

# L'utilisation du MeanShift pour la segmentation d'image

Nelle Varoquaux

January 16, 2012

## Abstract

*Peu connu de la littérature, l'algorithme Mean Shift présente un intérêt particulier pour les tâches de vision impliquant une étude d'un espace de caractères ou descripteurs. [4] remet au goût du jour cet algorithme en 2002. Nous présentons ici une étude détaillée celui-ci Mean Shift, ainsi qu'une application à la segmentation de l'image, telle que décrit dans [4]*

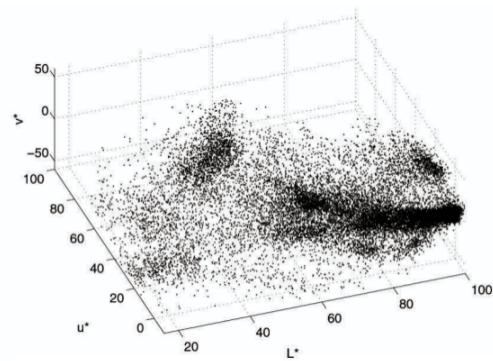
## 1 Introduction

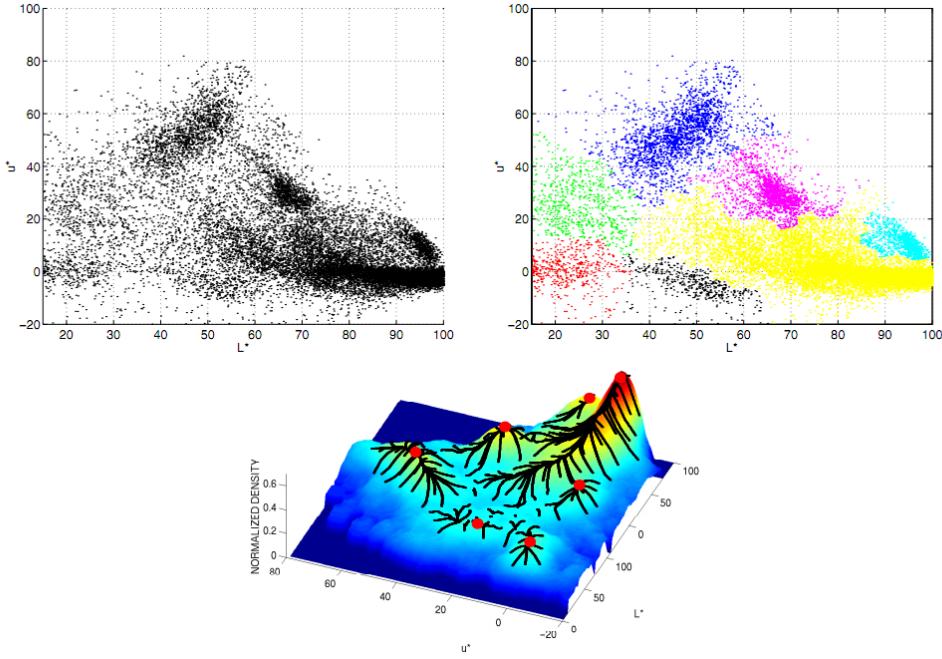
De nombreuses tâches de vision bas niveau nécessite l'étude d'un espace des caractéristiques, ou des descripteurs. De nombreuses techniques reposent sur une bonne évaluation des paramètres: ceux-ci sont souvent devinés.

Un espace de paramètres est obtenu en étudiant localement une image: l'image d'entrée peut par exemple être découpée en patchs, qui sont eux même prétraiter. Des descripteurs sont extraits, et associé à un point dans l'espace: le milieu du patch. Une fois que toute l'image est traité, on peut obtenir les descripteurs les plus significatifs, les descripteurs contenant le plus d'information, en calculant les régions les plus denses de l'espace de descripteurs: ces régions correspondent à des clusters.

La nature des descripteurs dépendent de l'application cherchée: cela peut être des descripteurs locaux à un pixel, telle que une représentation des couleurs, ou des descripteurs calculés à partir d'un petit patch de l'image, représentant une texture.

