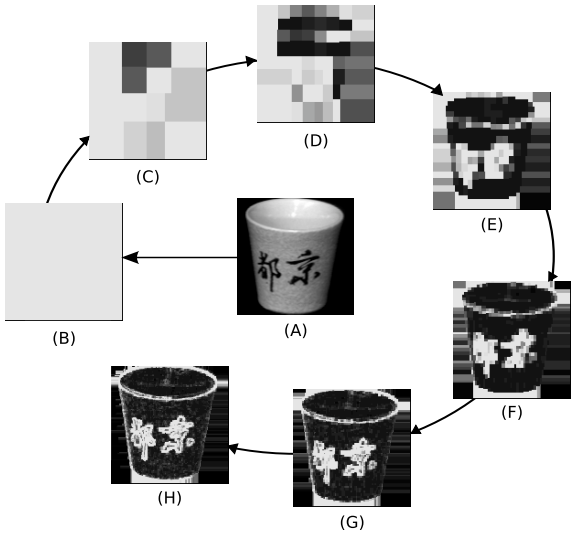

Fig. 4. Exemples de codification d'une même image (A) en fonction du nombre d'étage du quadtree guidé par mesure d'entropie (de l'étage 1 (B) à l'étage 8 (H)).

Fig. 5. Décomposition optimale d'un objet translaté sur deux étages. Les patches obtenus couvrent les mêmes régions et fournissent une même décomposition de l'image.

3) Gestion de l'arbre de décomposition d'une image: La construction de l'arbre de décomposition d'une image se fait simplement, la décomposition peut s'arrêter sous trois conditions différentes :

- Le patch à décomposer possède une entropie nulle, c'est à dire qu'il représente une surface de niveau de gris uniforme.
- Le patch à décomposer est trop petit pour pouvoir être à nouveau divisé. En effet il faut que la zone de l'image ait une certaine taille pour pouvoir calculer la description de l'image. Dans toute la suite, la décomposition s'arrête si le patch a sa largeur et/ou sa hauteur inférieure ou égale à 3 pixels.
- Il est aussi possible de fixer le nombre maximum d'étages de l'arbre.

Par la suite, le développement de l'arbre de décomposition est géré par ces trois conditions. Le nombre d'étages maximum est calculé en faisant l'hypothèse que chaque patch aura la même taille dans l'image. Ce nombre d'étage correspond à une taille moyenne des patches inférieure à 9 pixels. Ensuite pour chaque branche de l'arbre de décomposition les deux premières conditions sont testées pour savoir s'il est possible de continuer ce traitement.

III. Tests préliminaires de la décomposition optimale

La décomposition optimale est basée sur une mesure d'entropie de l'image. Ainsi toute modification de la répartition des pixels dans l'image et de leurs probabilités d'apparitions va influencer la mesure d'entropie. Ce changement, dans la mesure d'entropie, a pour conséquence de modifier la décomposition de l'image. Si elle est modifiée, alors elle risque de ne pas être suffisamment stable et robuste pour décrire des objets ou des lieux. Cette première série d'expériences a pour but de vérifier la stabilité de la décomposition en fonction des modifications possibles. Dans le cadre de la reconnaissance d'objets trois types de modifications de l'entropie d'une image ont été identifiés :

- Le déplacement d'une partie de l'image. Par exemple si un objet se déplace dans l'image, son changement de position va modifier la répartition spatiale des pixels et influencer la mesure de l'entropie.
- Le changement d'échelle. La différence a distance à laquelle un objet est observé, modifie la probabilité d'apparition des pixels sur toute l'image.
- La modification de l'arrière plan de l'image. Dans ce type de modification, l'objet reste identique, en échelle et en position, mais l'arrière plan de l'image est changé. Ce changement influence la décomposition, puisque la répartition des pixels est complètement modifiée.

Pour quantifier l'impact de ces modifications, une première partie expérimentale a été mise au point. Elle consiste à tester la conservation du vocabulaire, qui est utilisé pour coder un objet, lors de l'application de ces différentes modifications.

