

SELLAOUTI AYMEN  
Yasmine Jaâfra  
Atef Hammouda

*Méta-apprentissage pour le  
paramétrage de l'étape de  
segmentation d'images*

Cher auteur,

Le comité scientifique des Ateliers TAIMA'15 a terminé la sélection des papiers soumis et nous avons le plaisir de vous informer que votre communication T15-37 intitulée "*Méta-apprentissage pour le paramétrage de l'étape de segmentation d'images*" est acceptée en session orale.

Nous vous demandons maintenant de préparer, pour le 12 Avril 2015 (dernier délai de rigueur), la version définitive de votre papier, en tenant compte au mieux des remarques des relecteurs (cf. ci-joint).

Vous êtes invités à soumettre sur le site de TAIMA 2015 la version finale de votre papier selon le modèle indiqué (<http://www.arts-pi.org.tn/taima2015/soumission.php>) :

Veillez vous connecter à votre compte et charger une archive zip contenant :

1. Tous les fichiers sources :
  - Les fichiers LaTeX pour le texte (.tex),
  - Les fichiers RTF ou Word,
  - Les figures (format ps/eps, pdf/jpg).
2. Fichier PDF final (les papiers en MS-Word doivent être aussi convertis en fichier PDF)
3. Les fichiers RTF/Word (pour MS-Word)

Plus de détails sur les modalités de soumission de la version définitive sont disponibles sur le site des ateliers à l'adresse : <http://www.arts-pi.org.tn/taima2015/soumission.php>

En cas de problèmes de soumission veuillez envoyer un email à l'adresse suivante :  
[sabra.mab@gmail.com](mailto:sabra.mab@gmail.com)

# Méta-apprentissage pour le paramétrage de l'étape de segmentation d'images

## Meta-learning for image segmentation parameter selection

Aymen Sellaouti<sup>1,2</sup>, Yasmina Jaâfra<sup>1</sup>, et Atef Hamouda<sup>1</sup>

<sup>1</sup> Université de Tunis El Manar, Faculté des Sciences de Tunis, LIPAH,  
2092 Tunis, Tunisie.

<sup>2</sup> Université de Strasbourg, LSIT, Pôle API, Bd Sébastien Brant,  
67412 Illkirch, France.

{aymen.sellaouti, yasminda.jaafra}@gmail.com, atef.hamouda@yahoo.fr

**Résumé** L'étape de segmentation requiert généralement l'ajustement d'un ensemble de paramètres qui dépend de la variabilité des caractéristiques des images traitées. Cet article présente une approche de méta-apprentissage par stacked generalization pour le paramétrage de l'étape de segmentation dans l'analyse orientée objets des images satellitaires très haute résolution. La méta-connaissance est générée à partir de l'intégration des prédictions des apprenants de base à travers une technique de validation croisée. Une étude expérimentale sur des images satellitaires couvrant la zone urbaine de la région de Strasbourg nous a permis d'évaluer l'efficacité de notre système.

**Mots clés** Analyse orientée objets, segmentation, images satellitaires à très haute résolution, méta-apprentissage, stacked generalization.

**Abstract** Most image segmentations require control parameters setting that depends on the variability of processed images characteristics. This paper introduces a meta-learning system using stacked generalization to adjust segmentation parameters within an object-based analysis of very high resolution urban satellite images. Meta-knowledge is built from the integration of base-learners performance evaluated by cross-validation. An experimental study on satellite images covering the urban area of Strasbourg region enabled us to evaluate the effectiveness of the adopted approach.

**Key words** Object-based analysis, segmentation, very high resolution satellite image, meta-learning, stacked generalization.

## 1 Introduction

L'analyse d'image satellitaire basée objet, qui a pris de l'importance avec la mise en exploitation de l'imagerie à très haute résolution (THR), est venue combler le gap sémantique entre les données sous forme de pixels et les thèmes qui doivent être identifiés. L'analyse orientée objets implique l'intégration d'une étape préliminaire



dans la chaîne de traitement d'image afin de délimiter les frontières des objets d'intérêt possédant une réalité thématique. L'insertion de cette étape de segmentation préalable à la classification a montré son efficacité notamment dans le cas des images de télédétection THR [4]. D'une façon générale, les algorithmes de segmentation requièrent un paramétrage qui impacte directement le résultat de classification. Ce paramétrage dépend de la variabilité de certaines caractéristiques entre différentes images, telles que le contraste, le bruitage et la luminosité. Cette variabilité peut être expliquée par le changement des conditions environnementales telles que le temps de la journée, la saison de l'année et la présence de nuages. Le paramétrage de l'algorithme de segmentation doit être ainsi ajusté par rapport à ce changement de caractéristiques de l'image afin d'améliorer les résultats de la classification.