L'analyse de cette espace varie elle aussi selon la tâche à effectuer. Malgré le nombre important d'algorithmes de clustering, peu d'entre eux sont adaptés à l'étude d'un espace de descripteurs: beaucoup reposent sur une connaissance préalable du nombre de clusters ou de la forme de ceux-ci,





ou alors sont mal adaptés à l'étude d'un espace de descripteurs trop complexe, ce qui est souvent le cas dans le domaine du traitement d'image. Parmi les algorithmes de clustering les plus standards, on peut citer le k-means, qui nécessite une connaissance du nombre de cluster, et les algorithmes hiérarchiques, qui consiste à agréger des clusters entre eux, ou à les séparer, trop peu efficace pour le traitement d'image.

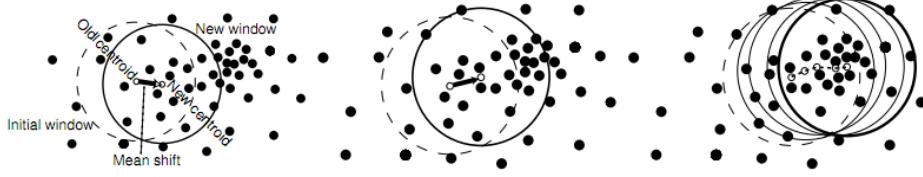
On présente ici une détection de mode, maxima locaux d'une estimation non paramétrique de densité de probabilité du noyau grâce à l'algorithme meanshift.

## 2 Mean shift

L'algorithme de clustering Mean Shift ne nécessite pas la connaissance au préalable du nombre de cluster, ni la forme des clusters. Présenté pour la première fois par [5] en 1975, le mean shift est un estimateur non paramétrique du gradient de la densité de probabilité. Disparu de la littérature, il est remis au goût du jour par [2] en 1995. Une application au filtrage et à la segmentation est ensuite proposée en 1997, 1999 dans [3]. Il est ensuite utilisé dans des domaines tels que le suivi d'objet, le maillage, et la segmentation d'image. Cette grande variété d'application s'explique par l'exploitation de l'espace des descripteurs.

L'algorithme repose sur la recherche de maximum d'une densité de probabilité, *un mode*. Une région est caractérisée par une densité de probabilité, elle même représentée par un mode. Plusieurs régions impliquent donc plusieurs modes. Trouver à quelle région appartient une donnée correspond donc à trouver le mode de cette donnée.

- Déterminer aléatoirement des régions d'intérêt
- Déterminer les centroïdes des régions



- Recalculer les régions d'intérêt autour des centroïdes
- Répéter les étapes jusqu'à la convergence

On note  $K(\mathbf{x})$  le noyau, qui indique comment  $\mathbf{x}$  contribue à l'estimation de la moyenne. On peut alors calculer la moyenne  $\mathbf{m}$  de  $\mathbf{x}$ :

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^n K(\mathbf{x} - \mathbf{x}_i)}{\sum_{i=1}^n K(\mathbf{x} - \mathbf{x}_i)}$$

On appelle la différence  $\mathbf{m}(\mathbf{x}) - \mathbf{x}$  mean shift.

Un noyau est vérifie les propriétés suivantes:

$$\begin{aligned} \int_{R^d} K(x) dx &= 1 \lim_{\|x\| \rightarrow \infty} \|x\|^d K(x) \\ \int_{R^d} x K(x) dx &= 0 \quad \int_{R^d} x x^T K(x) dx = c_k I \end{aligned}$$

La méthode la plus populaire pour estimer la densité de probabilité est la technique de la fenêtre de Parzen, ou estimation par Noyau. Cette méthode généralise la méthode d'estimation par histogramme. Celle-ci estime la densité en un point  $x$  par la proportion  $x_1, \dots, x_n$  aux alentours de  $x$ . Une technique classique pour effectuer une telle tâche consiste à tracer une boîte dont la largeur dépend d'un paramètre de lissage  $h$ , et à compter le nombre d'observation appartenant à cette boîte. La méthode du noyau remplace la boîte centrée en  $x$  de largeur  $h$  par une courbe centrée en  $x$ . La moyenne des courbes en cloche forme l'observateur.