A. Présentation des expériences

Pour observer l'influence de ces trois modifications, nous voulons tester la capacité de la décomposition à coder un objet de la même manière. La codification d'un objet est donnée par la fréquence d'apparition des différents descripteurs de texture lors de sa décomposition. Pour vérifier que les mêmes descripteurs sont utilisés quelque soit les conditions de découpage, nous allons comparer la codification de l'objet sur un fond uniforme avec les codifications obtenues lors des trois modifications.

La codification d'un objet est donnée par l'histogramme de présence des descripteurs de textures. La codification de référence est donnée par une image contenant l'objet sur un fond uniforme. L'objet est segmenté et seuls les descripteurs contenus dans la boîte engobante de l'objet sont utilisés dans le calcul de l'histogramme. L'objet est ensuite placé sur un fond différent, déplacé, ou changé d'échelle et une nouvelle codification est obtenue de la même manière, c'est à dire en ne tenant compte que des descripteurs présents dans la boîte englobante. La comparaison des codifications est faite en utilisant l'intersection des histogrammes de présence des descripteurs. L'arbre de découpage est constitué de 5 étages et les patches du voisinage de l'objet de chaque étage participent à l'histogramme d'utilisation du vocabulaire. Ce vocabulaire contient 32 mots ou textures différent(e)s et est obtenu à partir d'un apprentissage préalable.

IV. Système audio

A. Prédiction et modélisation des frames audio

Les sons sont des signaux non stationnaires et fortement redondants qui requièrent donc d'être segmentés en trames successives pour ensuite être codés. Le codage permet de générer un ensemble de coefficients représentatifs du spectre court-terme. Celui que nous utilisons s'appuie sur le modèle de la cochlée de Patterson [24]. Dans le modèle de Patterson, la bande passante de chaque filtre cochléaire est décrite par une bande passante rectangulaire équivalente (Equivalent Rectangular Bandwidth (ERB)). Chaque filtre modélise le signal présent en sortie d'un nerf de la cochlée. Pour N filtres ERB nous obtenons N signaux. Ces signaux sont segmentés en trames successives sur lesquelles nous calculons une norme en valeur absolue :

$$\mathbf{x}_k = [x_1^k, x_2^k, \dots x_N^k] \text{ with } x^k_{i,i=1\dots N} = \sum_{q=1}^{L} |y_i^k(q)|$$

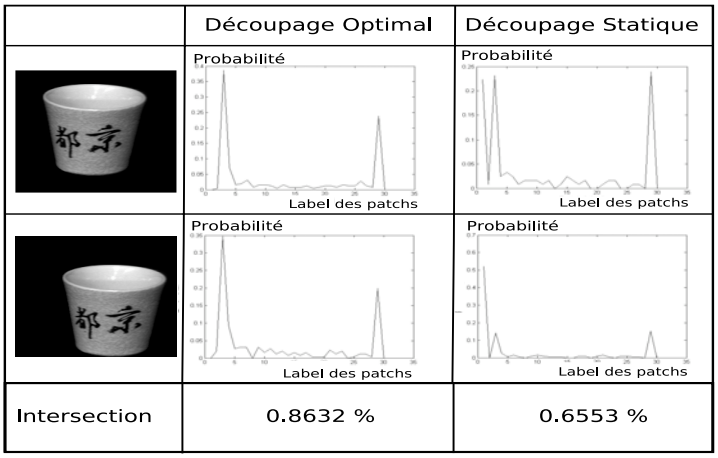

Fig. 6. Exemple de la probabilité de détection des labels du vocabulaire à partir de deux images contenant le même objet. Le découpage optimal présente la meilleure stabilité, puisqu'il conserve 86% contre 65% pour le découpage statique.

où x_i^k représente la composante i du vecteur de code $\mathbf{x}_k$ calculé sur la trame k et les y_i^k les échantillons du signal en sortie du $i$ème filtre.