Plusieurs techniques d'apprentissage automatique ont été proposées pour le paramétrage d'algorithmes de segmentation. Derivaux et al. [5] utilisent un algorithme génétique pour optimiser le paramétrage de l'algorithme de segmentation Watershed. Le paramétrage de la segmentation par fusion sous eCognition a été étudié par Benz et al. [2] dans le cadre d'une analyse orientée objets basée sur la logique floue. Martin et al. [7] examinent le paramétrage de trois algorithmes de segmentation à travers un modèle d'apprentissage basé sur les réseaux de neurones. La sélection de la technique d'apprentissage la plus appropriée à la nature du problème à résoudre n'est pas une tâche évidente puisqu'il n'existe pas, selon les théorèmes "No Free Lunch" [13], un algorithme meilleur que tous les autres sur l'ensemble des tâches à traiter. La solution spécifiant le modèle d'apprentissage optimal dans un contexte donné a été définie au sein de la discipline de l'apprentissage automatique sous le nom de méta-apprentissage. Il se distingue de l'apprentissage conventionnel par l'étendue de son niveau d'adaptation. Tandis que ce dernier possède un biais fixe a priori et accumule de l'expérience par rapport à une tâche donnée, le méta-apprentissage choisit, quant à lui, dynamiquement son biais selon le contexte d'étude.

Afin d'améliorer la performance globale de l'analyse d'image orientée objets, nous allons mettre en place une stratégie de méta-apprentissage par stacked generalization pour le paramétrage de l'algorithme de segmentation par la ligne de partage des eaux (Watershed). Cette approche consiste à combiner les résultats de classifieurs issus d'algorithmes d'apprentissage appliqués au même jeu de données afin de réaliser une performance de prédiction meilleure que celle obtenue par ces algorithmes employés individuellement.

Dans la suite de cet article, nous présentons tout d'abord, l'algorithme de segmentation utilisé et les paramètres associés. Nous décrivons par la suite l'approche de stacking implémentée ainsi que les résultats de l'expérimentation. Enfin la conclusion et les perspectives d'application sont proposées.

## 2 Paramétrage de l'algorithme de segmentation Watershed

L'algorithme de segmentation Watershed présenté dans [11] fait partie de la famille des algorithmes basés frontières. Il est considéré comme la méthode de segmentation principale de la morphologie mathématique. Il se base sur le principe de division de l'image en des zones homogènes appelées "bassins versants". Pour ce faire, l'image est considérée comme un relief topographique (ou carte d'altitude) où son gradient est utilisé dans l'affectation d'un niveau de gris à chaque pixel correspondant à la hauteur du point considéré. L'étape suivante appelée "montée des eaux" consiste à inonder la carte d'altitude obtenue à partir de ses minima locaux générant ainsi des bassins de rétention ou régions. Les frontières issues de la rencontre de deux bassins et empêchant leur fusion (digues) sont désignées lignes de partage des eaux.

Watershed se distingue par sa tendance à générer des images sur-segmentées où chaque objet d'intérêt est représenté par plusieurs régions (au lieu d'une). Sa performance dépend étroitement du réglage d'un ensemble de paramètres qui doivent être ajustés par rapport aux caractéristiques des images à traiter [5]. Notre approche consiste à introduire trois paramètres dans l'algorithme Watershed qui vont instaurer un seuillage minimal pour conditionner la construction des régions de l'image :

- Seuillage de gradient : une fois la surface topographique créée, tout pixel  $p$  ayant une valeur de gradient  $G(p)$  inférieure à un seuil donné  $h_{min}$  est mis à zéro.
- Profondeur de bassin : un bassin  $r_i$  ne sera séparé d'un autre par une ligne de partage des eaux que si sa profondeur  $d_i$  est supérieure à un seuil donné  $d_{min}$ .
- Fusion des régions : si la distance euclidienne entre moyennes spectrales de deux régions  $r_i$  et  $r_j$  est inférieure à un seuil prédéfini  $m_{min}$ , alors elles seront fusionnées.