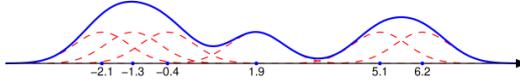


Figure 1: Rouge: courbe en cloche. Bleu: somme des courbes en cloche. L'estimateur est la moyenne des courbes rouges, ie, la courbe bleue divisée par 6

Formalisons cette notion. Sachant  $n$  point  $x_i, i = 1, \dots, n$  dans l'espace  $R^d$ , une bande passante  $H$  de taille  $d \times d$ , un noyau  $K(x)$ , l'estimateur de la densité est:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n K_H(x - x_i) \tag{1}$$

Pour les noyaux à symétrie circulaire, il suffit de définir le profile du noyau  $k(x)$ . On a alors:

$$K(x) = c_{k,d} k(\|x\|^2) \quad (2)$$

$x_{k,d}$  est une constante de normalisation strictement positive, permettant d'assurer la contrainte suivante:

$$\int_{R^d} K(x) dx = 1$$

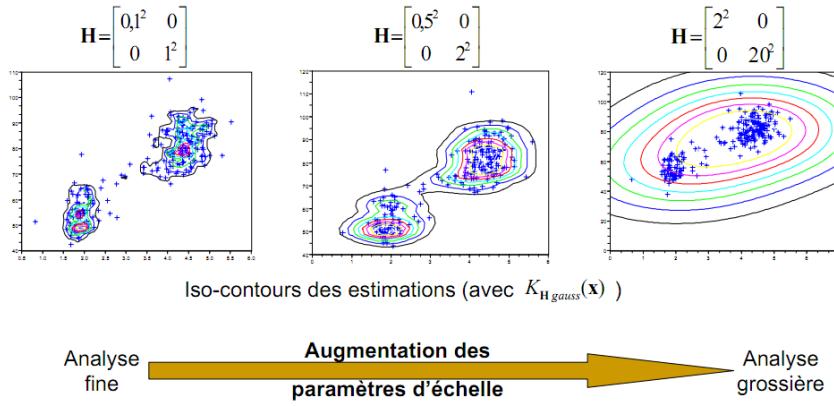


Figure 2: Influence des paramètres sur les estimations

Si complétement paramétrisée, la bande passante  $H$  permet de raffiner l'estimation. Elle est cependant souvent choisie comme étant diagonale  $H = \text{diag}[h_1^2, \dots, h_n^2]$ , ou proportionnelle à l'identité  $H = h \times I$ . Nous nous restreindrons à ce dernier cas, ce qui permet de ne définir qu'un seul paramètre. Avec cette contrainte supplémentaire, la densité s'écrit:

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (3)$$

[2] proposent deux exemples de noyaux:

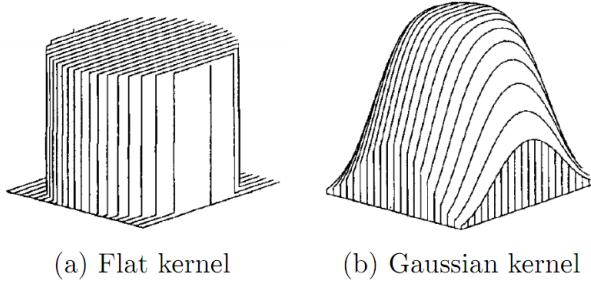
- noyau plat:

$$K(x) = 1 if \|x\| \leq 1 else 0$$

- noyau gaussian

$$K(x) = \exp(-\|x\|^2)$$

Nous avons maintenant tous les outils pour calculer le gradient de l'estimateur de densité:



$$\begin{aligned}
 \hat{\nabla} f_{h,K}(x) &= \nabla \hat{f}_{h,K}(x) \\
 &= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^n (x - x_i) g\left(\frac{\|x - x_i\|^2}{h}\right) \\
 &= \frac{2c_{k,d}}{nh^{d+2}} \left[ \sum_{i=1}^n (x - x_i) \right] \left[ \frac{\sum_{i=1}^n x_i g\left(\frac{\|x - x_i\|^2}{h}\right)}{\sum_{i=1}^n g\left(\frac{\|x - x_i\|^2}{h}\right)} - x \right]
 \end{aligned}$$