Il serait intéressant de pouvoir utiliser une durée de la trame de $40ms$ car cela permettrait de faciliter la synchronisation audio et vidéo puisque habituellement cela correspond à la fréquence d'aquisition des caméras. Malheureusement les scènes auditives sont sujettes à des contraintes de stationarité spatiale qui obligent à utiliser des periodes de 10 à 20 ms. Partant d'un echantillonage audio de 48kHz, nous avons utilisé des fenêtres concourantes pour obtenir 8 fenêtres audio qui correspondent à la durée d'obtention d'une image.

B. Décision locale

L'approche que nous proposons pour le traitement des données audio s'inspire d'une technique déjà employée dans les applications de reconnaissance du locuteur : la modélisation prédictive des sources sonores par des réseaux de neurones [21]. Cette technique permet en effet d'estimer une distance entre une source audio inconnue et un ensemble de sources de référence. Pour cela elle prend en compte les composantes spectrales du signal (dans notre cas les coefficients calculés sur les N filtres) en même temps que leur évolution dynamique lorsque ce signal est non stationnaire.

Soit $\mathbf{x}_k = [x_1^k, x_2^k, \dots x_N^k]$ le vecteur de codification cochleaire correspondant k^{eme} frame audio où N est le nombre de coefficients. Pour un problème à M classes d'objets à reconnaitre, M réseaux de neurones sont entrainés pour associer à chaque paire de vecteurs de code consecutifs $\mathbf{x}_{k-2}$, $\mathbf{x}_{k-1}$ la prédiction du prochain vecteur à venir $\mathbf{x}_k$. Une fois le processus d'apprentissage des M réseaux terminé, ceux ci représentent les M sons de chaque objet. Durant la phase de reconnaissance un signal inconnu provenant du flux audio est codé, puis tous les couples de vecteurs $\mathbf{x}_{k-2}$, $\mathbf{x}_{k-1}$ sont présentés en entrée des M réseaux (voir la figure 7). L'erreur minimale de prédiction entre le vrai signal $\mathbf{x}_k$ est la valeur prédite $\hat{\mathbf{x}}_k$ est donnée par : $\epsilon_k = \hat{\mathbf{x}}_k - \mathbf{x}_k$. La décision locale consiste à labéliser chaque signal inconnu par la classe du réseaux fournissant l'erreur minimale. Les réseaux considérés sont de simples perceptrons multi-couches.

V. FUSION ET DÉCISION

La reconnaissance des sons provenant des objets s'effectue en deux phases : une phase de décision locale effectuée au niveau du vecteur courant suivie d'une phase de décision plus globale sur des périodes de temps plus longues. Deux types de décisions locales sont effectuées. La première consiste à cumuler les erreurs audio et video locales, calculant ainsi une erreur locale commune, la seconde à calculer deux

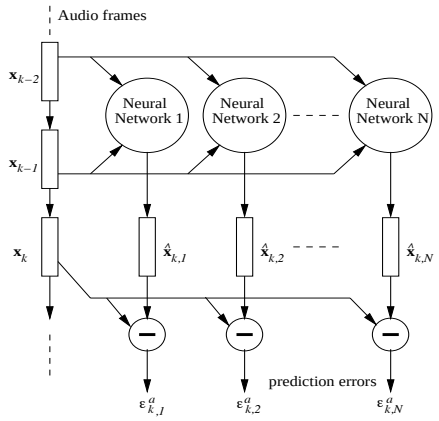

Fig. 7. Architecture du système de reconnaissance sonore.

erreurs indépendantes pour effectuer plus tard une décision globale. La décision globale étant bien sur d'affecter une classe d'objets à l'ensemble d'une sequence à partir des décisions locales.

A. Décision locale

Dans ce qui suit, $\epsilon^a_{k,i}$ représente l'erreur de prédiction calculée à partir des sorties des réseaux de neurones représentant les classes de sons $i, i \in \{1, \ldots, M\}$ pour des fenêtres audio k. On pose $\epsilon^v_{k,i}$ l'erreur équivalente produite par le système de reconnaissance de la vision. Nous nous proposons de comparer 5 méthodes de fusion entre les données audio et video pour le processus de décision locale.