L'application de notre algorithme de segmentation requiert ainsi le réglage des trois paramètres  $h_{min}$ ,  $d_{min}$  et  $m_{min}$ . Dans le cadre de notre expérimentation, nous allons définir 3 classes  $P_1$ ,  $P_2$  et  $P_3$  représentant des combinaisons différentes des paramètres  $h_{min}$ ,  $d_{min}$  et  $m_{min}$ . Chaque image de notre collection est segmentée en utilisant ces combinaisons.

## 3 Système de stacking pour le paramétrage de Watershed

Nous allons mettre en place une méthodologie pour produire, à partir d'exemples, les connaissances nécessaires au paramétrage de l'algorithme de segmentation Watershed. Elle se base sur une stratégie de méta-apprentissage par stacked generalization capable de déterminer le paramétrage de segmentation à partir de la combinaison de prédictions fournies par un ensemble d'algorithmes d'apprentissage. Notre approche

se compose d'une étape d'acquisition de connaissances et d'un système de stacking à deux niveaux d'apprentissage tels que illustrés dans la figure 1.

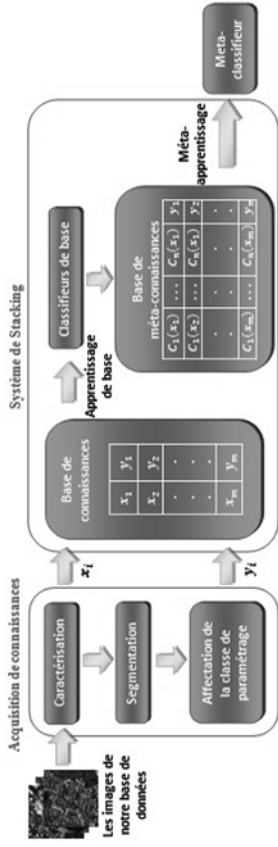


Figure 1. Approche de stacking pour le paramétrage de segmentation.

### 3.1 Acquisition de connaissances

Cette étape aboutira à la réalisation de la base de connaissances qui va constituer le point de départ de notre stratégie de méta-apprentissage. Il s'agit tout d'abord de caractériser les images de notre base de données afin de détecter celles qui nécessitent le même traitement de segmentation. Attig et al. [1] étudient l'impact de quatre descriptions différentes sur la détermination du bon paramétrage d'un algorithme de segmentation et confirment la pertinence des caractéristiques statistiques et de texture. Nous déterminons pour chaque image de notre base de données un vecteur  $x_i$  de 4 caractéristiques statistiques et de texture qui sont le contraste, l'énergie, l'homogénéité et la variance des sommes.

Nous avons fixé la plage des classes de paramétrage à trois combinaisons  $P_1$ ,  $P_2$  et  $P_3$  des variables  $h_{min}$ ,  $d_{min}$  et  $m_{min}$  définies ci-dessus. Les valeurs retenues ont été estimées suite à une procédure manuelle d'essai-erreur sur un échantillon d'images à travers l'évaluation des classifications issues des segmentations de ces images.

L'évaluation de la qualité de segmentation effectuée par les 3 classes de paramétrage sur chaque image revient à l'estimation de la précision de la classification appliquées à ces segmentations [5]. Cette évaluation nous permet d'affecter, à chaque image de notre base de données, le meilleur paramétrage de segmentation parmi  $P_1$ ,  $P_2$  ou  $P_3$ . Elle est déterminée à travers la concordance entre le résultat d'une classification et les benchmarks des classes route, bâtiment et végétation relatifs à l'image en utilisant une matrice de confusion.

À la fin de cette phase, nous affectons aux images de notre base de données, qui ont été caractérisées précédemment par leurs propriétés statistiques et de texture, leur classe de paramétrage de segmentation réelle parmi  $P_1$ ,  $P_2$  ou  $P_3$ . Nous

complétons ainsi la construction de la base de connaissances qui va servir comme point d'entrée pour notre système de méta-apprentissage.