avec  $g = -k'$ . On remarque que le premier terme est proportionnel à l'estimateur de densité au point  $x$  (calculé avec le noyau  $G = c_g g(\|x\|^2)$ ). Le deuxième terme est le vector mean shift  $m$ , qui pointe vers la direction de l'augmentation maximal de densité, et est proportionnal au gradient de l'estimateur de densité au point  $x$ , avec le noyau  $K$ . On peut donc réécrire l'algorithme mean shift au point  $x_i$  tel qu'il suit:

- Calculer le vecteur meanshift  $m(x_i^t)$
- Déplacer la fenêtre d'estimation de densité  $x_i^{t+1} = x_i^t + m(x_i^t)$ .
- Itérer jusqu'à converge, ie jusqu'à  $\nabla f(x_i) = 0$

Nous utiliserons l'implémentation du MeanShift en python de scikit-learn, ([6]).

### 3 Descripteurs pour la segmentation

Parmi les descripteurs intuitifs d'une image, on peut citer les différents espaces de couleurs. Des descripteurs de texture peuvent aussi être utilisés, ainsi qu'une combinaison des descripteurs de couleurs et de texture.

#### 3.1 Couleurs

On dispose de plusieurs espaces de couleurs, que nous pouvons utiliser comme descripteurs:

- niveau de gris: le descripteur le plus simple est certainement le niveau de gris. Étant d'une dimension, le calcul du meanshift est particulièrement rapide sur celui-ci. Il est d'un intérêt limité.