1) Décision sans fusion ($LD1$): Dans ce procédé de fusion, chaque modalité est traitée séparément. La décision globale est obtenue à partir de toutes les décisions locales $\{c^a_k, c^v_k\}$ estimées séparemment pour chaque flux sur chaque frame en minimisant l'erreur locale [28] :

$$\begin{cases} c^a_k = \arg\min_{i=1,M}\{\epsilon^a_{k,i}\}, \\ c^v_k = \arg\min_{i=1,M}\{\epsilon^v_{k,i}\}. \end{cases} \tag{1}$$

2) Décision sans fusion étendue ($LD2$): Cette méthode de fusion fonctionne comme $LD1$ en rajoutant une seconde erreur de prédiction définie par :

$$\begin{cases} c^{a(2)}_k = \arg\min_{i=1,\ldots,M, i\neq c^a_k}\{\epsilon^a_{k,i}\}, \\ c^{v(2)}_k = \arg\min_{i=1,\ldots,M, i\neq c^v_k}\{\epsilon^v_{k,i}\}. \end{cases} \tag{2}$$

ceci à partir de toutes les décisions locales $\{c^a_k, c^{a(2)}_k, c^v_k, c^{v(2)}_k\}$.

3) Décision avec une fusion simple ($LD3$): Pour chaque frame k, une décision locale est caculée en choisissant la classe minimisant l'erreur audio et video, chaque frame est d'abord labélisée selon la règle suivante :

$$c_k = \arg\min\{\epsilon^a_k, \epsilon^v_k\} \text{ with } \epsilon^m_k = min_{i=1,\ldots,M}\{\epsilon^m_{k,i}\} \tag{3}$$

Il faut remarquer que cette méthode nécessite de normaliser à priori ϵ^a_k et ϵ^v_k avant de pouvoir fournir la décision locale :

$$\epsilon^m_k \longleftarrow \frac{1}{\sigma^m}\left(\epsilon^m_k - \overline{\epsilon^m_k}\right) \tag{4}$$

$\overline{\epsilon^m_k}$ étant l'erreur moyenne et σ^m la variance, tous deux calculés à partir de toutes les frames audio et vidéo de la séquence.

4) Decision avec une fusion pondérée ($LD4$): Parmi les méthodes les plus simples de fusion, il existe un processus utilisant une erreur pondérée entre les deux flux :

$$c_k = \arg\min\{\alpha_k^a \epsilon_k^a, \alpha_k^v \epsilon_k^v\} \text{ with } \epsilon_k^m = min_{i=1,\dots,M}\{\epsilon_{k,i}^m\} \tag{5}$$

Le processus de pondération utilise un ratio entre l'erreur minimale notée $\varepsilon_k^{m(1)}$ et une seconde erreur minimale $\varepsilon_k^{m(2)}$ sur les flux audio et video comme définis précédemment :

$$\alpha_k^{m\in\{a,v\}} = \frac{\epsilon_k^{m(1)}}{\epsilon_k^{m(2)}}. \tag{6}$$

B. Décision globale

Dans la section précédente, il a été présenté 4 méthodes de fusion locale. A l'issue nous obtenons alors un ensemble de décisions locales $c_{k,k=1,\dots,K}$ où K représente le nombre total de trames. Le premier algorithme de fusion global ($GD1$) ne considère pas les décisions locales mais une somme de celles ci sur l'ensemble de la séquence. Le second algorithme ($GD2$) calcule une décision à partir de toutes les décisions locales de la manière suivante :

– ($GD1$) decision (*Erreur globale*) ou Decision Globale 1 :

$$c = \arg\min_{i,i=1,\dots,M} \left\{ \sum_{k=1}^{K} \epsilon_{k,i} \right\} \tag{7}$$

– ($GD2$) decision (*Majorité simple*) ou Décision Globale 2 :

$$c = \arg\max_i \{|U_i|\}, \tag{8}$$

où U_i est l'ensemble de toutes les trames d'une classe $i, i \in \{1, \dots, M\}$ et $|U_i|$ son cardinal.