### 3.2 Système de stacking

Le méta-apprentissage désigne la capacité d'un système d'apprentissage à augmenter son efficacité et son aptitude d'apprendre à apprendre à travers l'expérience. En effet, il offre aux systèmes d'apprentissage la possibilité d'adaptation au domaine ou à la tâche étudiés à travers l'identification des stratégies d'apprentissage les plus appropriées [10]. Le méta-apprentissage par stacked generalization initié par les travaux de Wolpert [12] consiste à exploiter la variabilité des résultats produits par l'application d'un ensemble d'algorithmes de base à un même jeu de données. L'objectif est de réaliser un système à deux niveaux d'apprentissage par la combinaison d'algorithmes d'apprentissage en incorporant leurs prédictions dans la méta-connaissances:

- Premier niveau ou apprentissage de base: à la suite de l'application d'apprenants sur un jeu de données, leurs prédictions sont intégrées dans la représentation des caractéristiques initiale par une procédure de validation croisée afin de construire la nouvelle base d'entraînement du méta-apprenant:

Soit  $D$  la base de connaissances obtenue dans l'étape précédente, formée des instances  $e_i = (x_i, y_i)$  où  $x_i$  est le vecteur de caractéristiques de l'image  $i$  et  $y_i$  sa classe de paramétrage réelle correspondante et  $L_1, \dots, L_n$  l'ensemble des  $n$  algorithmes d'apprentissage. Afin de réaliser une validation croisée à  $J$  blocs,  $D$  est divisé aléatoirement en  $J$  parties disjointes  $D^1, \dots, D^J$  de tailles égales. A chaque  $j$ ème bloc ( $j = 1 \dots J$ ), les algorithmes d'apprentissage  $L_1, \dots, L_n$  sont appliqués à la partie d'entraînement  $D - D^j$ . Les classifieurs induits par la phase d'entraînement  $C_1, \dots, C_n$  où  $C_k = L_k(D - D^j)$  ( $k = 1 \dots n$ ) sont appliqués, par la suite, à la partie de test  $D^j$ . La concaténation des prédictions de ces classifieurs pour chaque vecteur d'attributs  $x_i$  de  $D^j$  avec  $y_i$ , sa classe réelle correspondante, forme un nouveau jeu de données méta-niveau  $MD^j$  du bloc  $j$ .

- Deuxième niveau ou méta-apprentissage: cette étape consiste à appliquer un méta-apprenant à la nouvelle base de méta-connaissances obtenue afin d'induire des règles de prédiction de la classe appropriée pour un nouvel exemple traité selon la combinaison des prédictions des apprenants de base. En effet, à la fin de la procédure de validation croisée, l'union  $MD = \cup MD^j$  constitue notre nouvelle base de méta-connaissances, à laquelle le méta-apprenant  $L_M$  est appliqué pour induire le méta-classifieur  $C_M$ . La transformation du jeu de données initial  $D$  implique le transfert de l'information relative aux prédictions des classifieurs de base vers ce méta-classifieur. La dernière étape consiste à appliquer les apprenants de base  $L_1, \dots, L_n$  à la totalité du jeu de données  $D$



afin d'induire les classifieurs finaux du niveau de base  $C_1, \dots, C_n$  qui seront utilisés au cours de traitement d'une nouvelle instance.

Le traitement d'une nouvelle image par notre système de stacked generalization revient à lui attribuer, dans une première étape, le vecteur de prédictions des classifieurs finaux sur la base de ses caractéristiques statistiques et de texture déterminées. Ce vecteur permet, par la suite, au méta-classifieur de produire une prédiction finale de la classe de paramétrage de l'algorithme de segmentation Watershed pour l'image traitée.

## 4 Expérimentation et évaluation

Nous avons mené notre expérimentation sur 50 images THR (Quickbird) couvrant la zone urbaine de la région de Strasbourg. Un échantillon des images est présenté dans la figure 2.

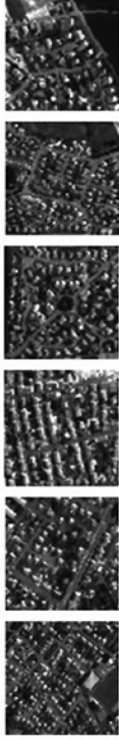


Figure 2. Un échantillon des images de test.

Les algorithmes d'apprentissage sélectionnés pour notre expérimentation sont la machine à vecteurs de support (SVM) et l'analyse discriminante utilisés comme des apprenants de base alors que l'arbre de décision va remplir le rôle du méta-apprenant et d'apprenant de base. Les choix effectués sont fondés sur les travaux de Seewald [8] et Skalak [9]. Ces apprenants sont appliqués à la base de connaissances par validation croisée à 25 blocs affectant trois prédictions de classe de paramétrages à chacune des images de la base de connaissances. La transformation de celle-ci par l'intégration d'information relative aux prédictions des apprenants de base nous permet de réaliser notre base de méta-connaissances et de passer vers le deuxième niveau d'apprentissage. Le méta-apprenant est entraîné sur cette base générant le méta-classifieur qui va être utilisé dans les prédictions de classes de paramétrage pour les nouvelles instances. La figure 3 illustre la différence entre deux classifications issues d'un paramétrage de segmentation ajusté et d'un autre aléatoire.