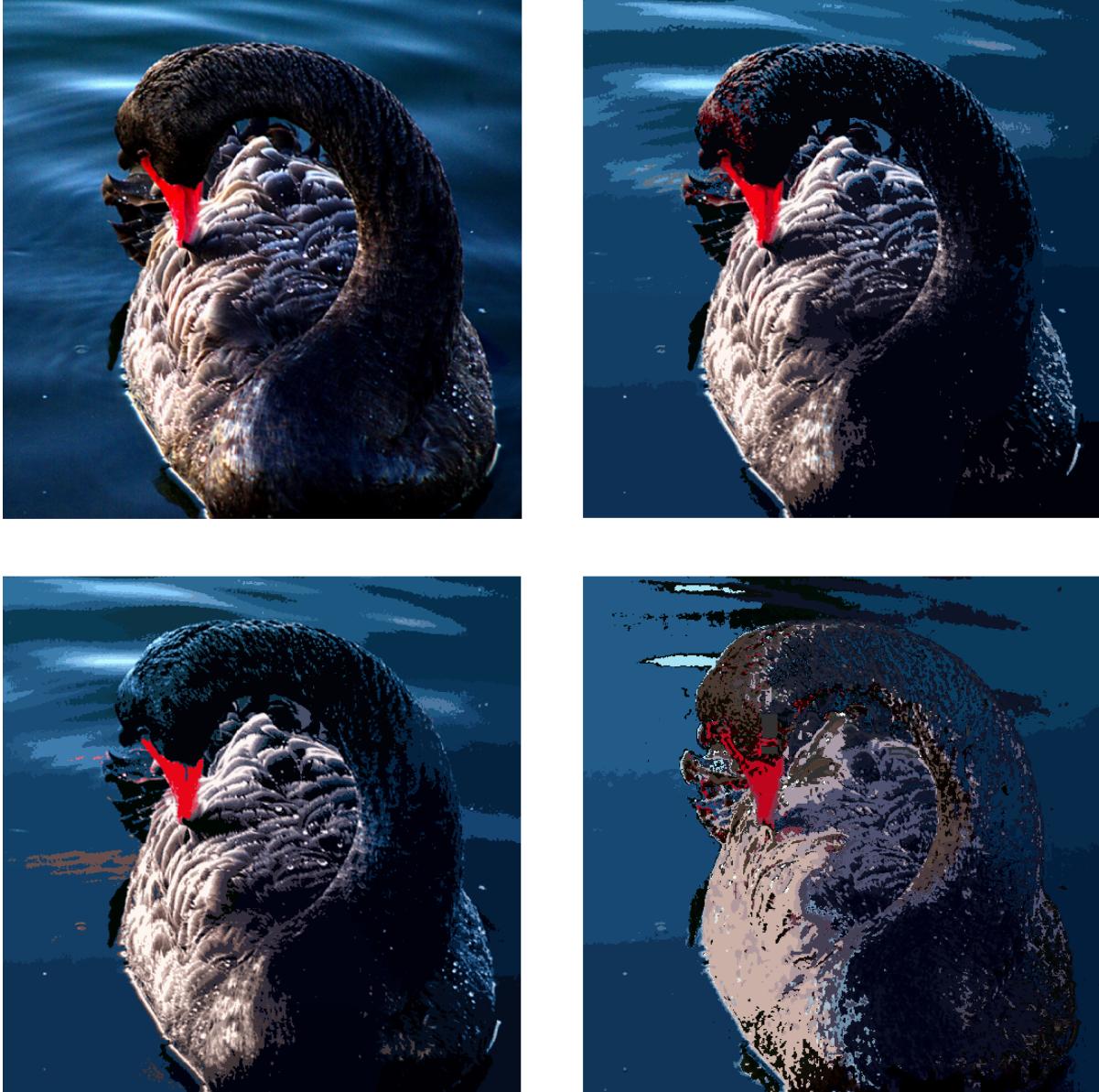


Figure 3: Haut gauche - Image originale: Black Swan, 512 \* 512, par Agnès Maillard. Haut gauche - MeanShift RGB - MeanShift HSV - Meanshift XYZ. Les meanshift sont calculées avec les mêmes paramètres.

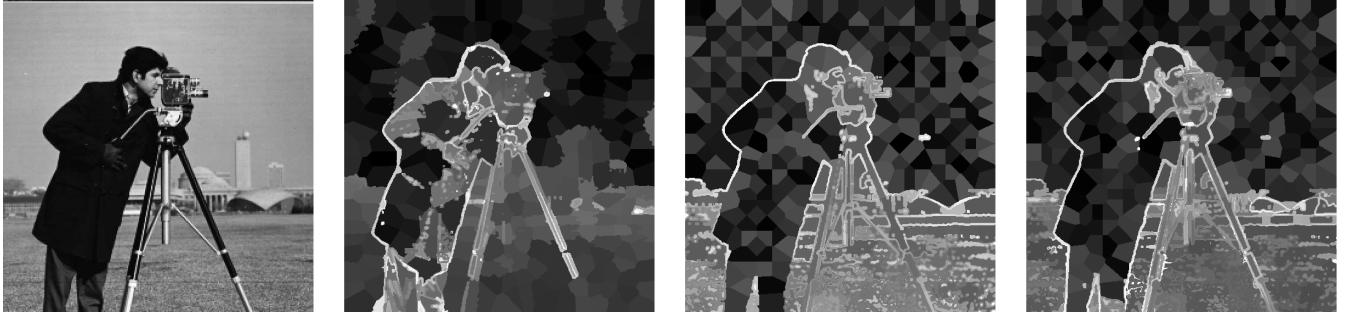


Figure 4: Cameraman, segmentée avec plusieurs descripteurs de texture (histogramme de niveaux de gris. Image original, MeanShift sur histogramme quantifié sur 16bins, 4bins et 2bins et descripteurs spatiaux)

- RGB: cet espace correspond aux couleurs primaires, qui correspondent aux trois longueurs d'onde auxquelles répondent l'œil humain.
- HSV (hue saturation value): cet espace de teinte, saturation et valeur, est obtenu par une transformation non linéaire de l'espace de couleur RGB
- XYZ: cet espace se rapproche vers une description des couleurs conforme à la vision humaine, celui introduisant la notion subjective de la luminance.

Nous utilisons la librairie scikits-image pour effectuer les changements d'espace de couleur.