Ce type de méthode est souvent utilisé lorsque l'on a des contraintes de rapidité de temps de calcul, comme c'est le cas dans les applications robotiques, et que l'on ne peut mettre en oeuvre des méthodes statistiques (comme les modèles de Markov par exemple).

VI. Expérimentations

A. Conditions expérimentales

Nous avons mis au point une base de données comprenant 28 objets (jouets sonores) en mouvement devant un fond uniforme. Chaque objet émet un son particulier dû à son mouvement et à son interaction avec le sol. La figure 8 montre l'ensemble de ces objets. On peut reconnaître des sous-catégories sur le plan des formes, des couleurs et des sons émis. Les objets 1 à 5 sont sphériques et émettent des sons de roulement relativement semblables. Les objets 13 à 19 sont des objets à roues de forme et de dimensions semblables mais de couleurs différentes. Les objets 26 à 28 sont de formes dissemblables et émettent des sons caractéristiques qui leur sont propres (des sons artificiels non liés à leur déplacement).

Dans le but d'étudier toutes les configurations, trois différents enregistrements ont été effectués pour chaque objet, menant à trois bases différentes :

1) $B1$: Base d'apprentissage ;
2) $B2$: Base de test sans occlusions visuelles.
3) $B3$: Base de test avec des occlusions visuelles pendant le mouvement (voir la figure 8).
4) $B4$: Base de test avec des occlusions sonores générées par l'ajout artificiel d'un bruit blanc gaussien sur les signaux issus de B2.

Tous les objets effectuent un mouvement horizontal de la gauche vers la droite pour $B1$ et $B3$, et de la droite vers la gauche pour $B2$ et $B4$. Un panneau vertical positionné au milieu de la scène permet l'introduction d'occlusions visuelles. Le panneau couvre approximativement 30% de la scène (voir la

Fig. 8. *A gauche* : Vue globale des objets de la base de données. *A droite* : Images extraites de 4 séquences comportant des occlusions visuelles.

figure 8). Le bruit blanc ajouté à $B2$ possède un RSB de $-10dB$ et l'occlusion est présente durant 50% du temps.

Les signaux audio numériques proviennent des microphones équipant la caméra, échantillonnés à une fréquence de 48000Hz. Dans le but de caractériser finement les informations spectrales, nous calculons $N = 13$ coefficients par trames, ce qui représente un bon compromis entre la nécessité d'avoir des vecteurs de faible dimension, et une perte d'informations minimale. En conséquence de ce choix, les réseaux, de type perceptron multicouche à une couche cachée, comportent 26 entrées et 13 sorties.

B. Résultats experimentaux

1) Classification sans fusion: Après extraction et apprentissage des codes, tant visuels qu'auditifs, à partir des 28 objets de la base $B1$, nous avons effectué un premier test de reconnaissance sur les bases $B2$, $B3$ et $B4$, sans fusion : les objets sont reconnus en utilisant séparément les flux audio et vidéo et le critère $GD1$ (Eq. 7). Le tableau I montre les résultats obtenus sur $B2$, $B3$ et $B4$.

Dans le cas des objets de la base $B2$, sans perturbations les scores sont respectivement de 96,43% et 71,43%. Ils tendent à montrer qu'il est plus difficile de reconnaître des objets à partir des modalités considérées séparément. De fait, beaucoup de petits objets de la base émettent des sons similaires, introduisant la possibilité de confusions. Il semble que l'extraction des caractéristiques visuelles est suffisamment robuste puisque la modalité permet de reconnaître correctement 27 objets, soit un score de 96,43%.