Le tableau 1 présente les prédictions de classes de paramètres de segmentation réalisées par les classifieurs de base et le système de stacking pour un échantillon d'images. La valeur ajoutée de notre système est mise en exergue dans les cas où les prédictions des apprenants de base sont divergentes (cas 2 et 6) ou que leurs trois prédictions sont incorrectes (cas 7) et que notre système de stacking parvient à prédire la classe de paramétrage correcte.

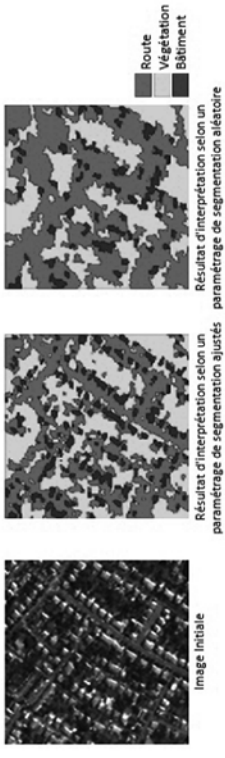


Figure 3. Impact d'ajustement des paramètres de Watershed sur l'interprétation d'image.

Table 1. Une comparaison entre les prédictions du système de stacking et les classifieurs de base.

	Arbre de décision	SVM	Analyse discriminante	Stacking	Classe réelle
Cas 1	1	1	1	1	1
Cas 2	2	3	1	3	3
Cas 3	2	2	2	2	2
Cas 4	3	1	1	1	1
Cas 5	2	2	1	2	2
Cas 6	1	2	3	1	1
Cas 7	1	2	2	3	3
Cas 8	1	3	3	1	1
Cas 9	3	1	1	1	2
Cas 10	2	1	1	2	3

La performance globale de notre approche de méta-apprentissage est mesurée par le pourcentage de prédictions correctes communément utilisé dans les problèmes d'apprentissage et appelé encore ratio de taux de réussite [3]:

$$RTR = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total des prédictions}} \times 100$$

Dans la phase d'estimation de leur approche de stacking, Lisin et al. [6] comparent la performance du système de méta-apprentissage à celle des apprenants de base employés individuellement. Dans notre étude, le système de stacked generalization dépasse nettement, avec un RTR de 64%, les performances de l'arbre de décision, SVM et analyse discriminante dont les RTR ont atteint respectivement 52%, 50% et 42%.

## 5 Conclusion

Nous avons proposé dans cet article une méthodologie de méta-apprentissage par stacked generalization pour le paramétrage de l'algorithme de segmentation Watershed. Nous avons mené l'expérimentation de notre méthodologie sur des images couvrant la zone urbaine de la région de Strasbourg. Les résultats obtenus montrent que

la performance de notre système de stacked generalization dépasse nettement celle des apprenants de base appliqués individuellement, confirmant notre hypothèse de base que le méta-apprentissage augmente l'efficacité de la tâche d'analyse d'images orientée objets. En guise de perspective pour notre recherche, nous envisageons d'intégrer une étape de feedback pour l'enrichissement de la base de connaissances au cours de l'exécution de nouvelles images. Il serait également intéressant d'augmenter le nombre d'images de test en diversifiant la nature des scènes afin d'obtenir une variabilité plus importante des caractéristiques des images.