### 3.2 Textures

Il existe plusieurs manières de représenter les textures. Nous nous limiterons ici aux textures microscopiques: nous ignorerons donc le cas des textures macroscopiques, dont nous pouvons distinguer les éléments géométriques (pommes, galets etc...). Il est cependant bon de rappeler qu'une texture macroscopique peut devenir microscopique avec un changement d'échelle.

Une texture microscopique est très bien représentée par les statistiques de premier ordre:

- Moyenne:  $\mu(I) = \frac{1}{N^2} \sum_z I(z)$
- Variance:  $\sigma^2(I) = \frac{1}{N^2} \sum_z (I(z) - \mu)^2$
- Energie:  $E(I) = \frac{1}{N^2} \sum_z (I(z))^2$
- Entropie:  $H(I) = - \sum_{g=1}^G f(g) \log(f(g))$

Ces quantités ne dépendent que de l'histogramme de l'image, et peuvent donc prendre des valeurs arbitraires sous l'effet d'un changement de contraste. Ces descripteurs sont estimés sur des voisinages bornés, avec des patchs glissants: ils sont donc approximativement liés à une localisation.

Nous utiliserons des histogrammes de niveaux de gris, calculés sur des petits patchs d'image de taille  $5 \times 5$ . Ces histogrammes peuvent être quantifiés sur un nombre variable de bins afin d'avoir des descripteurs variés. 3.2 montre le résultat d'une segmentation en utilisant des descripteurs différents, de tailles variantes entre 5 et 19 éléments.

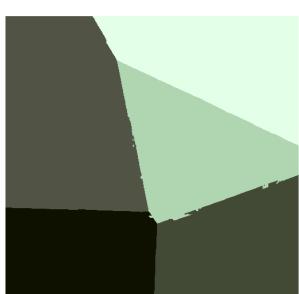
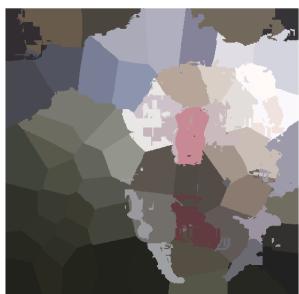
## 4 Conclusion



Malgré la volonté de [4] de limiter le nombre de paramètre à trouver, le meanshift nécessite de trouver la bande passeante ou le quantile le plus performant. Ne pas le choisir correctement peut mener à des résultats catastrophiques. Dans certains cas, il est donc plus simple d'utiliser le k-means, dont le paramètre  $k$  correspond au nombre de cluster, et est donc plus intuitif à trouver (Figure 3.2). Cependant, il existe des moyens d'estimer la bande passeante, comme décrit dans [1], afin d'avoir un algorithme non paramétrique.



## References



- [1] Miguel Á. Carreira-perpiñán. Fast nonparametric clustering with gaussian blurring mean-shift. In *In Int. Conf. on Machine Learning*. ACM, 2006.
- [2] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17:790–799, August 1995.
- [3] D Comaniciu and P Meer. Robust analysis of feature spaces : Color image segmentation dorin comaniciu. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, C(08855):750–755, 1997.
- [4] D Comaniciu and P Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [5] K Fukunaga and L Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.
- [6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Duchesnay E. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.