TABLE I

TESTS DE RECONNAISSANCE SUR $B2$, $B3$, $B4$ SANS FUSION, EN UTILISANT L'ALGORITHME $GD1$

Databases	Video	Audio
B2	96,43%	71.43%
B3	57.14%	71.43%
B4	96,43%	10.71%

Les occlusion visuelles introduites dans la base $B3$ font fortement baisser les taux de reconnaissance de 96,43% à 57,14%. La robustesse de l'algorithme est remise en cause sur l'ensemble de la séquence. La reconnaissance est seulement possible à partir des trames non occultées mais elles ne permettent pas malgré tout d'obtenir un résultat convainquant. Les résultats sont les mêmes dans le cas de la modalité auditive lorsqu'elle est affectée par un bruit blanc. Les scores tombent de 71,43% to 10,71%.

2) Classification avec fusion: Les expérimentations que nous présentons dans cette section visent à comparer les quatre algorithmes de fusion proposés et à montrer l'apport de la fusion de données visuelles

et sonores pour la reconnaissance d'objets en mouvement. Nous avons testé les quatre algorithmes de fusion locale $LD1$ à $LD4$ sur les trois bases de test $B2$, $B3$ et $B4$.

TABLE II

TESTS DE RECONNAISSANCE SUR $B2$, $B3$, $B4$ AVEC L'ALGORITHME DE DÉCISION GLOBALE $GD1$ ET LES FUSIONS LOCALES $LD1$ À $LD4$ AVEC L'ALGORITHME DE DÉCISION GLOBALE $GD2$

Decision →	$GD1$	$GD2$			
Bases		$LD1$	$LD2$	$LD3$	$LD4$
B2	89.29%	100%	96.43%	82.14%	100%
B3	75.00%	78.57%	75.00%	71.43%	78.57%
B4	14.29%	96.43%	60.71%	67.86%	96.43%

La première colonne de la table peut être directement comparée aux résultats précédemment présentés dans la table I. Dans les deux cas, l'algorithme de décision globale utilisé est l'algorithme $GD1$. Sur $B2$, la fusion opérée ajuste les scores de 96.43% en vision et 71,43% en son à un niveau intermédiaire de 89,29%. En revanche, en présence d'une occultation visuelle (base $B3$), le score après fusion remonte au delà (75%) des scores initiaux (57.14% et 71.43%). Ceci illustre la complémentarité des deux modalités lorsqu'elle sont exploitées par l'étage de fusion. Sur $B4$, la dégradation apportée sur la modalité sonore n'est que très faiblement récupérée par la modalité visuelle, après fusion.

Les colonnes suivantes reportent les résultats obtenus avec le deuxième algorithme de fusion globale $GD2$ et les quatre variantes sur les décision locales $LD1$ à $LD4$. Sans occultation (base $B2$), la fusion permet de monter les taux de reconnaissance jusqu'à 100% (pour $LD1$ et $LD4$). Dans le cas de l'occultation visuelle, l'amélioration des scores apportée par la fusion $GD2$ est du même ordre de grandeur que celle apportée par la fusion $GD1$. Les algorithmes $LD1$ et $LD4$ présentent toujours de meilleurs performances. Enfin, concernant l'occultation sonore, l'algorithme $GD2$ s'avère nettement supérieur à l'algorithme $GD1$ puisque l'on remonte les scores de 14.29% à 96.43%.

Ces résultats montrent qu'une décision globale calculée à partir de décisions locales ($GD2$) s'avère être un mécanisme de fusion plus adapté qu'une décision globale calculée après sommation des erreurs locales ($GD1$).