## References

1. A. Attig and P. Perner. A study on the case image description for learning the model of the watershed segmentation. *Transactions on Case-Based Reasoning*, 2(1):41–53, 2009.
2. U. Benz, P. Hofmann, G. Willhauck, I. Lingenfelder, and M. Heynen. Multi-resolution, object-oriented fuzzy analysis of remote sensing data for gis-ready information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58(3):239–258, 2004.
3. P. Brazdil, C. Soares, and J. D. Costa. Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Machine Learning*, 50(3):251–277, 2003.
4. C. Cleve, M. Kelly, F. Kearns, and M. Moritz. Classification of the wildland urban interface: A comparison of pixel and object-based classifications using high-resolution aerial photography. *Computers, Environment and Urban Systems*, 32(4):317–326, 2008.
5. S. Derivaux, S. Lefevre, C. Wemmert, and J. Korczak. On machine learning in watershed segmentation. In *IEEE International Workshop on Machine Learning in Signal Processing (MLSP)*, pages 187–192, 2007.
6. D. Lisin, M. Mattar, M. Blaschko, M. Benfield, and E. Learned-Miller. Combining local and global image features for object class recognition. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, volume 3, page 47, 2005.
7. V. Martin, N. Maillot, and M. Thonnat. A learning approach for adaptive image segmentation. In *International Conference on Computer Vision Systems*, page 40, 2006.
8. A. Seewald. *Towards Understanding Stacking: Studies of a General Ensemble Learning Scheme*. The Vienna University of Technology, 2003.
9. D. Skalak. *Prototype Selection for Composite Nearest Neighbor Classifiers*. PhD thesis, 1997.
10. R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95, 2002.
11. L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Pattern Recognition and Machine Intelligence*, 13(6):583–598, 1991.
12. D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.
13. D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

# Apprentissage des paramètres de bruit d'un système linéaire gaussien couple

Valérian Némesin<sup>1</sup> et Stéphane Derrode<sup>2</sup>

<sup>1</sup> Institut Fresnel (CNRS UMR 7249), Aix-Marseille Université, 13397 Marseille Cedex 20, France.  
valerian.nemesin@centrale-marseille.fr

<sup>2</sup> LIRIS (CNRS UMR 5205), École Centrale de Lyon, 69134 Écully, France.  
stephane.derrode@ec-lyon.fr

**Résumé** Nous proposons un algorithme non supervisé d'apprentissage *partiel et contraint* des paramètres d'un système linéaire gaussien couple, par recherche du maximum de vraisemblance grâce à un algorithme EM (*Expectation-Maximization*). Le terme "partiel" signifie que seule une partie des paramètres est à estimer (l'autre partie étant supposée connue grâce à la physique du problème), tandis que "contraint" signifie que les paramètres restant à estimer peuvent être liés entre-eux (*p. ex.* par un facteur de proportionnalité inconnu). Ici, nous nous restreignons au cas particulier où seule la matrice de covariance des bruits doit être estimée, comme c'est le cas lors de la poursuite d'une cible en traitements vidéo.

**Mots clés** Système linéaire gaussien, Apprentissage non supervisé, Algorithme EM.

**Abstract** We propose an algorithm for the *partial and constrained* unsupervised learning of parameters involved in a pairwise linear Gaussian system, by means of the EM (*Expectation-Maximization*) principle. The "partial" term means that only a subset of the parameters is unknown and needs to be estimated (the other part is assumed to be known based on a priori knowledge about the studied system), whereas the "constrained" term means that the remaining parameters to be estimated can be self-linked (*e.g.* by an unknown scale factor). Here we limit ourselves to the particular case where only the noise covariance matrix must be estimated, as in the case of target tracking in video processing.

**Key words** Gaussian state-space systems, Unsupervised parameter estimation, EM algorithm.

## 1 Introduction

Un problème important en traitement du signal consiste à estimer un ensemble d'états cachés  $\mathbf{x} = \{\mathbf{x}_n\}_{n \in [0:N]}$ ,  $\mathbf{x}_n \in \mathbb{R}^{n_x}$ , à partir d'observations  $\mathbf{y} = \{\mathbf{y}_n\}_{n \in [0:N]}$ ,  $\mathbf{y}_n \in \mathbb{R}^{n_y}$ . Dans ce travail, nous nous intéressons au système linéaire gaussien couple (SLGC) suivant

$$\begin{pmatrix} \mathbf{x}_{n+1} \\ \mathbf{y}_n \end{pmatrix}_{t_{n+1}} = \underbrace{\begin{pmatrix} \mathbf{F}^{xx} & \mathbf{F}^{xy} \\ \mathbf{F}^{yx} & \mathbf{F}^{yy} \end{pmatrix}}_{\mathbf{F}} \underbrace{\begin{pmatrix} \mathbf{x}_n \\ \mathbf{y}_{n-1} \end{pmatrix}}_{t_n} + \underbrace{\begin{pmatrix} \omega_{n+1}^x \\ \omega_{n+1}^y \end{pmatrix}}_{\omega_{n+1}}, \quad (1)$$