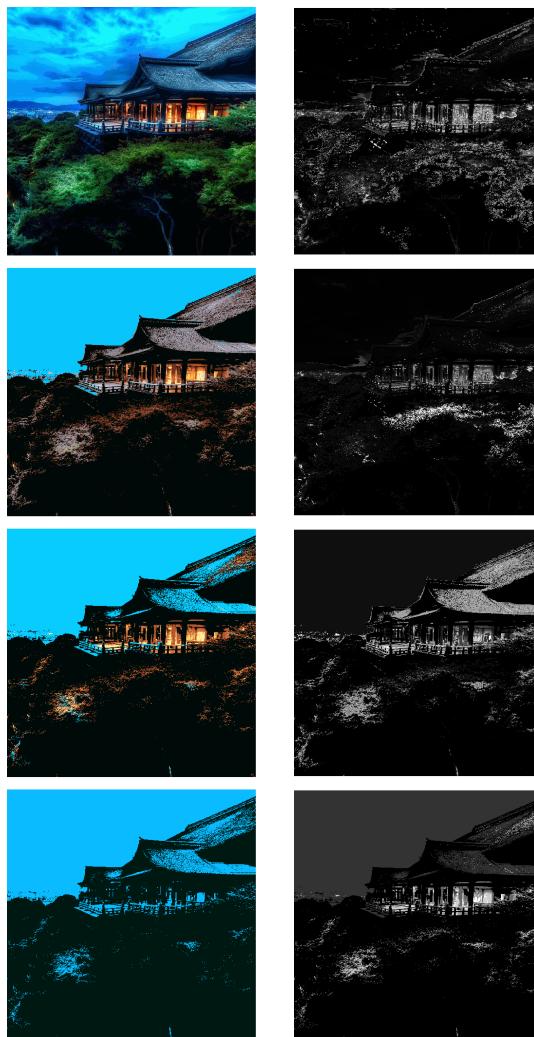


Figure 5: Chinese Temple (HDR) by Trey Ratcliff, segmented using RGB (left) and texture (right), with varying quantiles

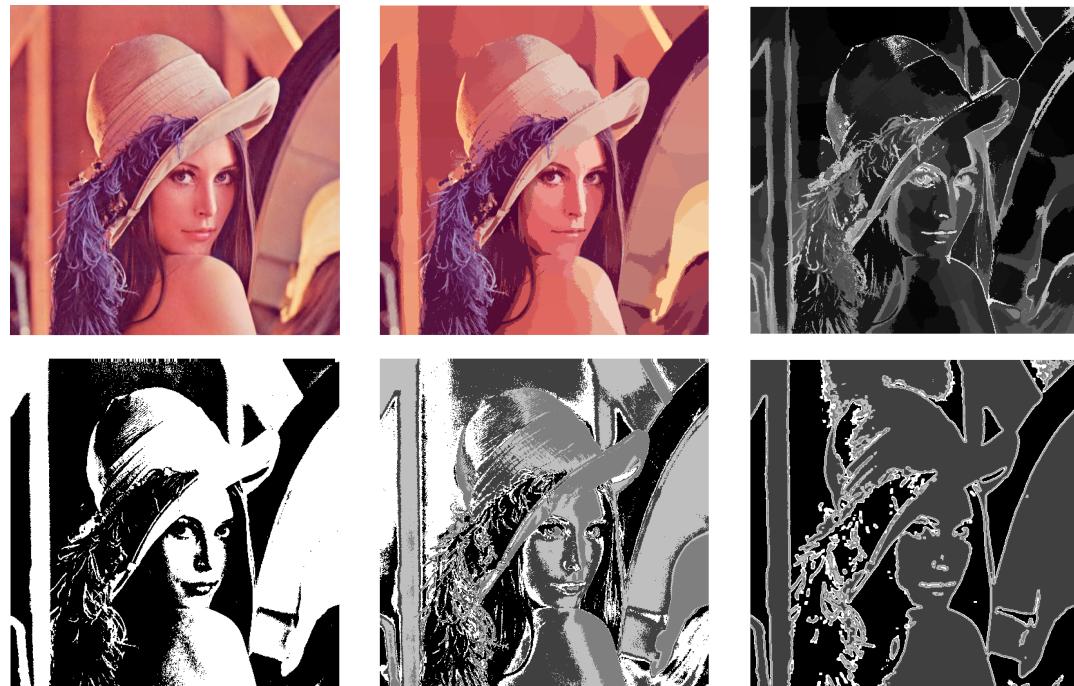


Figure 6: Lena - Haut gauche: image originale - haut milieu: image segmentée à l'aide du meanshift, dans l'espace des descripteurs RGB + spatial - haut droit: image segmentée avec le meanshift dans l'espace des descripteurs de textures (6bins), des couleurs (RGB) et spatiale - bas milieu: image segmentée avec le k-means,  $k = 2$ , dans l'espace des couleurs et spatiale - bas droit: image segmentée avec le k-means,  $k = 5$ , dans l'espace des couleurs RGB et spatial - bas droit: image segmentée avec le k-means,  $k = 5$ , dans les espaces des descripteurs de textures (6bins), des couleurs (RGB) et spatiale.