La table II nous montre que dans le cadre de l'occlusion visuelle, les scores en vision qui descendent de 96.4% à 89.28% (table I) remontent à 92% avec $LD1$ et jusqu'à 96%, i.e. le score d'origine, avec $LD4$. Concernant la modalité sonore, la chute du score enregistrée en présence de bruit (de 78.57% à 7.14%, sur la table I) est bien compensée par la modalité visuelle qui permet de remonter au score d'origine de 78.57% avec les fusions $LD1$ et $LD2$. L'apport des décisions prises au second niveau, c'est à dire en considérant la comparaison des deuxièmes plus petites erreurs (algorithme $LD2$) n'apporte pas ici de réelle amélioration. Cette information est plus souvent utilisée pour caractériser la vraisemblance d'une décision. C'est précisément dans ce cadre que nous l'exploitons dans l'algorithme $LD4$, en définissant une pondération des modalités (coefficients α, Eq. 6). Cet algorithme de fusion est plus performant dans l'amélioration des scores lors d'occultations visuelles plutôt qu'en présence de perturbations sonores. Enfin, la première colonne de la table II montre les scores obtenus lorsqu'on utilise l'algorithme de fusion globale $GD1$. Les erreurs sont sommées sur l'ensemble des trames et modalités puis une décision globale sur la séquence est prise en suivant le principe de maximisation de la vraisemblance. Il est intéressant de noter que ce type d'algorithme est cette fois plus avantageux lorsque les données sonores sont perturbées puisqu'il permet de remonter les scores de 7.14% à 82.14%. En revanche, l'algorithme $GD2 - LD4$ reste supérieur lors d'occultations visuelles.

VII. CONCLUSIONS

Nous avons présenté dans cet article des méthodes de fusion de données audio-vidéo dans le cas particulier d'une reconnaissance d'objets par un système de vision artificiel. Les travaux présentés visaient

à introduire une nouvelle méthode d'extraction de patches images pour fournir matière à un système de codage par sac de mots. Comme l'ont montré les résultats obtenus, le système autorise une décomposition unique d'une image à partir du moment où le contenu et la distribution spatiale de l'image est le même. Le système de reconnaissance audio a utilisé des méthodes classiques, le but étant de montrer que moyennant une extraction efficace d'information nécessite des méthodes simples de fusion entre modalités. Les résultats montrent que l'utilisation de modalités complémentaires permet d'atteindre des taux élevés de reconnaissance. A long terme il s'agira de développer des espaces de fusion plus homogènes de telles sorte que des espaces communs de projection de données s'opèrent au plus bas niveau dans le processus de reconnaissance brisant le schéma classique de deux voies parallèles pour les deux modalités pour une fusion à long terme.

VIII. Remerciements

Les auteurs remercient C. C pour l'ensemble de son travail efficace, son aide et son indéfectible effort qui ont permis de mener à bien ce papier dans les délais les plus courts.

References

[1] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2003.

[2] Stepan Obdrzalek and Jiri Matas. Object recognition using local affine frames on distinguished regions. In Paul L. Rosin and David Marshall, editors, Proceedings British Machine Vision Conference, volume 1, pages 113–122.BMVA, 2002.

[3] Se, S. and Lowe, D. and Little, J., Vision-based Mobile robot localization and mapping using scale-invariant features, 2001, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA),Seoul, Korea.

[4] Yang, M.-H., Roth, D. and Ahuja, N. (2000) Learning to Recognize 3D Objects with SNoW in ECCV.

[5] M. J. Beal, H. Attias and N. Jojic. "Audio-Video Sensor Fusion with Probabilistic Graphical Models", In pp 736–752, *ECCV 2002*.

[6] M. Beal, N. Jojic and H. Attias. "A graphical model for audiovisual object tracking". In : *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp 828–836, 2003.

[7] Beltran-Gonzalez C. and G. Sandini. "Visual attention priming based on crossmodal expectations". In : *Intelligent Robots and Systems*, pp 1060–1065, 2005. (IROS 2005).

[8] R. A. Brooks, C. Breazeal, R. Irie, C. Kemp, M. Marjanovic, B. Scassellati, and M. Williamson. "Alternative essences of intelligence". In *Proceedings of the 15th AAAI*, AAAI Press, pp 961–968, 1998.

[9] Filliat, D. "A visual bag of words method for interactive qualitative localization and mapping". In : *International Conference on Robotics and Automation (ICRA)*, 2007.

[10] J. W. Fisher and Trevor Darrell. "Informative Subspaces For Audio-Visual Processing : High-Level Function From Low-Level Fusion" (2002)

[11] Frederic Jurie and Bill Triggs, "Creating Efficient Codebooks for Visual Recognition", *International Conference on Computer Vision*, 2005.

[12] Viola, P. and Jones, M. (2001). "Rapid object detection using a boosted cascade of simple features", pp 1332-1338,San Diego.(2007)

[13] , A. Haasch, S. Hohenner, S. Huwel, M. Kleinehagenbrock, S. Lang, I Topsis, G. A. Fink, J. Fritsch, B. Wrede, and G. Sagerer. "BIRON, The Bielefeld Robot Companion". In : *Proc. Int. Worshop on Advances In Service Robots*, Stuttgart, 2004.

[14] J. Hershey and J. R. Movellan, "Audio vision : Using audiovisual synchrony to locate sounds", in *Advances in Neural Information Processing Systems* 12. S. A. Solla, T. K. Leen and K. R. Muller (eds.), pp 813–819. MIT Press., 2000.

[15] Thomas Hofmann. "Probabilistic Latent Semantic Analysis". In : *Proc. of Uncertainty in Artificial Intelligence*. 1999.

[16] Irie, R. "Mutlimodal Sensory Integration for localization" in *Humanoid Robot*. In Proceedings of Second IJCAI Workshop on Computational Auditory Scene Analysis (CASA'97), IJCAI-97.

[17] S. Lang, M. Kleinehagenbrock, S. Hohenner, J. Fritsch, G. A. Fink, and G. Sagerer. "Providing the basis for human-robot-interaction : A multi-modal attention system for a mobile robot". In *Proc. Int. Conf. on Multimodal Interfaces*, pp 28–35, Vancouver, Canada, November 2003. ACM.

[18] Lowe, D., "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision (IJCV)*,91–110, 2004.

[19] S. Argentieri, P. Danes, P. Souères and P. Lacroix. "An Experimental Testbed for Sound Source Localization with Mobile Robots using Optimized Wideband Beamformers" In pp 909–914, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005

[20] S. Argentieri, P. Danes, P. Souères and P. Lacroix. "Modal Analysis Based Beamforming for Nearfield or Farfield Speaker Localization" in Robotics, pp 866–871, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006

[21] Melouk, A. and Gallinary, P. "A discriminative neural predictive system for speech recognition". *ICASSP*, Vol. 2, pp 533–536 (1993).

[22] Mermelstein, P. "Distance measures for speech recognition, psychological and instrumental". *Pattern Recognition and Artificial Intelligence*, pp 374–388, 1976.

[23] K. Nakadai, H. G. Okuno, and H. Kitano, "Robot recognizes three simultaneous speech by active audition", in *Proc. of IEEE International Conference on Robotics and Automation (ICRA 2003)*.

[24] R. D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C.Zhang, and M. H. Allerhand, "Complex sounds and auditory images," In *Auditory Physiology and Perception*, (Eds.) Y Cazals, L. Demany, K.Horner, Pergamon, Oxford, 1992, pp 429–446.

[25] J. Peskin and B. Scassellati. "Image Stabilization through Vestibular and Retinal Feedback". In R. Brooks, ed. *Research Abstract*, MIT Artificial Intelligence Laboratory. 1997.

[26] Rumelhart, D.E. and Hinton, G.E. and Williams, R.J. "Learning representations by back-propagating errors". *Nature*, Vol. 323, pp 533–536 (1986)

[27] I. Ulrich and I. Nourbakhsh. "Appearance Based Place Recognition for topological Localization". In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp 1023–1029, 2000.

[28] Ming Liu, Ziyou Xiong, Stephen M. Chu, Zhenqiu Zhang and Thomas S.Huang. "Audio visual word spotting". *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[29] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. (2002). "Shape distributions". *ACM Transactions on Graphics*